Rev: 1.3

Nanochap

Hangzhou Nannochap Electronics Co., Ltd.

ENS001-A_Datasheet

26 September 2023

Revision History

Rev	Date	Author	Description
1.0	29/11/2022	Nanochap	First Draft
1.1	30/3/2023	Nanochap	Second Draft
1.2	31/3/2023	Nanochap	Third Draft Add reminder
1.3	7/7/2023	Nanochap	Modify some parameters

Contents

List of Tables

List of Figures

1 Overview

The ENS1A is a single chip stimulator device. It integrates a powerful ARM MCU with an integrated battery charger circuit, power supply switcher, a high-voltage boost converter and a high-compliance stimulation block. The ENS1A can be utilized in targeted applications with minimal off-chip components. The stimulator block is designed to drive anodic and cathodic stimulation currents pulses up to 60mA. Eight channels of drivers are available in this device. The ENS1A generates multiple stimulation patterns to support various applications. The system can be configured to support intermediate frequency physiotherapy, conventional TENS, Muscle Rehabilitation and implantable stimulation.

1.1 Applications

- Muscle Strengthening and Weak Muscle Rehabilitation
- Intermediate Frequency Physiotherapy
- Deep Brain Stimulation
- Spinal Cord Stimulation
- Cochlear Implant

1.2 Features

- Operation temperature range: -40℃ to 85℃
- Operating voltage range: 2.5V to 5V
- Integrated 5~60V High Voltage Boost Converter
- Core
  - 32-bit ARM Cortex-M0 CPU
  - Frequency up to 32MHz
- Memories
  - 32Kbytes MTP memory
  - 8Kbytes SRAM
- Clocks
  - HSI RC 4-32MHz
  - Up to 32MHz external clock
  - LSI RC about 32KHz

- LSE OSC with external crystal 32.768KHz

- Low power mode
  - Sleep mode
  - low-power run mode
  - Low-power sleep mode
  - Stop mode
- 4 Wide-Range Drivers (8 electrodes) (Max 60V)
  - Output current: 33uA~67 mA, with 255 steps
  - Output unit current: 33uA~264uA, 8 steps
  - 2us~infinity pulse width
  - Up to 250 kHz sinusoidal/triangle/square or arbitrary waveform
  - Can be used for TENS, IFT, EMS
- 8 Middle-Range Drivers (16 electrodes) (Max 60V)
  - Output current: 50uA~52 mA, with 255 steps
  - Output unit current: 50, 72 ~ 204 uA (8 steps)
  - 2us~infinity pulse width
  - Up to 250 kHz sinusoidal/triangle/square or arbitrary waveform
  - Can be used for DBS, SCS
- 23 Channels Low-Range Drivers (24 electrodes) (Max 60V)
  - 8uA~2 mA output current, 255 steps of 8uA
  - 2us~infinity pulse width
  - Up to 250 kHz sinusoidal/triangle/square or arbitrary waveform
  - Can be used for Cochlear Implant
- Peripheral analog circuits
  - 12-bit ADC: 0 to VDD conversion range
  - Temperature Sensor
  - COMPx2
  - PGA
  - Integrated battery charger
  - Power-on/Power down reset (POR/PDR)

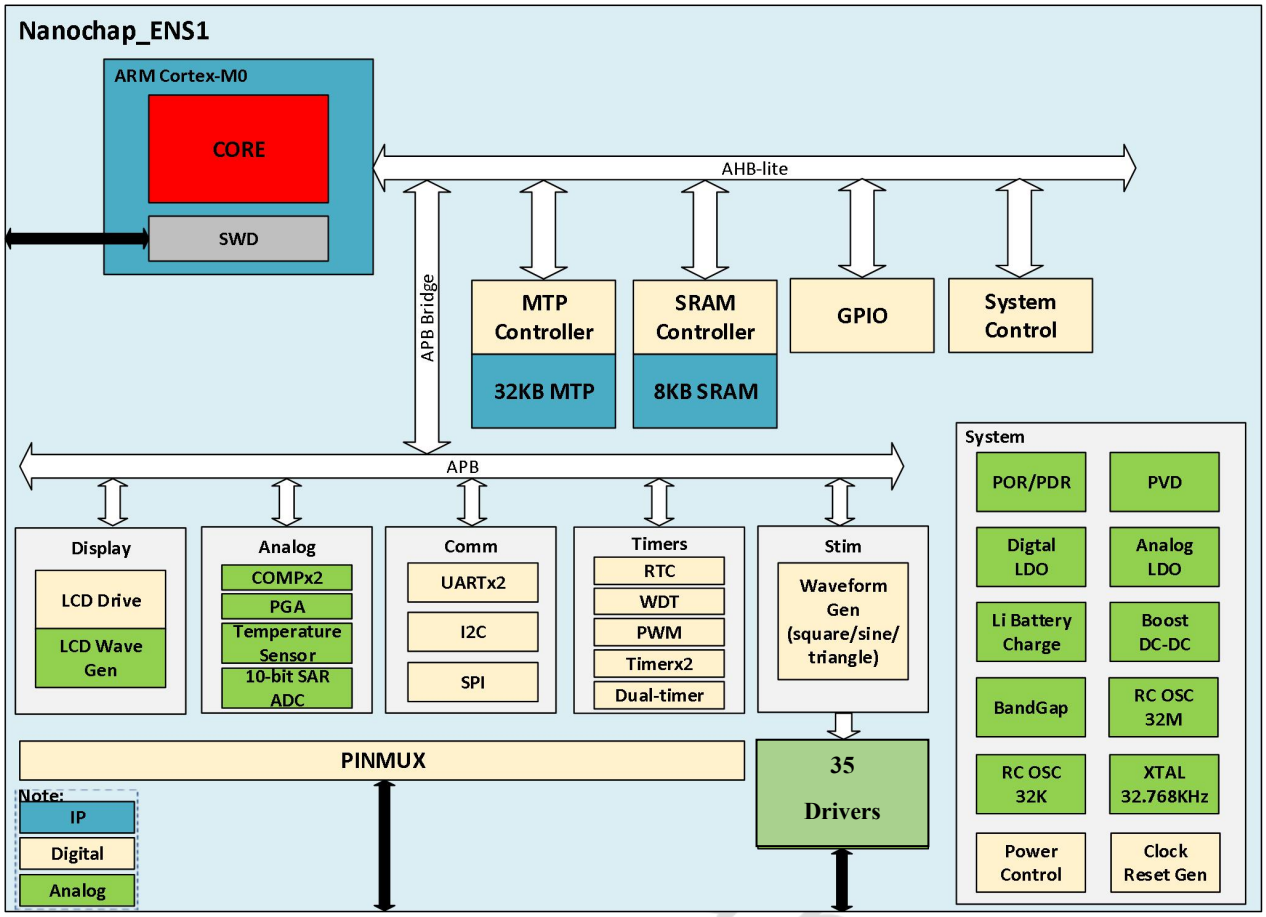
- Low Voltage Detector (LVD)

- 24 GPIOs
- LCD Drive
  - COMx4, SEGx16
- 96-bit unique ID
- Communication Interface
  - UART x2, with hardware flow control
  - SPI x2, with master/slave mode
  - I^2C x2, with master/slave mode
- Various timers
  - Real Time Clock (RTC)
  - Watchdog Timer (WDT)
  - Pulse Width Modulation (PWM)
  - 32-bit Timer x2
  - 32-bit or 16-bit dual-timer
  - SysTick Timer

1.3 Block Diagram

Figure 1 Top Level Block Diagram

Block diagram:



2 Pin Description

2.1 Pin and Package definition

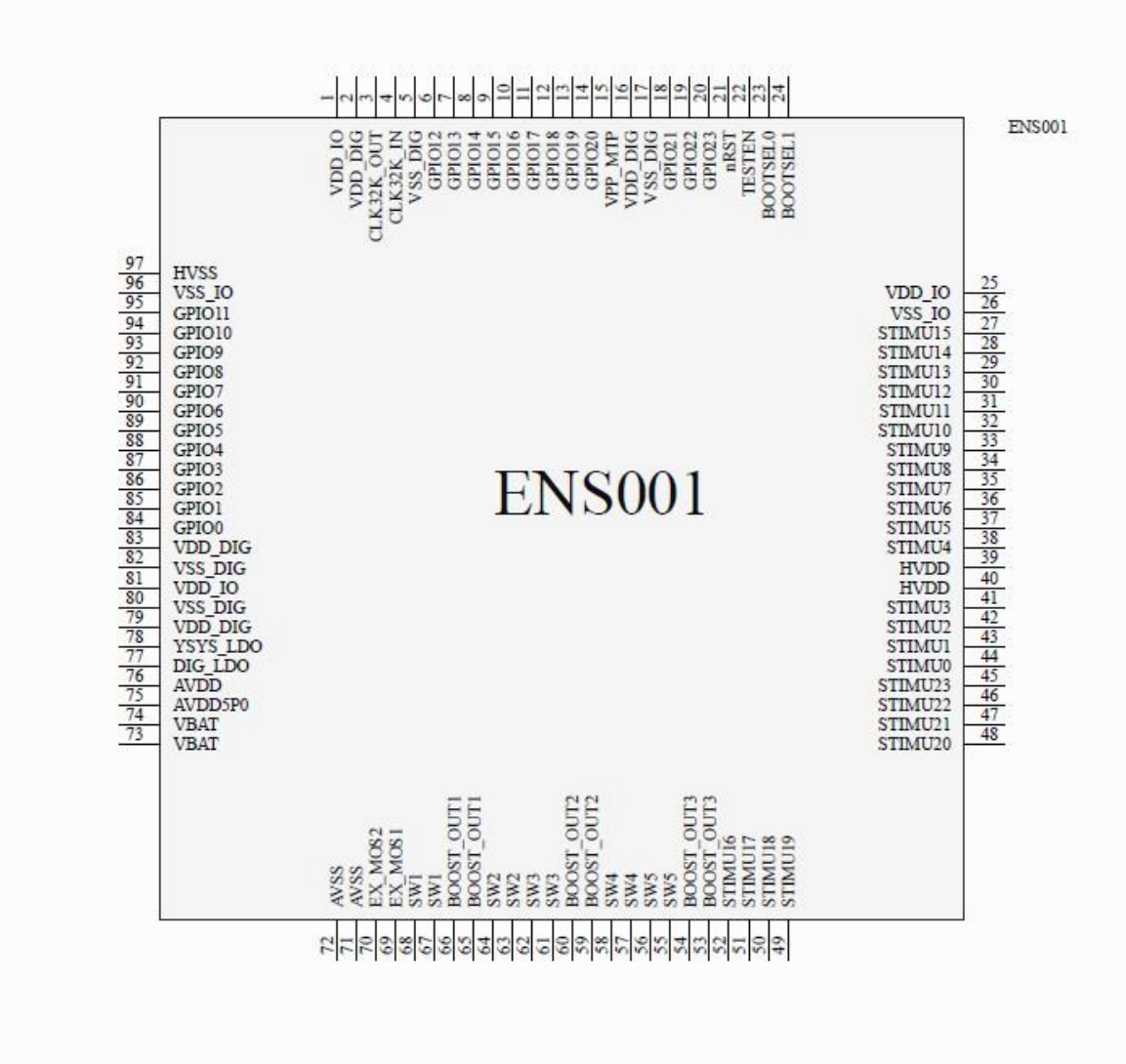


Table 1 Interface Signal Table

Pin No.	PIN NAME	Primary Funtion (ALT Function 0)	ALT Function 1	ALT Function 2	ALT Function 3	
1	VDD_IO	3.3V IO Supply	-	-	-	-
2	VDD_DIG	1.8V Digital Core Supply	-	-	-	-
3	CLK32K_OUT	32 kHz Crystal OUT	-	-	-	-
4	CLK32K_IN	32 kHz Crystal IN	-	-	-	-
5	VSS_DIG	Digital Ground	-	-	-	-
6	GPIO12	GPIO12	UART1_RXD	MCO	SEG4	COMP1_VIP0
7	GPIO13	GPIO13	UART1_TXD	SPI0_NSS1	SEG5	COMP1_VIP1
8	GPIO14	GPIO14	UART1_RTS_N	SPI0_NSS2	SEG6	COMP1_VIN0
9	GPIO15	GPIO15	UART1_CTS_N	SPI0_NSS3	SEG7	COMP1_VIN1
10	GPIO16	GPIO16	PWM1_OUT	SPI1_SCK	SEG8	PGA_VIP0
11	GPIO17	GPIO17	PWM2_OUT	SPI1_MOSI	SEG9	PGA_VIP1
12	GPIO18	GPIO18	PWM3_OUT	SPI1_MISO	SEG10	PGA_VIN0
13	GPIO19	GPIO19	PWM4_OUT	SPI1_NSS0	SEG11	PGA_VIN1
14	GPIO20	GPIO20	PWM5_OUT	COMP0_OUT	SEG12	PGA_EXVCM
15	VPP_MTP	MTP Memory VPP pin	-	-	-	-
16	VDD_DIG	1.8V Digital Core Supply	-	-	-	-
17	VSS_DIG	Digital Ground	-	-	-	-
18	GPIO21	GPIO21	PWM6_OUT	COMP1_OUT	SEG13	ADC_IN0
19	GPIO22	GPIO22	TIMER0_EXTIN	-	SEG14	ADC_IN1
20	GPIO23	GPIO23	TIMER1_EXTIN	-	SEG15	ADC_IN2
21	nRST	Reset, Active LOW, Default = 1	-	-	-	-
22	TESTEN	TEST Enable, Active HIGH, Default = 0	-	-	-	-
23	BOOTSEL0	Boot Select 0, Default = 0	-	-	-	-
24	BOOTSEL1	Boot Select 1, Default = 0	-	-	-	-
25	VDD_IO	3.3V IO Supply	-	-	-	-
26	VSS_IO	IO Ground	-	-	-	-
27	STIMU15	Driver B and Driver C Outputs /DB_ELE15/DC_ELE7	-	-	-	-
28	STIMU14	Driver B and Driver C Outputs /DB_ELE14/DC_ELE6	-	-	-	-
29	STIMU13	Driver B and Driver C Outputs	-	-	-	-

		/DB_ELE13/DC_ELE5				
30	STIMU12	Driver B and Driver C Outputs /DB_ELE12/DC_ELE4	-	-	-	-
31	STIMU11	Driver B and Driver C Outputs /DB_ELE11/DC_ELE3	-	-	-	-
32	STIMU10	Driver B and Driver C Outputs /DB_ELE10/DC_ELE2	-	-	-	-
33	STIMU9	Driver B and Driver C Outputs /DB_ELE9/DC_ELE1	-	-	-	-
34	STIMU8	Driver B and Driver C Outputs /DB_ELE8/DC_ELE0	-	-	-	-
35	STIMU7	Driver A and Driver B Outputs /DA_CH3_Anode/DB_ELE7	-	-	-	-
36	STIMU6	Driver A and Driver B Outputs /DA_CH3_Cathode/DB_ELE6	-	-	-	-
37	STIMU5	Driver A and Driver B Outputs /DA_CH2_Anode/DB_ELE5	-	-	-	-
38	STIMU4	Driver A and Driver B Outputs /DA_CH2_Cathode/DB_ELE4	-	-	-	-
39	HVDD	High Voltage Supply for Stimulation Drivers	-	-	-	-
40	HVDD	High Voltage Supply for Stimulation Drivers	-	-	-	-
41	STIMU3	Driver A and Driver B Outputs /DA_CH1_Anode//DB_ELE3	-	-	-	-
42	STIMU2	Driver A and Driver B Outputs /DA_CH1_Cathode/DB_ELE2	-	-	-	-
43	STIMU1	Driver A and Driver B Outputs /DA_CH0_Anode/DB_ELE1	-	-	-	-
44	STIMU0	Driver A and Driver B Outputs /DA_CH0_Cathode/DB_ELE0	-	-	-	-

45	STIMU23	Driver B and Driver C Outputs /DB_ELE23/DC_SW7	-	-	-	-
46	STIMU22	Driver B and Driver C Outputs /DB_ELE22/DC_SW6	-	-	-	-
47	STIMU21	Driver B and Driver C Outputs /DB_ELE21/DC_SW5	-	-	-	-
48	STIMU20	Driver B and Driver C Outputs /DB_ELE20/DC_SW4	-	-	-	-
49	STIMU19	Driver B and Driver C Outputs /DB_ELE19/DC_SW3	-	-	-	-
50	STIMU18	Driver B and Driver C Outputs /DB_ELE18/DC_SW2	-	-	-	-
51	STIMU17	Driver B and Driver C Outputs /DB_ELE17/DC_SW1	-	-	-	-
52	STIMU16	Driver B and Driver C Outputs /DB_ELE16/DC_SW0	-	-	-	-
53	BOOST_OUT3	DC-DC BOOSTER OUTPUT 3	-	-	-	-
54	BOOST_OUT3	DC-DC BOOSTER OUTPUT 3	-	-	-	-
55	SW5	DC-DC BOOSTER INDUCTOR-DIODE 5	-	-	-	-
56	SW5	DC-DC BOOSTER INDUCTOR-DIODE 5	-	-	-	-
57	SW4	DC-DC BOOSTER INDUCTOR-DIODE 4	-	-	-	-
58	SW4	DC-DC BOOSTER INDUCTOR-DIODE 4	-	-	-	-
59	BOOST_OUT2	DC-DC BOOSTER OUTPUT 2	-	-	-	-
60	BOOST_OUT2	DC-DC BOOSTER OUTPUT 2	-	-	-	-
61	SW3	DC-DC BOOSTER INDUCTOR-DIODE 3	-	-	-	-

62	SW3	DC-DC BOOSTER INDUCTOR-DIODE 2	-	-	-	-
63	SW2	DC-DC BOOSTER INDUCTOR-DIODE 2	-	-	-	-
64	SW2	DC-DC BOOSTER INDUCTOR-DIODE 2	-	-	-	-
65	BOOST_OUT1	DC-DC BOOSTER OUTPUT 1	-	-	-	-
66	BOOST_OUT1	DC-DC BOOSTER OUTPUT 1	-	-	-	-
67	SW1	DC-DC BOOSTER INDUCTOR-DIODE 1	-	-	-	-
68	SW1	DC-DC BOOSTER INDUCTOR-DIODE 1	-	-	-	-
69	EX_MOS1	Gate-control signal for external main switching NMOS for DC-DC Booster	-	-	-	-
70	EX_MOS2	Gate-control signal for external over-voltage discharging NMOS for DC-DC booster	-	-	-	-
71	AVSS	Analog Ground	-	-	-	-
72	AVSS	Analog Ground	-	-	-	-
73	VBAT	4.2V Battery Input	-	-	-	-
74	VBAT	4.2V Battery Input	-	-	-	-
75	AVDD5P0	5V DC input for charging battery	-	-	-	-
76	AVDD	3.3V LDO Output	-	-	-	-
77	DIG_LDO	1.8V LDO Output	-	-	-	-
78	VSYS_LDO	4.23V LDO output for battery charging circuit	-	-	-	-
79	VDD_DIG	1.8V Digital Core Supply	-	-	-	-
80	VSS_DIG	Digital Ground	-	-	-	-
81	VDD_IO	3.3V IO Supply	-	-	-	-
82	VSS_DIG	Digital Ground	-	-	-	-
83	VDD_DIG	1.8V Digital Core Supply	-	-	-	-
84	GPIO0	SWCLK	GPIO0	HSE_CLK	-	-
85	GPIO1	SWDIO	GPIO1	-	-	-
86	GPIO2	GPIO2	UART0_RXD	SPI1_NSS1	-	-
87	GPIO3	GPIO3	UART0_TXD	SPI1_NSS2	-	-
88	GPIO4	GPIO4	UART0_RTS_N	SPI1_NSS3	COM0	PGA_OUT
89	GPIO5	GPIO5	UART0_CTS_N	-	COM1	ANA_BIST

90	GPIO6	GPIO6	I2C0_SCL	–	COM2	-
91	GPIO7	GPIO7	I2C0_SDA	–	COM3	-
92	GPIO8	GPIO8	SPI0_SCK	I2C1_SCL	SEG0	COMP0_VIP0
93	GPIO9	GPIO9	SPI0_MOSI	I2C1_SDA	SEG1	COMP0_VIP1
94	GPIO10	GPIO10	SPI0_MISO	—	SEG2	COMP0_VIN0
95	GPIO11	GPIO11	SPI0_NSS0	RTC_1HZ	SEG3	COMP0_VIN1
96	VSS_IO	IO Ground	-	-	-	-
97	HVSS	High Voltage Ground - GND BOTTOM PAD	-	-	-	-

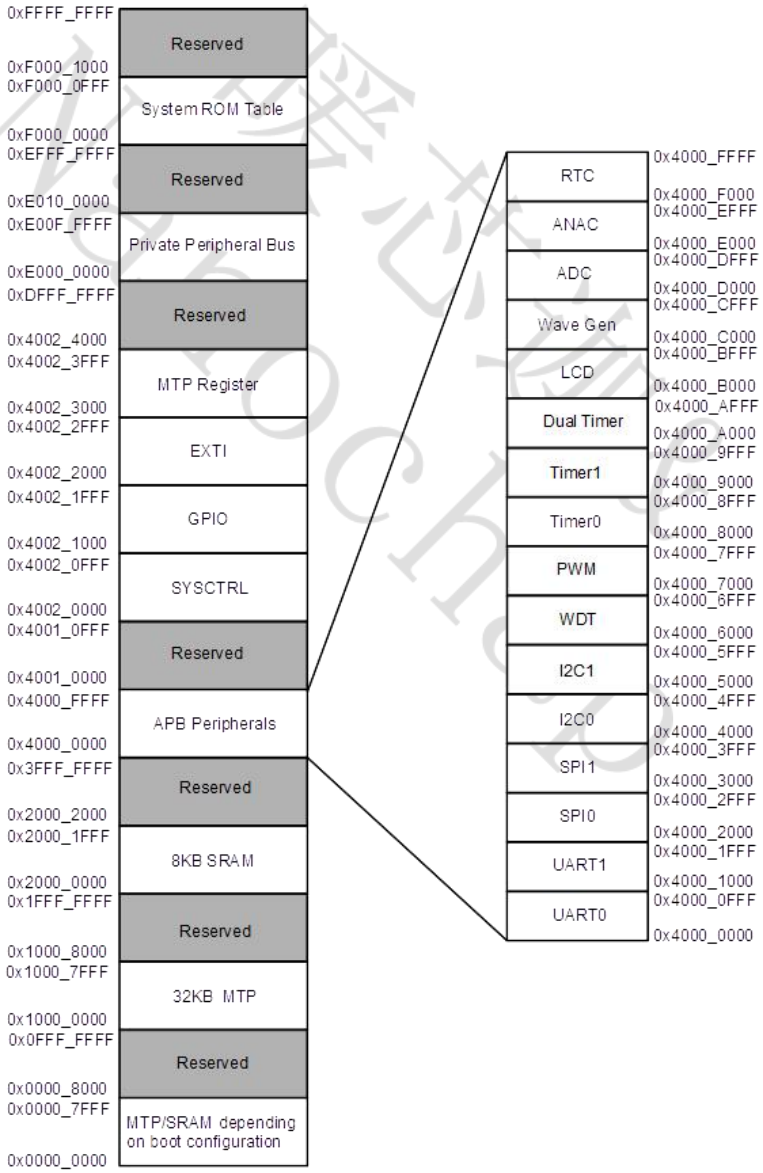
Note: Please refer to schematic diagram for design.If you need to use the BOOST circuit overvoltage, overtemperature function, please contact the manufacturer.

3 Memory

3.1 Memory Map

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space. The bytes are coded in memory in Little Endian format. All the memory map areas that are not allocated on-chip memories and peripherals are considered “Reserved”.

Figure 2 Memory Map Diagram



3.2 Embedded SRAM

The embedded SRAM capability is 8Kbytes. This SRAM can be accessed by bytes, half-words (16 bits) or full words (32 bits). This memory can be addressed at maximum system clock frequency without wait state.

The SRAM start address is 0x2000_0000.

The CPU can access the SRAM from address 0x0000_0000 when physical remap is selected through BOOT pin or REMAP register.

3.3 Boot Configuration

Three boot modes can be selected through the BOOT0 and BOOT1 pins, as shown in the following table.

Table 2 Boot Modes

Boot mode selection		Boot mode	Aliasing
Boot1 pin	Boot0 pin		
X	0	MTP base	MTP main area from base is selected as boot area for application code usage
0	1	MTP high 4KByte	MTP high 4KByte area is selected as boot area for bootloader usage
1	1	Embedded SRAM	Embedded SRAM is selected as boot area for debug usage

After startup delay has elapsed, CPU fetches the top-of-stack value from address 0x0000_0000, then starts code execution from the boot memory at 0x0000_0004.

Boot from MTP main area at base address: the MTP memory is aliased in the boot memory space (0x0000_0000), but still accessible from its original memory space (0x1000_0000). In other words, the MTP memory contents can be accessed starting from address 0x0000_0000 or 0x1000_0000.

Boot from MTP high 4KByte area: the MTP high 4Kbyte area is used for bootloader, it is aliased in the boot memory space (0x0000_0000), but still accessible from its original memory space (0x1000_7000) The remaining 28KByte space is used for application code.

Boot from the embedded SRAM: the SRAM is aliased in the boot memory space (0x0000_0000), but it is still accessible from its original memory space (0x2000_0000).

Once the BOOT0 and BOOT1 pins are selected, the application software can modify the memory in the code area. This modification is performed by programming the REMAP bits in SYSCTRL register.

BOOT0 and BOOT1 default is weak pull-down, when boot from bootloader space, that is boot0 pin need to be pulled high, software can clear the GPIO_PD[24] register to save power.

4 Multi Time Program (MTP)

4.1 Overview

YEG8K32F18B5AA1 is an embedded MTP IP macro fabricated in the Globalfoundries 0.18um BCDlite 1.8V - ULL 6V_40V-65V Process. The memory arrays are partitioned into three memory blocks each. One is the MTP block (8K x 32 bits); one is the Information block (32 x 32 bits); the other is the EEPROM block (512 x 8 bits).

The YEG8K32F18B5AA1 supports two operating mode: User mode and Test mode.

User modes provide five memory operation: reset(RESET) / standby(STANDBY) / static(STATIC) / read(READ) / write with internal high voltage (INTHV WRITE).

Test modes provide four memory operations: measure memory cell current(CLEN) / read for data retention test(MRGN READ) / write with internal high voltage for data retention test(MRGN INTHV WRITE) / write all MTP density with external high voltage(EXTHV WRITE ALL).

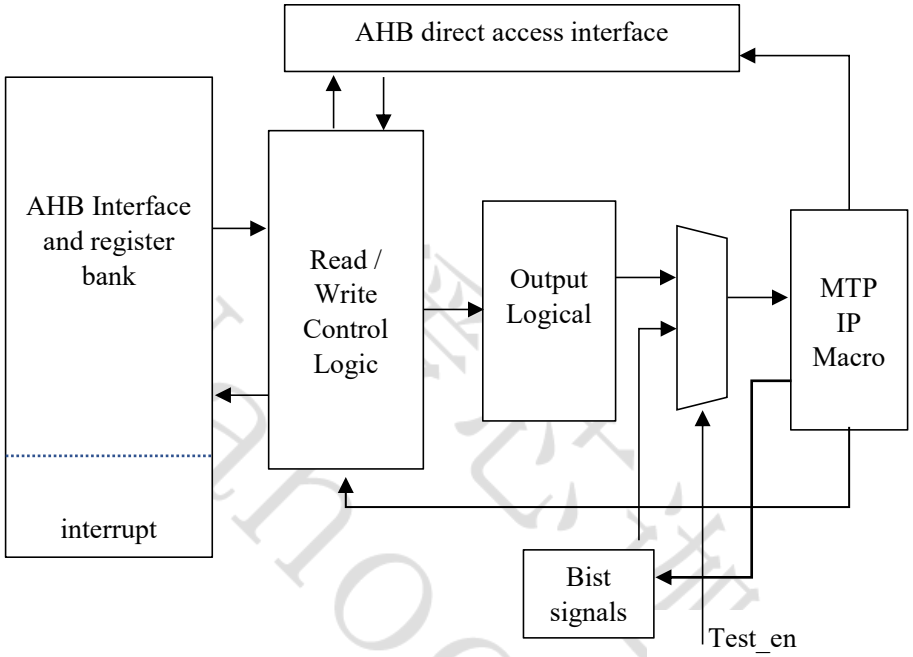
4.1.1 Feature List

- Memory organization: MTP block(8K x 32 bits); information block(32 x 32 bits); EEPROM block(512 x 8 bits).
- MTP block supports 32-bit read and sector write operation.
- Information block support 32-bit read and 32-bit write operation.
- EEPROM block support byte read and byte write operation.
- Re-write data without erase operation.
- Write time is not a fixed value.
- BUSY signal provides a hardware method of detecting write operation completion.
- Writing MTP key: 0x5A5A5A5A.

4.2 Block Diagram

A functional block diagram of the MTP controller is shown as below.

Figure 3 MTP Block Diagram



4.3 Function Description

4.3.1 Supported user operation mode

User modes can operate in one of the five following modes

- Reset(RESET)
- Standby(STANDBY)
- Static(STATIC)
- Read(READ)
- Write with internal high voltage(INTHV WRITE)

Operating mode selection is showed as below

Table 3 User operation mode

Pin Name	RESET	STANDBY	STATIC	READ		INTHV WRITE	
RESETB[2]	L	H	H	H		H	
CS	L	L	H	H		H	
READ	L or H	L or H	L	H		L	
WR	L or H	L or H	L	L		H	
CLEN	L or H	L or H	L	L		L	
IFREN	L or H	L or H	L or H	L or H	L	L or H	L
EEPROM	L or H	L or H	L or H	L	H	L	H
SRL	L or H	L or H	L	L		L	
MRGN	L or H	L or H	L	L		L	
HVEN	L or H	L or H	L	L		L	
WRALL	L or H	L or H	L	L		L	
VPP	VDD2						

4.3.2 Definition of Sector Write operation in MTP block

Memory array in MTP block is integrated by several sectors. The ADDRESS of each sector defined and described as below table.

Only “sector write” operation is allowed on MTP block.

Write data into memory array randomly is not allowed. Write operations are initiated continuously from the 1st address to the last address of the target sector.

Table 4 MTP block memory array configuration

Sector NO.	Used Address	Sector NO.	Used Address
Sector 0	0000H ~ 03FFH	Sector 4	1000H ~ 13FFH
Sector 1	0400H ~ 07FFH	Sector 5	1400H ~ 17FFH
Sector 2	0800H ~ 0BFFH	Sector 6	1800H ~ 1BFFH
Sector 3	0C00H ~ 0FFFH	Sector 7	1C00H ~ 1FFFH

4.3.3 Definition of Bootloader Sector in MTP block

Bootloader sector address is sector7 in MTP block. When DBGPA_EN and BOOTLD_WEN turn off, bootloader sector can be written by SWD. The software application only can read bootloader sector.

4.3.4 Definition of trim and ID configuration information region in information block

Trim configuration information region address is from 0x0018 to 0x001B. Device ID configuration information region address is from 0x001C to 0x001E. They is used to store analog trim values and device ID values that will be automatically loaded to relational configuration registers after power on and reset.

4.3.5 Definition of user configuration information region in information block

User configuration information region address is 0x001F. It is used to store user configuration information that will be automatically loaded to configuration registers after power on and reset.

Table 5 user configuration information definition

Bit	Field Name	Field Description	Default
31:24	-	-	-
23:16	BOOTLD_SWDPA_EN	Bootloader SWD access protection enable: 0x55: on Other: off	0x55
15:8	DBGPA_EN	Debug port access protection enable: 0xAA: on Other: off	0xAA
7:0	SEC_ACL_EN	application code lock enable: 0X33: enable aclock Other: forbid aclock	0x33

4.3.6 User access authority for MTP

Table 6 MTP access authority

MTP region		MTP_DBGPA_EN	SEC_ACL	BOOTLD_SWDPA_EN	SWD	Application
MTP block	Sector0 - sector6	1	0	x	-	Read/write
			1	x	-	-
		0	0	x	Read/write	Read/write
			1	x	Read/write	-
	Bootloader region Sector7	1	x	1	-	Read
				0	-	Read
		0	x	1	-	Read
				0	Read/write	Read
Information block	Other region	1	x	x	-	Read/write
		0	x	x	Read/write	Read/write
	Trim and ID region	1	x	x	-	Read
		0	x	x	Read/write	Read
	User region	1	x	x	-	Read
		0	x	x	Read/write	Read
EEPROM block		1	x	x	-	Read/write
		0	x	x	Read/write	Read/write

4.3.7 Timing parameters

Parameter		Symbol	Min.	Typ.	Max.	Unit
VDD1 Setup Time		Tpws	100	-	-	ns
VDD1 Hold Time		Tpwh	100	-	-	ns
VDD2 Setup Time		Trsts	100	-	-	ns
VDD2 Hold Time		Trsth	100	-	-	ns
Setup Time From RESETB to CS		Trscs	40	-	-	us
Hold Time From RESETB to CS		Trsch	100	-	-	ns
CS to READ / WR Setup time		Tcsctrls	100	-	-	ns
Address/Data Setup Time		Tads	20	-	-	ns
Address Hold Time		Tadhr	50	-	-	ns
Read Pulse Width	MTP and Information blocks	Trpw	60	-	-	ns
	EEPROM block		250	-	-	ns
READ Access Time	MTP and Information blocks	Trac	-	-	60	ns
	EEPROM block		-	-	250	ns
Read Cycle Time	MTP and Information blocks	Trc	80	-	-	ns
	EEPROM block		270	-	-	ns
DOUT Hold Time		Tdoh	3	-	-	ns
WR Pulse Width		Twpw	100	-	-	ns
Address/Data Hold Time for Write		Tadhw	50	-	-	ns
Write Time	MTP and Information blocks	Twr	-	0.5	-	ms
(changed code)	EEPROM block		-	0.15	-	ms
Write Time (non-changed code)			1	-	-	us
BUSY Access Time to WR		Tbas	-	-	100	ns
Write Recovery Time		Twrc	100	-	-	ns
Measurement Current Wait Time		Tmcw	100	-	-	ns
Address Hold Time for CLEN		Tadhc	20	-	-	ns

5 System Control Unit (SCU)

5.1 PMU

By default, MCU is in run mode after a system or a power-on reset. In run mode the CPU is clocked by HCLK (default is 8MHz from HSI RC) and the program code is executed. Several low-power modes are available to save power when the CPU does not need to be kept running. When RTC/LCD are not used in some applications, the LSI RC (about 32kHz) and LSE OSC with external crystal (accurate 32.768kHz) can be disabled to save power by software. Also there are peripherals enable bits in SYSCTRL register, if some peripherals are not used, user can disable the peripherals to gate their clocks.

Five power modes are supported:

- Run mode: normal run mode, CPU is running at high frequency clock, all peripherals can be active.
- Sleep mode: HCLK is gated. CPU/All AHB peripherals clock off, all APB peripherals and core peripherals such NVIC, SysTick, etc. can run and wake up the CPU when an interrupt or an event occurs.
- Low-power run mode: system clock(SYSCLK) is switched to LFCLK which is about 32KHz. To save power, User can copy program to SRAM to execute.
- Low-power sleep mode: system clock(SYSCLK) is switched to LFCLK which is about 32KHz. Same as sleep mode, HCLK is gated.
- Stop mode: HSI RC is disabled. LFCLK can be kept running.

In addition, the power consumption in run mode can be reduced by one of the following means:

- Slowing down the system clocks
- Gating the clocks to the APB and AHB peripherals when they are unused.

If the SLEEPONEXIT bit of the SCR is set to 1, when the processor completes the execution of an exception handler and returns to thread mode it immediately enters sleep mode. Use this mechanism in applications that only require the processor to run when an interrupt occurs.

Normally, the processor wakes up only when it detects an exception with sufficient priority to cause exception entry.

Some embedded systems might have to execute system restore tasks after the processor wakes up, and before it executes an interrupt handler. To achieve this set the PRIMASK bit to 1. If an interrupt arrives that is enabled and has a higher priority than current exception priority, the processor wakes up but does not execute the interrupt handler until the processor sets PRIMASK to zero.

Table 7 Low-power Mode Summary

Mode name	Entry	Wakeup source	Wakeup system clock	Effect on clocks
Sleep (Sleep-now or Sleep-on-exit)	WFI or return from ISR	Any interrupt	Same as before entering sleep mode	HCLK is off
	WFE	Wakeup event		
Low-power run	Set SYSCLK_SEL to 2'b1x	Set SYSCLK_SEL to 2'b0x	No change	None
Low-power sleep	Set SYSCLK_SEL to 2'b1x + WFI or return from ISR	Any interrupt	Same as before entering low-power sleep mode	HCLK is off
	Set SYSCLK_SEL to 2'b1x + WFE	Wakeup event		
Stop	SLEEPDEEP bit + WFI or return from ISR or WFE	Any EXTI line (configured in the EXTI registers) Specific peripherals events	HSI RC	All clocks off except LSI and LSE

Table 8 Functionalities depending on the working mode

Function	Run	Sleep	Low-power run	Low-power sleep	Stop
CPU	Y	-	Y	-	-
MTP memory	Y	-	Y	-	-
SRAM	Y	-	Y	-	-
HSI	O	O	O	O	-
LSI	O	O	O	O	O
LSE	O	O	O	O	O
LVD	O	O	O	O	O

COMP0/1	O	O	O	O	O
ADC	O	O	O	O	-
Wave generator	O	O	O	O	-
RTC	O	O	O	O	O
LCD	O	O	O	O	O
UART0/1	O	O	O	O	-
I2C0/1	O	O	O	O	-
SPI0/1	O	O	O	O	-
Timer/dual-timer	O	O	O	O	-
WDT	O	O	O	O	-
PWM	O	O	O	O	-
GPIOs	O	O	O	O	O

Y = Yes. O = Optional

5.2 Reset Control

ENS1A reset control includes the control of two kinds of reset, power reset and system reset. The power on reset, known as a cold reset, resets the full system during a power up. A system reset resets the processor core and peripheral IP components with the exception of DBG control and RTC.

The Power reset is generated by Power On reset and Power Down reset (POR/PDR reset). The power reset sets all registers to their reset values. The power reset which active signal is low will be de-asserted when the internal LDO voltage regulator ready to provide 1.8V power. When power down, the PDR threshold voltage can be configured as xxx(TBD). The RESET service routine vector is fixed at address 0x0000_0004 in the memory map.

Debug logic and RTC can be only reset by POR/PDR.

A system reset is generated by the following events:

- Power on reset
- External reset pin
- Watchdog timer reset
- The SYSRESETREQ bit in Cortex-M0 AIRCR register is set as soft reset.
- CPU lockup (when CPU is in HardFault or NMI handler, another HardFault event occurs)

5.3 Clock Control

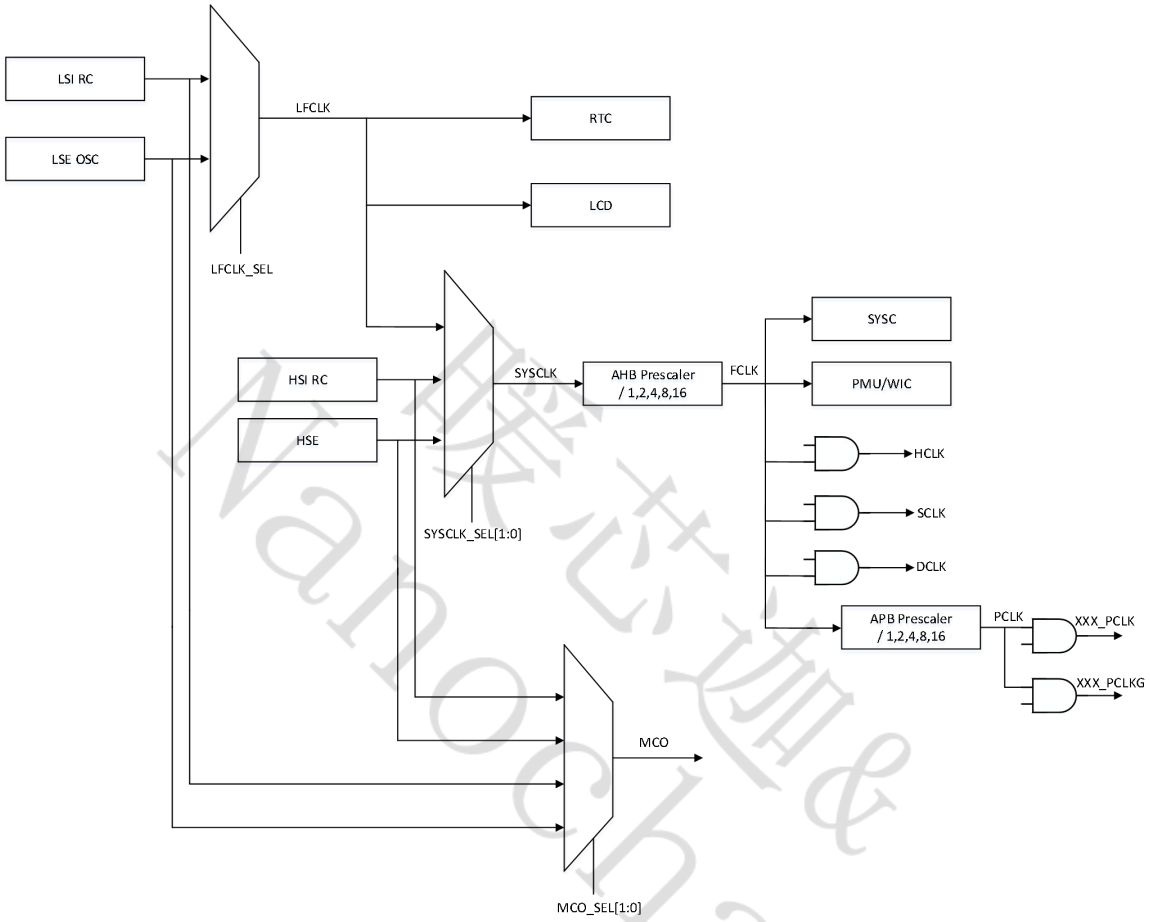
5.3.1 Clock Tree Structure

Four different clock sources can be used:

- **HSI RC** - a high-speed fully-integrated RC oscillator producing HSI clock (about 4-32MHz)
- **HSE -** a high-speed external clock (up to 32MHz)
- **LSI RC** - a low-speed fully integrated RC oscillator producing LSI clock (about 32KHz)
- **LSE OSC** - a low-speed oscillator with external crystal resonator (accurate 32.768KHz)

The clock tree structure as below:

Figure 4 Clock Tree



One of the following clocks can be selected as system clock (SYSCLK):

- LFCLK (from LSI or LSE)
- HSI
- HSE

The system clock maximum frequency is 32MHz. Upon system reset, the HSI clock is selected as system clock.

A switch from one clock source to another occurs only if the target clock source is ready.

LFCLK can be selected from LSI or LSE. LSI and LSE clock can be disabled by software.

For LSE oscillator, at normal operation, set LSE_OSCEN and LSE_CLKEN to 1. At stand-by condition, set LSE_CLKEN to 0. The clock output at XTALOUTCORE can be disabled, while keep the oscillator core running. If there is a need to drive a clock signal into the chip via the input pin for testing purpose, a bypass multiplexer is included

in the cell which routes clock signals driven at INVINPAD directly to XTALOUTCORE. To enable this feature, set OSCEN to 0, which also turns off the oscillator core.

5.3.2 Clock Description in Different Power Modes

AHB and APB peripheral clocks can be disabled by software.

Sleep and low-power sleep modes stop the HCLK.

Stop modes stop all clocks and HSI RC is disabled except LSI and LSE, when leaving the Stop mode, HSI becomes automatically the system clock.

The MCO pins output, independently of each other, the clock selected from: LSI/LSE/HSI/HSE.

Below table shows the clock status (on/off/user) in each low-power mode.

Table 9 Clock description in different power modes

Clocks	Run	LP run	Sleep	LP sleep	Stop	Description
LSI	User	On	User	User	User	32KHz internal RC oscillator
LSE		Selection depends on user				32.768KHz crystal oscillator
LFCLK						Switched low frequency Clock from LSI and LSE
HSI	On	User	On	On	Off	HSI RC 4-32MHz
HSE	Selection depends on user	User	Selection depends on user	Selection depends on user	User	Up to 32MHz external clock
SYSCLK	On	On	On	On	Off	Switched system clock from HSI, HSE and LFCLK
FCLK	On	On	On	On	Off	AHB Free-running clock divided from SYSCLK
HCLK	On	On	Off	Off	Off	AHB bus clock, which is used for CPU/MTP/SRAM/GPIO
SCLK	On	On	On	On	Off	CPU core system clock, which is used for NVIC

DCLK	Dynamic	Dynamic	Dynamic	Dynamic	Off	CPU core debug clock, when there is no debug request, it's gated
PCLK	On	On	On	On	Off	APB free-running clock
PCLKG	Dynamic	Dynamic	Off	Off	Off	APB gated bus clock, when there is no APB cycle, it's gated

5.3.3 Clock Switch Sequence

When switching SYSCLK from HSI to HSE, configure as below sequence

1) Make sure HSE clock is running normally
2) Set SYSCLK_SEL to 2'b01
3) Polling SYSCLK_SWSTS to be 2'b01
4) Set HSI_EN to 0

When switching SYSCLK from HSI to LFCLK, configure as below sequence

1) Make sure LFCLK is running normally, if LFCLK source need to be changed, do the LFCLK switch sequence firstly
2) Set SYSCLK_SEL to 2'b1x
3) Polling SYSCLK_SWSTS to be 2'b1x
4) Set HSI_EN to 0

When switching LFCLK from LSI to LSE, configure as below sequence

5) Set both LSE_OSCEN and LSE_CLKEN to 1
6) Wait LSE stable (stable time to be confirmed)
7) Set LFCLK_SEL to 1
8) Polling LFCLK_SWSTS to be 1
9) Set LSI_EN to 0

5.4 Registers

Table 10 SYSCTRL Registers

Offset	Acronym	Register Description
00h	CLK_CFG	Clock Configuration Register
04h	HSI_CTRL	HSI Control Register
08h	LSI_CTRL	LSI Control Register
0Ch	LSE_CTRL	LSE Control Register
10h	AHB_CLKEN	AHB Peripheral Clock Enable Register
14h	APB_CLKEN	APB Peripheral Clock Enable Register
18h	PERI_CLKEN	Peripheral Working Clock Enable Register
1Ch	SLP_PCLKEN	APB Peripheral Clock Enable in Sleep/Stop mode Register
20h	RST_CTRL	Reset Control Register
24h	RST_FLAG	Reset Flag Register
28h	PRST_KEY	Peripheral Reset Enable Key Register
2Ch	AHB_RST	AHB Peripheral Reset Register
30h	APB_RST	APB Peripheral Reset Register
34h	SYS_CFG	System Configuration Register
38h	PMU_CTRL	PMU Control Register

Offset Address: 03-00h

Clock Configuration Register

Bit	Field Name	Attribute	Default	Field Description
31:18	-	RO	0	Reserved
17:16	MCO_SEL	RW	0	MCU clock output selection 00: HSI 01: HSE 10: LSI 11: LSE
15	-	RO	0	Reserved
14:12	APB_PRES	RW	0	APB clock prescaler 0xx: PCLK not divided 100: PCLK divided by 2 101: PCLK divided by 4 110: PCLK divided by 8 111: PCLK divided by 16

11	-	RO	0	Reserved
10:8	AHB_PRES	RW	0	AHB clock prescaler 0xx: HCLK not divided 100: HCLK divided by 2 101: HCLK divided by 4 110: HCLK divided by 8 111: HCLK divided by 16
7:6	-	RO	0	Reserved
5	LFCLK_SWSTS	RO	0	LFCLK switch status 0: LSI as LFCLK 1: LSE as LFCLK
4	LFCLK_SEL	RW	0	LFCLK switch selection 0: LSI as LFCLK 1: LSE as LFCLK
3:2	SYSCLK_SWSTS	RO	0	System clock switch status 00: RCHF as system clock 01: EXTHF as system clock 1x: LFCLK as system clock
1:0	SYSCLK_SEL	RW	0	System clock switch selection 00: HSI as system clock 01: HSE as system clock 1x: LFCLK as system clock The setting is forced by hardware to 00 (HSI selected) when MCU exits Stop mode.

Offset Address: 07-04h

HSI Control Register

Bit	Field Name	Attribute	Default	Field Description
31:6	-	RO	0	Reserved
5:4	HSI_FREC	RW	01b	HSI OSC frequency selection 00: 4MHz 01: 8MHz 10: 16MHz 11: 32MHz
3:1	-	RO	0	Reserved
0	HSI_EN	RW	1	HSI OSC clock enable Set and cleared by software. Cleared by hardware to stop HSI OSC when entering stop mode. Forced by hardware to keep the HSI OSC ON when it is used directly or indirectly as system clock (when leaving stop mode) 0: HSI OSC OFF 1: HSI OSC ON

Offset Address: 0B-08h

LSI Control Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	LSI_EN	RW	1	LSI enable bit 0: disabled 1: enabled

Offset Address: 0F-0Ch

LSE Control Register

Bit	Field Name	Attribute	Default	Field Description
31:29	-	RO	0	Reserved
1	LSE_CLKEN	RW	0	LSE clock enable bit 0: clock output disabled 1: clock output enabled
0	LSE_OSCEN	RW	0	LSE oscillator enable bit 0: turn off the oscillator core 1: turn on the oscillator core

Offset Address: 13-10h

AHB Peripheral Clock Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:4	-	RO	0	Reserved
2:0	AHB_PORT_CLKEN	RW	111b	Bit2: MTP register access HCLK enable bit Bit1: EXTI HCLK enable bit Bit0: GPIO HCLK enable bit

Offset Address: 17-14h

APB Peripheral Clock Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:0	APB_PORT_CLKEN	RW	0	Bit15: RTC PCLK enable bit Bit14: Analog control PCLK enable bit Bit13: ADC control PCLK enable bit Bit12: Waveform generator PCLK enable bit Bit11: LCD drive PCLK enable bit Bit10: Dual-timers PCLK enable bit Bit9: Timer 1 PCLK enable bit

				Bit8: Timer 0 PCLK enable bit Bit7: PWM PCLK enable bit Bit6: WDT PCLK enable bit Bit5: I2C1 PCLK enable bit Bit4: I2C0 PCLK enable bit Bit3: SPI1 PCLK enable bit Bit2: SPI0 PCLK enable bit Bit1: UART1 PCLK enable bit Bit0: UART0 PCLK enable bit

Offset Address: 1B-18h

Peripheral Working Clock Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:2	-	RO	0	Reserved
1	LCD_CLKEN	RW	0	LCD working clock enable 0: disabled 1: enabled
0	RTC_CLKEN	RW	0	RTC working clock enable 0: disabled 1: enabled

Offset Address: 1F-1Ch

APB Peripheral Clock Enable in Sleep/Stop mode Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:0	APB_SLP_CLKEN	RW	0xFFFF	Bit15: RTC PCLK enable during sleep and stop mode Bit14: Analog control PCLK enable during sleep and stop mode Bit13: ADC control PCLK enable during sleep and stop mode Bit12: Waveform generator PCLK enable during sleep and stop mode Bit11: LCD drive PCLK enable during sleep and stop mode

				Bit10: Dual-timers PCLK enable during sleep and stop mode Bit9: Timer 1 PCLK enable during sleep and stop mode Bit8: Timer 0 PCLK enable during sleep and stop mode Bit7: PWM PCLK enable during sleep and stop mode Bit6: WDT PCLK enable during sleep and stop mode Bit5: I2C1 PCLK enable during sleep and stop mode Bit4: I2C0 PCLK enable during sleep and stop mode Bit3: SPI1 PCLK enable during sleep and stop mode Bit2: SPI0 PCLK enable during sleep and stop mode Bit1: UART1 PCLK enable during sleep and stop mode Bit0: UART0 PCLK enable during sleep and stop mode

Offset Address: 23-20h

Reset Control Register

Bit	Field Name	Attribute	Default	Field Description
31:30	-	RO	0	Reserved
0	LOCKUP_RESETEN	RW	0	CPU Lockup reset enable bit 0: disabled 1: enabled

Offset Address: 27-24h

Reset Flag Register

Bit	Field Name	Attribute	Default	Field Description
31:5	-	RO	0	Reserved
4	PINRST_FLAG	RW1C	0	Reset pin flag This bit is set by a reset from NRST pin occurs It is cleared by writing to this bit, or by a POR 0: No reset pin low-level occurred 1: reset pin low-level occurred
3	LOCKUPRST_FLA	RW1C	0	Lockup reset flag

	G			This bit is set by hardware when a lockup reset occurs It is cleared by writing 1 to this bit, or by a POR 0: No lockup reset occurred 1: Lockup reset occurred
2	WDTRST_FLAG	RW1C	0	Watchdog reset flag This bit i set by hardware when a watchdog reset occurs It is cleared by writing 1 to this bit, or by a POR 0: No watchdog reset occurred 1: Watchdog reset occurred
1	SOFTRST_FLAG	RW1C	0	Software reset flag This bit is set by hardware when a software writing to SYSRESETREQ in NVIC space occurs It is cleared by writing 1 to this bit, or by a POR 0: No software reset occurred 1: Software reset occurred
0	PORRST_FLAG	RW1C	1	POR/PDR reset flag This bit is set by hardware when a POR/PDR reset occurs It is cleared by writing 1 to this bit 0: No POR/PDR reset occurred 1: POR/PDR reset occurred

Offset Address: 2B-28h

Peripheral Reset Enable Key Register

Bit	Field Name	Attribute	Default	Field Description
31:0	PRST_KEY	WO	0	Peripheral reset enable key register Writing 0x1A2B_3C4D enables the peripheral soft reset function, writing other values disables the peripheral soft reset function Write-only, reading this register returns 0.

Offset Address: 2F-2Ch

AHB Peripheral Reset Register

Bit	Field Name	Attribute	Default	Field Description
1	-	RO	0	Reserved
2:0	AHB_PORT_RST	RW	0	Active-high Bit2: MTP register access reset bit Bit1: EXTI reset bit Bit0: GPIO reset bit.

Offset Address: 33-30h

APB Peripheral Reset Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:0	APB_PORT_RST	RW	0	Active-high Bit15: RTC reset bit Bit14: Analog control reset bit Bit13: ADC control reset bit Bit12: Waveform generate reset bit Bit11: LCD drive reset bit Bit10: Dual-timers reset bit Bit9: Timer 0 reset bit Bit8: Timer 1 reset bit Bit7: PWM reset bit Bit6: WDT reset bit Bit5: I2C1 reset bit Bit4: I2C0 reset bit Bit3: SPI1 reset bit Bit2: SPI0 reset bit Bit1: UART1 reset bit

				Bit0: UART0 reset bit

Offset Address: 37-34h

System Configuration Register

Bit	Field Name	Attribute	Default	Field Description
31:10	-	RO	0	Reserved
9:8	BOOT_MODE	RO	0	Boot mode selected by boot pins status bits This bits are read-only. They indicate the boot mode selected by the BOOT[1:0] pins. 00: MTP main array at base address boot mode 01: MTP high 4KBytes boot mode 1x: Embedded SRAM boot mode
7:2	-	RO	0	Reserved
1:0	REMAP	RW	0	Memory mapping selection bits These bits are set and cleared by software. This bit controls the memory's internal mapping at address 0x0000_0000. After reset these bits take on the memory mapping selected by the BOOT pins 00: MTP main array at base address mapped at 0x0000_0000 01: MEP high 4KBytes mapped at 0x0000_0000 1x: SRAM mapped at 0x0000_0000

Offset Address: 3B-38h

PMU Control Register

Bit	Field Name	Attribute	Default	Field Description
31:6	-	RO	0	Reserved
5:4	WKUP_DLY	RW	0	Delay time when waking from stop mode 00: 1us 01: 2us 10: 4us 11: 8us
3:1	-	RO	0	Reserved

0	WKFREQ_SEL	RW	0	HSI frequency select when waking from stop mode 0: Same as before entering stop mode 1: 8MHz

6 CPU

6.1 Overview

The Cortex-M0 processor is an entry-level 32-bit ARM Cortex processor designed for a broad range of embedded applications. It is built on a highly area and power optimized 32-bit processor core, with a 3-stage pipeline von Neumann architecture. The Cortex-M0 processor implements the ARMv6-M architecture, which is based on the 16-bit Thumb instruction set and includes Thumb-2 technology with high code density.

6.2 Processor Configuration

Table 11 Processor Configuration

Feature	Options	Configuration
Interrupts	1~32	32
Data endianness	Little/big	Little
SysTick Timer	Present or absent	Present
Watchpoints	0~2	2
Breakpoints	0~4	4
Debug	Present or absent	Present
JTAGnSW	JTAG or SWD	SWD
Multiplier	Fast or small	Fast (single cycle)
Reset-all-register	Yes or no	Yes
WIC	Present or absent	Present

6.3 Core Registers

Table 12 Core Registers

Name	Type	Description
R0-R12	RW	General purpose registers
SP (R13)	RW	Stack Pointer, SP can be selected as MSP or PSP by configuring CONTROL registers Handler mode: MSP (Main Stack Pointer) Thread mode: PSP (Process Stack Pointer)

LR (R14)	RW	Link Register, it stores the return information for subroutines, function calls, and exceptions
PC (R15)	RW	Program Counter, it contains the current program address
APSR	RW	Application Program Status Register, it contains the current state of the condition flags
IESR	RO	Interrupt Program Status Register, it contains the exception number of the current ISR
EPSR	RO	Execution Program Status Register, it contains the Thumb state bit
PRIMASK	RW	Priority Mask Register, it prevents activation of all exceptions with configurable priority
CONTROL	RW	Control the stack used when the processor is in Thread mode

6.4 Exceptions and Interrupts

The system exceptions and interrupts are managed by NVIC. The interrupt management function is controlled by a number of programmable registers located within the SCS connected PPB bus. The NVIC:

- Support 32 interrupts and 1 NMI
- Flexible interrupt management (Tail-chaining and Late-arriving)
- Handles nested interrupts
- Vectored exception entry
- Interrupt masking
- Four interrupt priority levels

When an exception is accepted, the R0-R3, R12, R14, PC and xPSR are pushed to the current stack memory automatically, LR is updated to EXC_RETURN which is used during exception return.

At the end of exception handling process, if there is no any other exception to be serviced, the register values previously stored on the stack memory are restored and the interrupted program is resumed.

If CPU is in HardFault or NMI handling process, another HardFault occurs, CPU will enter Lockup state, CPU core will be reset, if LOCKUPRESET is set, allow the system to be reset.

Table 13 Interrupt Vector Table

Position	IRQ number	Exception type	Priority	Vector address
0	-	Initial SP value	-	0x0000_0000
1	-	Reset	-3, the highest	0x0000_0004
2	-14	NMI (WDT)	-2	0x0000_0008
3	-13	Hardfault	-1	0x0000_000C
4-10	-	Reserved	-	-
11	-5	SVC	Configurable	0x0000_002C
12-13	-	Reserved	-	-
14	-2	PendSV	Configurable	0x0000_0038
15	-1	SysTick	Configurable	0x0000_003C
16	0	LVD (EXTI line 24)	Configurable	0x0000_0040
17	1	RTC	Configurable	0x0000_0044
18	2	COMP0 (EXTI line 25)	Configurable	0x0000_0048
19	3	COMP1 (EXTI line 26)	Configurable	0x0000_004C
20	4	GPIO0_7 (EXTI line 0-7)	Configurable	0x0000_0050
21	5	GPIO8_15 (EXTI line 8-15)	Configurable	0x0000_0054
22	6	GPIO16_23 (EXTI line 16-23)	Configurable	0x0000_0058
23	7	MTP	Configurable	0x0000_005C
24	8	Charger_ok(EXTI line 27)	Configurable	0x0000_0060
25	9	Charger_end(EXTI line 28)	Configurable	0x0000_0064
26	10	ADC	Configurable	0x0000_0068
27	11	LCD	Configurable	0x0000_006C
28	12	UART0	Configurable	0x0000_0070
29	13	UART1	Configurable	0x0000_0074
30	14	SPI0	Configurable	0x0000_0078
31	15	SPI1	Configurable	0x0000_007C

32	16	I2C0 Event	Configurable	0x0000_0080
33	17	I2C0 Error	Configurable	0x0000_0084
34	18	I2C1 Event	Configurable	0x0000_0088
35	19	I2C1 Error	Configurable	0x0000_008C
36	20	PWM	Configurable	0x0000_0090
37	21	Basic TIM0	Configurable	0x0000_0094
38	22	Basic TIM1	Configurable	0x0000_0098
39	23	Dual TIM	Configurable	0x0000_009C
40	24	Over temperature (EXTI 29)	Configurable	0x0000_00A0
41	25	-	Configurable	0x0000_00A4
42	26	-	Configurable	0x0000_00A8
43	27	-	Configurable	0x0000_00AC
44	28	-	Configurable	0x0000_00B0
45	29	-	Configurable	0x0000_00B4
46	30	-	Configurable	0x0000_00B8
47	31	-	Configurable	0x0000_00BC

6.5 Debug

Basic debug functionality includes:

- Processor halt, single-step
- Processor core register access
- Four hardware breakpoints
- Two watchpoints
- Unlimited software breakpoints (BKPT instruction)
- Full system memory access
- SWD Interface

7 Cortex-M0 Peripherals

7.1 Overview

The address map of the *Private peripheral bus* (PPB) is:

Table 14 Core Peripheral Register Regions

Address	Core peripheral
0xE000E008-0xE000E00F 0xE000ED00-0xE000ED3F	System Control Block
0xE000E010-0xE000E01F	SysTick Timer
0xE000E100-0xE000E4EF 0xE000EF00-0xE000EF03	Nested Vectored Interrupt Controller

7.2 Nested Vectored Interrupt Controller

This section describes the *Nested Vectored Interrupt Controller* (NVIC) and the registers it uses. The NVIC supports:

- 1 to 32 interrupts.
- A programmable priority level of 0-192 in steps of 64 for each interrupt. A higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Level and pulse detection of interrupt signals.
- Interrupt tail-chaining
- An external *Non-maskable interrupt* (NMI).

The processor automatically stacks its state on exception entry and unstacks this state on exception exit, with no instruction overhead. This provides low latency exception handling.

7.2.1 Registers

Table 15 NVIC Registers

Address	Acronym	Type	Reset value	Register Description
0xE000E100	ISER	RW	0x00000000	Interrupt Set-enable Register
0xE000E180	ICER	RW	0x00000000	Interrupt Clear-enable Register
0xE000E200	ISPR	RW	0x00000000	Interrupt Set-pending Register
0xE000E280	ICPR	RW	0x00000000	Interrupt Clear-pending Register
0xE000E400- 0xE000E41C	IPR0-7	RW	0x00000000	Interrupt Priority Register

Address: 0xE000E100

Interrupt Set-enable Register

Bit	Field Name	Attribute	Default	Field Description
31:0	SETENA	RW	0	Interrupt set-enable bits Write: 0 = no effect 1 = enable interrupt. Read: 0 = interrupt disabled 1 = interrupt enabled

Address: 0xE000E180

Interrupt Set-enable Register

Bit	Field Name	Attribute	Default	Field Description
31:0	CLRENA	RW	0	Interrupt clear-enable bits Write: 0 = no effect 1 = disable interrupt. Read: 0 = interrupt disabled 1 = interrupt enabled

Address: 0xE000E200

Interrupt Clear-pending Register

Bit	Field Name	Attribute	Default	Field Description
31:0	CLRPEND	RW	0	Interrupt clear-pending bits Write: 0 = no effect 1 = removes pending state an interrupt.

				Read: 0 = interrupt is not pending 1 = interrupt is pending Writing 1 to an ICPR bit does not affect the active state of the corresponding interrupt

Address: 0xE000E280

Interrupt Set-pending Register

Bit	Field Name	Attribute	Default	Field Description
31:0	SETPEND	RW	0	Interrupt set-pending bits Write: 0 = no effect 1 = changes the interrupt state to pending. Read: 0 = interrupt is not pending 1 = interrupt is pending Writing 1 to the ISPR bit corresponding to: ● An interrupt that is pending has no effect ● A disabled interrupt sets the state of that interrupt to pending.

Address: 0xE000E400-0xE000E41C

Interrupt Priority Register

Bit	Field Name	Attribute	Default	Field Description
31:24	PRI_(4n+3)	RW	0	Each priority filed holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[7:6] of each field, bits [5:0] read as zero and ignore writes. This means writing 255 to a priority register saves value to 192 to the register.
23:16	PRI_(4n+2)	RW	0	
15:8	PRI_(4n+1)	RW	0	
7:0	PRI_(4n)	RW	0	

7.3 System Control Block

The *System Control Block* (SCB) provides system implementation information, and system control. This includes configuration, control, and reporting of the system exceptions.

Table 16 SCB Registers

Address	Acronym	Type	Reset value	Register Description
0xE000ED00	CPUID	RO	0x410CC200	CPUID Register
0xE000ED04	ICSR	RW	0x00000000	Interrupt Control and State Register
0xE000ED0C	AIRSR	RW	0xFA050000	Application Interrupt and Reset Control Register
0xE000ED10	SCR	RW	0x00000000	System Control Register
0xE000ED14	CCR	RO	0x00000204	Configuration and Control Register
0xE000ED1C	SHPR2	RW	0x00000000	System Handler Priority Register 2
0xE000ED20	SHPR3	RW	0x00000000	System Handler Priority Register 3

Address: 0xE000ED00

CPUID Register

Bit	Field Name	Attribute	Default	Field Description
31:24	Implementer	RO	0x41	Implementer code: 0x41 = ARM
23:20	Variant	RO	0	Variant number: 0x0 = Revision 0
19:16	Constant	RO	0xC	Constant that defines the architecture of the processor: 0xC = ARMv6-M architecture
15:4	Partno	RO	0xC20	Part number of the processor: 0xC20 = Cortex-M0
3:0	Revision	RO	0	Revision number: 0x0 = Patch 0

Address: 0xE000ED04

Interrupt Control and State Register

Bit	Field Name	Attribute	Default	Field Description
31	NMIPENDSET	RW	0	NMI set-pending bit.

				Write: 0 = no effect 1 = changes NMI exception state to pending. Read: 0 = NMI exception is not pending 1 = NMI exception is pending Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor ix executing that handler.
30:29	-	RO	0	Reserved
28	PENDSVSET	RW	0	PendSV set-pending bit. Write: 0 = no effect 1 = changes PendSV exception state to pending. Read: 0 = PendSV exception is not pending 1 = PendSV exception is pending Writing 1 to this bit is the only way to set the PendSV exception state to pending.
27	PENDSVCLR	WO	0	PendSV clear-pending bit. Write: 0 = no effect 1 = removes the pending state from the PendSV exception.
26	PENDSTSET	RW	0	SysTick exception set-pending bit. Write: 0 = no effect 1 = changes sysTick exception state to pending. Read: 0 = SysTick exception is not pending

				1 = SysTick exception is pending.
25	PENDSTCLR	WO	0	SysTick exception clear-pending bit. Write: 0 = no effect 1 = removes the pending state from the SysTick exception.
24:23	-	RO	0	Reserved
22	ISRPENDING	RO	0	Interrupt pending flag, excluding NMI and Faults: 0 = interrupt not pending 1 = interrupt pending
21:18	-	RO	0	Reserved
17:12	VECTPENDING	RO	0	Indicates the exception number of the highest priority pending enabled exception: 0 = no pending exceptions Nonzero = the exception number of the highest priority pending enabled exception.
11:6	-	RO	0	Reserved
5:0	VECTACTIVE	RO	0	Contains the active exception number: 0 = Thread mode Nonzero = The exception number of the currently active exception

Address: 0xE000ED0C

Application Interrupt and Reset Control Register

Bit	Field Name	Attribute	Default	Field Description
31:16	VECTKEY	RW	0xFA05	Register key On writes, write 0x05FA to VECTKEY, otherwise the write is ignored Read as 0xFA05
15	ENDIANESS	RO	0	Data endianness implemented: 0 = Little-endian 1 = Big-endian
14:3	-	RO	0	Reserved

2	SYSRESETREQ	WO	0	System reset request: 0 = no effect 1 = requests a system level reset. This bit reads as 0.
1	VECTCLRACTIVE	WO	0	Reserved for debug use. This bit reads as 0. When writing to the register you must write 0 to this bit, otherwise behavior is unpredictable.
0	-	RO	0	Reserved

Address: 0xE000ED10

System Control Register

Bit	Field Name	Attribute	Default	Field Description
31:5	-	RO	0	Reserved
4	SEVONPEND	RW	0	Send Event on Pending bit: 0 = only enabled interrupts or events can wakeup the processor, disabled interrupts are excluded 1 = enabled events and all interrupts, including disabled interrupts, can wakeup the processor. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction or an external event.
3	-	RO	0	Reserved
2	SLEEPDEEP	RW	0	Controls whether the processor uses sleep or deep sleep as its low power mode: 0 = sleep 1 = deep sleep.
1	SLEEPONEXIT	RW	0	Indicates sleep-on-exit when returning from Handler mode to Thread mode: 0 = do not sleep when returning to Thread mode. 1 = enter sleep, or deep sleep, on return from an ISR to Thread mode.

				Setting this bit to 1enables an interrupt driven application to avoid returning to an empty main application.
0	-	RO	0	Reserved

Address: 0xE000ED14

Configuration and Control Register

Bit	Field Name	Attribute	Default	Field Description
31:10	-	RO	0	Reserved
9	STKALIGN	RO	1	Always reads as one, indicates 8-byte stack alignment on exception entry. On exception entry, the processor uses bit[9] of the stacked PSR to indicate the stack alignment. On return from the exception it uses this stacked bit to restore the correct stack alignment.
8:4	-	RO	0	Reserved
3	UNALIGN_TRP	RO	1	Always reads as one, indicates that all unaligned accesses generate a Hardfault.
2:0	-	RO	0	Reserved.

Address: 0xE000ED1C

System Handler Priority Register 2

Bit	Field Name	Attribute	Default	Field Description
31:24	PRI_11	RW	0	Priority of system handler 11, SVCALL
23:0	-	RO	0	Reserved

Address: 0xE000ED20

System Handler Priority Register 3

Bit	Field Name	Attribute	Default	Field Description
31:24	PRI_15	RW	0	Priority of system handler 15, SysTick exception
23:16	PRI_14	RW	0	Priority of system handler 14, PendSV
15:0	-	RO	0	Reserved

7.4 SysTick Timer

When enabled, the timer counts down from the reload value to zero, reloads (wraps to) the value in the SYST_RVR on the next clock cycle, then decrements on subsequent clock cycles. Writing a value of zero to the SYST_RVR disables the counter on the next wrap. When the counter transitions to zero, the COUNTFLAG status bit is set to 1. Reading SYST_CSR clears the COUNTFLAG bit to 0.When the processor is halted for debugging the counter does not decrement.

Software can configure the SysTick timer to select SCLK as its clock source, or an alternative clock source: FCLK clock divided by 8.

7.4.1 Registers

Table 17 SysTick Timer Registers

Address	Acronym	Type	Reset value	Register Description
0xE000E010	SYST_CSR	RO	0x00000000	SysTick Control and Status Register
0xE000E014	SYST_RVR	RW	0x00000000	SysTick Reload Value Register
0xE000E018	SYST_CVR	RW	0x00000000	SysTick Current Value Register
0xE000E01C	SYST_CALIB	RW	0x00000000	SysTick Calibration Value Register

Address: 0xE000E010

SysTick Control and Status Register

Bit	Field Name	Attribute	Default	Field Description
31:17	-	RO	0	Reserved
16	COUNTFLAG	RO	0	Returns 1 if timer counted to 0 since the last read of this register
15:3	-	RO	0	Reserved
2	CLKSOURCE	RW	0	Selects the SysTick timer source: 0 = external reference clock 1 = processor clock
1	TICKINT	RW	0	Enables SysTick exception request: 0 = counting down to zero does not assert the SysTick exception request 1 = counting down to zero to asserts the SysTick exception request
0	ENABLE	RW	0	Enables the counter: 0 = counter disabled 1 = counter enabled

Address: 0xE000E014

SysTick Control and Status Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	RELOAD	RW	0	Value to load into the SYST_CVR when counter is enabled and when it reaches 0. The RELOAD value can be any value in the range 0x00000001-0x00FFFFFF. You can program a value of 0, but this has no effect because the SysTick exception request and COUNTFLAG are activated when counting from 1 to 0. To generate a multi-shot timer with a period of N processor clock cycles, use a RELOAD value of N-1. For example, if the SysTick interrupt is required every 100 clock pulses, set RELOAD to 99.

Address: 0xE000E018

SysTick Current Value Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	CURRENT	RW	0	Reads return the current value of the SysTick counter. A write of any value clears the field o 0, and also clears the SYST_CSR.COUNTFLAG bit to 0.

Address: 0xE000E01C

SysTick Calibration Value Register

Bit	Field Name	Attribute	Default	Field Description
31	NOREF	RO	0	Reads as zero. Indicates that separate reference clock is provided.
30	SKEW	RO	1	Reads as one. Calibration value for the 10ms is inexact
29:24	-	RO	0	Reserved
23:0	TENMS	RO	0	Calibration value is not available

8 General Purpose I/Os (GPIO)

8.1 Overview

The GPIO block implements an AHB-Lite interfaced General Purpose I/O block with 24 I/Os where each can be independently configured via memory mapped registers. All registers are readable and writeable through the AHB interface.

Actual control signal functions and count are subject to change depending on the chosen third-party IOPAD model.

8.1.1 Feature List

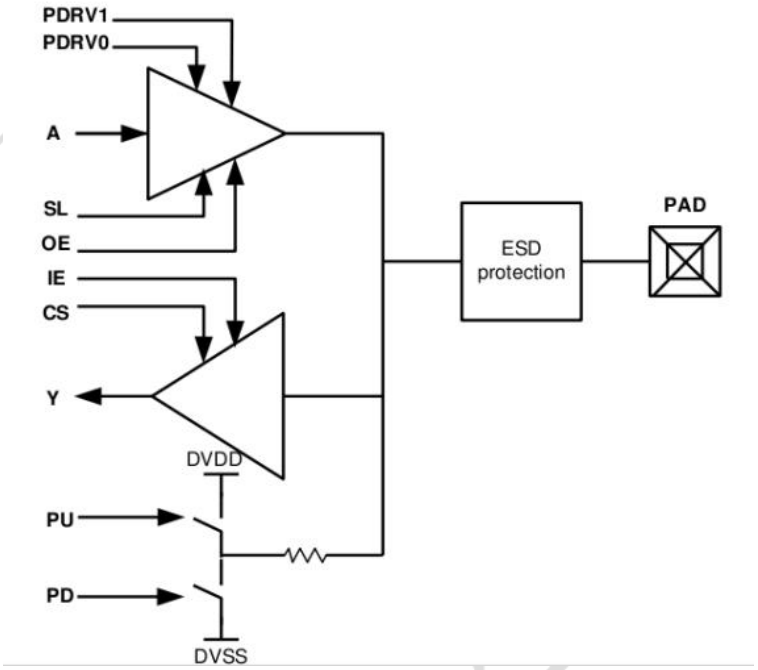
- Input/Output direction control
- Each pin weak pull-up/pull-down
- Output push-pull/open drain enable control
- Output bit set/reset control to support atomic read/modify write operations
- Output drive strength select
- Input buffer select (CMOS/Schmitt)
- Slew rate select
- Analog channel enable
- Alternate function
- Two HCLK cycle toggle output capability

8.2 Function Description

8.2.1 Tri-state Digital I/O Cell

The I/O pad is a 5V tri-state bi-directional I/O pad with programmable output drive strengths of 4mA, 8mA, 12mA and 16mA. It can be programmed as CMOS input or Schmitt trigger input and with or without pull-up/down. In the fast slew 16mA drive strength mode, it can runs at a frequency of 100MHz with 30pF capacitive load. The use of the control pins PDRV0, PDRV1, SL, CS is as shown in register table.

Figure 5 Function Schematic of I/O Cell



The logical operation of the cell is as shown below.

Table 18 Logical Operation Table

Driver Function

Input				Output
OE	PU	PD	A	PAD
0	0	0	X	Hi-Z
0	0	1	X	weak 0
0	1	0	X	weak 1
0	1	1	X	Hi-Z
1	X	X	0	0
1	X	X	1	1

Receiver Function

Input				Output
IE	PU	PD	PAD	Y
0	X	X	X	0
1	X	X	0	0
1	X	X	1	1
1	0	1	weak 0	0
1	1	0	weak 1	1

8.2.2 Alternate Function Multiplexer

The device I/O pins are connected to on-board peripherals/modules through a multiplexer that allows only one peripheral alternate function connected to an I/O pin at a time. In this way, there can be no conflict between peripherals available on same I/O pin.

Each I/O has 4 alternate functions by configuring GPIO_ALTF registers. When GPIOx_ALTF[1:0] is configured as 2'b11, the pad is used for analog function. Detail alternate function mapping information is shown in register table.

After reset the multiplexer selection is alternate function 0. GPIO0/1 default are as debug pins: SWCLK and SWDIO. SWCLK is pull-down and SWDIO is pull-up after reset.

8.2.3 Input Configuration

When GPIO pin is configured as input by configuring GPIO_IE registers:

- The schmitt trigger or CMOS input can be selected
- The weak pull-up or pull-down resistors can be selected
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- A read access to the input data register provides the I/O state

8.2.4 Output Configuration

When GPIO pin is configured as output by configuring GPIO_OE registers:

- The open-drain mode pull-pull mode can be selected
- The weak pull-up or pull-down resistors can be selected
- The output slew rate can be configured as fast or slow

- The output drive strength can be configured as 4mA/8mA/12mA/16mA

8.2.5 Analog Function

When GPIO pin is configured as analog function:

- Output is disabled
- Input is disabled
- Pull-up and pull-down is disabled
- A read access to the input data register returns 0
- If IO is used for multiple analog function, only one function is enabled simultaneously

8.2.6 I/O Data Bitwise Handling

To each bit in GPIO_DATAOUT, correspond two control bits in GPIO_BITSET and GPIO_BITCLR registers. When write 1 to GPIO_BITSET[i], sets the corresponding DATAOUT[i] bit. When write 1 to GPIO_BITCLR[i], resets the corresponding DATAOUT[i] bit.

Writing any bit to 0 in GPIO_BITSET[i] or GPIO_BICLR[i] does not have any effect on the corresponding bit in GPIO_DATAOUT. If there is an attempt to both set and reset a bit in GPIO_BITSET and GPIO_BITCLR, the set action takes priority.

Read from GPIO_DATAOUT, GPIO_BITSET or GPIO_BITCLR all return the current DATAOUT value.

The purpose of bitwise handling is to allow atomic read/modify accesses to any of the GPIO_DATAOUT registers. In this way, there is no risk of an IRQ occuring between the read and the modify access.

8.3 Registers

Table 19 GPIO Registers

Offset	Acronym	Register Description
00h	GPIO_DATAIN	GPIO Datain Register
04h	GPIO_DATAOUT	GPIO Dataout Register
08h	GPIO_BITSET	GPIO Dataout Bit Set Register
0Ch	GPIO_BITCLR	GPIO Dataout Bit Clear Register
10h	GPIO_OE	GPIO Output Enable Register
14h	GPIO_IE	GPIO Input Enable Register
18h	GPIO_PU	GPIO Pull-up Register
1Ch	GPIO_PD	GPIO Pull-down Register
20h	GPIO_CS	GPIO CMOS/Schmitt Input Type Register
24h	GPIO_SL	GPIO Slew Rate Register
28h	GPIO_PDRV0	GPIO Output Drive Strength 0 Register
2Ch	GPIO_PDRV1	GPIO Output Drive Strength 1 Register
30h	GPIO_ODEN	GPIO Open-drain Enable Register
34h	GPIO_ALTFL	GPIO0~15 Alternate Function Select Register
38h	GPIO_ALTFH	GPIO16~24 Alternate Function Select Regiser
3Ch	GPIO_ANAE	GPIO Analog Channel Enable Register

Offset Address: 03-00h

GPIO Datain Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_DATAIN	RO	0	GPIO input data value

Offset Address: 07-04h

GPIO Dataout Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_DATAOUT	RW	0	GPIO output data value

Offset Address: 0B-08h

GPIO Dataout Bit Set Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved

23:0	GPIO_BITSET	RW1S	0	A write of 1 set the output data value; A write of 0 does not have any effect Read from the register return the output data value

Offset Address: 0F-0Ch

GPIO Dataout Bit Clear Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_BITCLR	RW1C	0	A write of 1 clear the output data value; A write of 0 does not have any effect Read from the register return the output data value

Offset Address: 13-10h

GPIO Output Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_OE	RW	0	GPIO output enable bit 0: disabled 1: enabled

Offset Address: 17-14h

GPIO Input Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_IE	RW	0	GPIO input enable bit 0: disabled 1: enabled

Offset Address: 1B-18h

GPIO Pull-up Register

Bit	Field Name	Attribute	Default	Field Description
31:26	-	RO	0	Reserved
25:0	GPIO_PU	RW	0000002h	Bit25: Boot1 pin pull-up enabled Bit24: Boot0 pin pull-up enabled Bit23-0: GPIO23~0 pin pull-up enabled

Offset Address: 1F-1Ch

GPIO Pull-down Register

Bit	Field Name	Attribute	Default	Field Description
31:26	-	RO	0	Reserved
25:0	GPIO_PD	RW	3000001h	Bit25: Boot1 pin pull-down enabled Bit24: Boot0 pin pull-down enabled Bit23-0: GPIO23~0 pin pull-down enabled

Offset Address: 23-20h

GPIO CMOS/Schmitt Input Type Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_CS	RW	0	GPIO input type select 0: CMOS buffer 1: Schmitt trigger

Offset Address: 27-24h

GPIO Slew Rate Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_SL	RW	0	GPIO output slew rate select 0: fast

				1: slow

Offset Address: 2B-28h

GPIO Output Drive Strength 0 Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_PDRV0	RW	0	GPIO output drive strength 0 select, {GPIO_PDRV1, GPIO_PDRV0} 00: 4mA 01: 8mA 10: 12mA 11: 16mA

Offset Address: 2F-2Ch

GPIO Output Drive Strength 1 Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_PDRV1	RW	0	GPIO output drive strength 1 select, {GPIO_PDRV1, GPIO_PDRV0} 00: 4mA 01: 8mA 10: 12mA 11: 16mA

Offset Address: 33-30h

GPIO Open-drain Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_ODEN	RW	0	GPIO output mode select 0: push-pull mode 1: open-drain mode

Offset Address: 37-34h

GPIO0~15 Alternate Function Select Register

Bit	Field Name	Attribute	Default	Field Description
31:30	GPIO15_ALTF	RW	0	GPIO15 alternate function select 00: GPIO15 01: UART1_CTS_N 10: SPI0_NSS3 11: Analog function
29:28	GPIO14_ALTF	RW	0	GPIO14 alternate function select 00: GPIO14 01: UART1_RTS_N 10: SPI0_NSS2 11: Analog function
27:26	GPIO13_ALTF	RW	0	GPIO13 alternate function select 00: GPIO13 01: UART1_TXD 10: SPI0_NSS1 11: Analog function
25:24	GPIO12_ALTF	RW	0	GPIO12 alternate function select 00: GPIO12 01: UART1_RXD 10: MCO 11: Analog function
23:22	GPIO11_ALTF	RW	0	GPIO11 alternate function select 00: GPIO11 01: SPI0_NSS0 10: RTC_1Hz 11: Analog function
21:20	GPIO10_ALTF	RW	0	GPIO10 alternate function select

				00: GPIO10 01: SPI0_MISO 10: Reserved 11: Analog function
19:18	GPIO9_ALTF	RW	0	GPIO9 alternate function select 00: GPIO9 01: SPI0_MOSI 10: I2C1_SDA 11: Analog function
17:16	GPIO8_ALTF	RW	0	GPIO8 alternate function select 00: GPIO8 01: SPI0_SCK 10: I2C1_SCL 11: Analog function
15:14	GPIO7_ALTF	RW	0	GPIO7 alternate function select 00: GPIO7 01: I2C0_SDA 10: Reserved 11: Analog function
13:12	GPIO6_ALTF	RW	0	GPIO6 alternate function select 00: GPIO6 01: I2C0_SCL 10: Reserved 11: Analog function
11:10	GPIO5_ALTF	RW	0	GPIO5 alternate function select 00: GPIO5 01: UART0_CTS_N 10: Reserved 11: Analog function

9:8	GPIO4_ALTF	RW	0	GPIO4 alternate function select 00: GPIO4 01: UART0_RTS_N 10: SPI1_NSS3 11: Analog function
7:6	GPIO3_ALTF	RW	0	GPIO3 alternate function select 00: GPIO3 01: UART0_TXD 10: SPI1_NSS2 11: Analog function
5:4	GPIO2_ALTF	RW	0	GPIO2 alternate function select 00: GPIO2 01: UART0_RXD 10: SPI1_NSS1 11: Analog function
3:2	GPIO1_ALTF	RW	0	GPIO1 alternate function select 00: SWDIO 01: GPIO1 10: Reserved 11: Analog function
1:0	GPIO0_ALTF	RW	0	GPIO0 alternate function select 00: SWCLK 01: GPIO0 10: HSE_CLK 11: Analog function

Offset Address: 3B-38h

GPIO16~23 Alternate Function Select Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:14	GPIO23_ALTF	RW	0	GPIO23 alternate function select 00: GPIO23 01: TIMER1_EXTIN 10: Reserved 11: Analog function
13:12	GPIO22_ALTF	RW	0	GPIO22 alternate function select 00: GPIO22 01: TIMER0_EXTIN 10: Reserved 11: Analog function
11:10	GPIO21_ALTF	RW	0	GPIO21 alternate function select 00: GPIO21 01: PWM_OUT6 10: COMP1_OUT 11: Analog function
9:8	GPIO20_ALTF	RW	0	GPIO20 alternate function select 00: GPIO20 01: PWM_OUT5 10: COMP0_OUT 11: Analog function
7:6	GPIO19_ALTF	RW	0	GPIO19 alternate function select 00: GPIO19 01: PWM_OUT4 10: SPI1_NSS0 11: Analog function

5:4	GPIO18_ALTF	RW	0	GPIO18 alternate function select 00: GPIO18 01: PWM_OUT3 10: SPI1_MISO 11: Analog function
3:2	GPIO17_ALTF	RW	0	GPIO17 alternate function select 00: GPIO17 01: PWM_OUT2 10: SPI1_MOSI 11: Analog function
1:0	GPIO16_ALTF	RW	0	GPIO16 alternate function select 00: GPIO16 01: PWM_OUT1 10: SPI1_SCK 11: Analog function

Offset Address: 3F-3Ch

GPIO Analog Channel Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:0	GPIO_ANAE	RW	0	GPIO analog channel enable bit 0: Analog channel disabled 1: Analog channel enabled

9 Extended Interrupt and Event Controller (EXTI)

9.1 Overview

The Extended interrupt and event controller (EXTI) manages the CPU and system wakeup through configurable and direct event inputs (lines). It provides wakeup requests to the power control, and generates an interrupt request to the CPU NVIC and events to the CPU event input.

The EXTI wakeup requests allow the system to be woken up from stop mode.

The interrupt request and event request generation can also be used in run mode.

9.1.1 Feature List

- System wakeup upon event on any input
- Wakeup flag and CPU interrupt generation for events not having a wakeup flag in their source peripheral
- Configurable events (from I/Os, peripherals not having an associated interrupt pending status bit)
  - Selectable active trigger edge
  - Independent rising and falling edge interrupt pending status bits
  - Individual interrupt and event generation mask, used for conditioning the CPU wakeup, interrupt and event generation
  - SW trigger possibility
- Direct events (from peripherals having an associated flag and interrupt pending status bit)
  - Fixed rising edge active trigger
  - No interrupt pending status bit in the EXTI
  - Individual interrupt and event generation mask for conditioning the CPU wakeup and event generation
  - No SW trigger possibility

9.2 Function Description

9.2.1 EXTI connections

The peripherals able to generate wakeup or interrupt events when the system is in stop mode are connected to the EXTI.

- Peripheral wakeup signals that generate a pulse or that do not have an interrupt status bits in the peripheral, are connected to an EXTI configurable line. For these events the EXTI provides a status pending bit which requires to be cleared. It is the EXTI interrupt associated with the status bit that interrupts the CPU
- Peripheral interrupt and wakeup signals that have a status bit in the peripheral which requires to be cleared in the peripheral, are connected to an EXTI direct line. There is no status pending bit within EXTI. The interrupt or wakeup is cleared by the CPU in the peripheral. It is the peripheral interrupt that interrupts the CPU directly.

The EXTI configurable event interrupts are connected to the NVIC of the CPU.

The dedicated EXTI CPU event is connected to the CPU rxev input.

The EXTI CPU wakeup signals are connected to the PMU block, and are used to wake up the system and CPU sub-system bus clocks.

Table 20 EXTI line connections

EXTI line	Line source	Line type
0~23	GPIO	Configurable
24	LVD output	Configurable
25	COMP0 output	Configurable
26	COMP1 output	Configurable
27	Charge_ok	Configurable
28	Charge_end	Configurable
29	Over temperature	Configurable
30	Rtc_alarm	Direct
31	Rtc_wut	Direct

9.2.2 EXTI configurable event input wakeup

The software interrupt event register allows triggering configurable events by software, writing the corresponding register bit, irrespective of the edge selection setting.

The rising edge and falling edge selection registers allow to enable and select the configurable event active trigger edge or both edges.

The CPU has its dedicated interrupt mask register and a dedicated event mask registers. The enabled event allows generation an event on the CPU. All events for a CPU are OR-ed together into a single CPU event signal. The event pending registers (EXTI_RPR and EXTI_FPR) is not set for an unmasked CPU event.

The configurable events have unique interrupt pending request registers, shared by the CPU. The pending register is only set for an unmasked interrupt. Each configurable event provides a common interrupt to the CPU. The configurable event interrupts need to be acknowledged by software in the EXTI_RPR and/or EXTI_FPR registers.

When a CPU interrupt or CPU event is enabled, the asynchronous edge detection circuit is reset by the clocked delay and rising edge detect pulse generator. This guarantees the wakeup of the EXTI fclk clock before the asynchronous edge detection circuit is reset.

For the configurable event inputs, the software can generate an event request by setting the corresponding bit of the software interrupt/event register EXTI_SWIER, which has the effect of a rising edge on the event input. The pending rising edge event flag is set in the EXTI_RPR register, irrespective of the EXTI_RTSR register setting.

9.2.3 EXTI direct event input wakeup

The direct events do not have an associated EXTI interrupt. The EXTI only wakes up the system and CPU sub-system clocks and may generate a CPU wakeup event. The peripheral synchronous interrupt, associated with the direct wakeup event wakes up the CPU.

The EXTI direct event is able to generate a CPU event. This CPU event wakes up the CPU. The CPU event may occur before the interrupt flag of the associated peripheral is set.

For direct event inputs, when enabled in the associated peripheral, an event request is generated on the rising edge only. There is no corresponding CPU pending bit in the EXTI. When the associated CPU interrupt is unmasked, the corresponding CPU subsystem is woken up. The CPU is woken up (interrupted) by the peripheral synchronous interrupt.

9.3 Registers

Table 21 EXTI Registers

Offset	Acronym	Register Description
00h	EXTI_RTSR	EXTI Rising Trigger Selection Register
04h	EXIT_FTSR	EXTI Falling Trigger Selection Register
08h	EXTI_SWIER	EXTI Software Interrupt Event Register
0Ch	EXTI_RPR	EXTI Rising Edge Pending Register
10h	EXTI_FPR	EXTI Falling Edge Pending Register
14h	EXTI_IMR	EXTI CPU Wakup with Interrupt Mask Register
18h	EXTI_EMR	EXTI CPU Wakeup with Event Mask Register

Offset Address: 03-00h

EXTI Rising Trigger Selection Register

Bit	Field Name	Attribute	Default	Field Description
31:30	-	RO	0	Reserved
29:0	RISE_TRGEN	RW	0	Rising trigger event configuration bit of configurable line x (x = 29 to 0) Each bit enables/disables the rising edge trigger for event and interrupt on the corresponding line. 0: Disable 1: Enable

Offset Address: 07-04h

EXTI Falling Trigger Selection Register

Bit	Field Name	Attribute	Default	Field Description
31:30	-	RO	0	Reserved
29:0	FALL_TRGEN	RW	0	Falling trigger event configuration bit of configurable line x (x = 29 to 0) Each bit enables/disables the falling edge trigger for event and interrupt on the corresponding line. 0: Disable 1: Enable

Offset Address: 0B-08h

EXTI Software Interrupt Event Register

Bit	Field Name	Attribute	Default	Field Description
31:30	-	RO	0	Reserved
29:0	SW_TRGEN	WO	0	Software rising trigger event trigger on line x (x = 29 to 0) Setting of any bit by software triggers a rising edge event on the corresponding line x, resulting in an interrupt, independently of EXTI_RTSR and EXTI_FTSR settings. This bits are automatically cleared by HW.Reading of any bit always returns 0. 0: No effect 1: Rising edge event generated on the corresponding line, follow by an interrupt

Offset Address: 0F-0Ch

EXTI Rising Edge Pending Register

Bit	Field Name	Attribute	Default	Field Description
31:30	-	RO	0	Reserved
29:0	RISE_PEND	RW1C	0	Rising edge event pending for configurable line x (x = 29 to 0) Each bit is set upon a rising edge event generated by hardware or by software on the corresponding line. Each bit is cleared by writing 1 into it. 0: No rising edge trigger request occurred 1: Rising edge trigger request occurred

Offset Address: 13-10h

EXTI Falling Edge Pending Register

Bit	Field Name	Attribute	Default	Field Description
31:30	-	RO	0	Reserved
29:0	FALL_PEND	RW1C	0	Falling edge event pending for configurable line x (x = 29 to 0) Each bit is set upon a rising edge event generated by hardware or by software on the corresponding line. Each bit is cleared by writing 1 into it. 0: No falling edge trigger request occurred

				1: Falling edge trigger request occurred

Offset Address: 17-14h

EXTI CPU Wakeup with Interrupt Mask Register

Bit	Field Name	Attribute	Default	Field Description
31:0	INT_MASK	RW	0	CPU wakeup with interrupt mask on line x (x = 31 to 0) Setting/clearing each bit unmasks/masks the CPU wakeup with interrupt, by an event on the corresponding line. 0: wakeup with interrupt masked 1: wakeup with interrupt unmasked

Offset Address: 1B-18h

EXTI CPU Wakeup with Event Mask Register

Bit	Field Name	Attribute	Default	Field Description
31:0	EVT_MASK	RW	0	CPU wakeup with event generation mask on line x (x = 31 to 0) Setting/clearing each bit unmasks/masks the CPU wakeup with event generation on the corresponding line. 0: wakeup with event generation masked 1: wakeup with event generation unmasked

10 Serial Peripheral Interface (SPI)

10.1 Overview

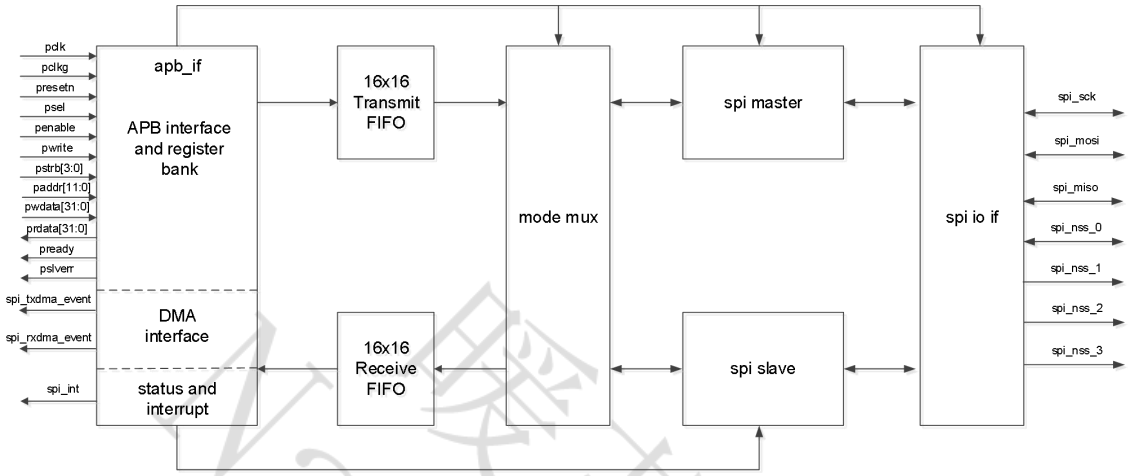
The Serial Peripheral Interface (SPI) is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (4 to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communication between the device and external peripherals. Typical applications include interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, SPI EPROMS, and analog-to-digital converters.

10.1.1 Feature List

- Master or slave mode
- Full-duplex synchronous transfers on three lines
- Half-duplex synchronous transfer on two lines (with bidirectional data line)
- Simplex synchronous transfers on two lines (with unidirectional data line)
- Transmit/Receive 16x16bits FIFOs
- 4-16 bit data size selection
- Master mode baud rate generator up to $F_{pclk}/2$
- Slave mode baud rate generator up to $F_{pclk}/4$
- Multiple slave chip select (nss0~nss3)
- NSS management by hardware or software
- Programmable clock polarity and phase
- Programmable data order with MSB or LSB shifting
- Interrupt capability
- DMA event support

10.2 **Block Diagram**

Figure 6 SPI Block Diagram



spi_apb_if as apb slave and can generate registers, dma events and interrupt.

Two 16x16bits transmit and receive FIFOs for buffering output or input data.

spi_mode_mux merges spi master and spi slave data or request to/from FIFOs.

spi master is responsible for protocol implementation including parallel-to-serial converter in tx direction, serial-to-parallel converter in rx direction, sck generation based on phase, polarity and baud rate.

spi slave is responsible for protocol implementation including parallel-to-serial converter in tx direction, serial-to-parallel converter in rx direction, rx data sampling based on phase and polarity.

spi_io_if does the pinmux between master and slave which is connected with pads.

10.3 Function Description

10.3.1 SPI Pins

MISO: Master In/Slave Out data. In the general case, this pin is used to transmit data in slave mode and receive data in master mode.

MOSI: Master Out/Slave In data. In the general case, this pin is used to transmit data in master mode and receive data in slave mode.

SCK: Serial Clock output pin for SPI master or input pin for SPI slave.

NSS0: Slave0 select pin. Output for SPI master or input pin for SPI slave.

NSS1: Slave1 select pin. Output for SPI master.

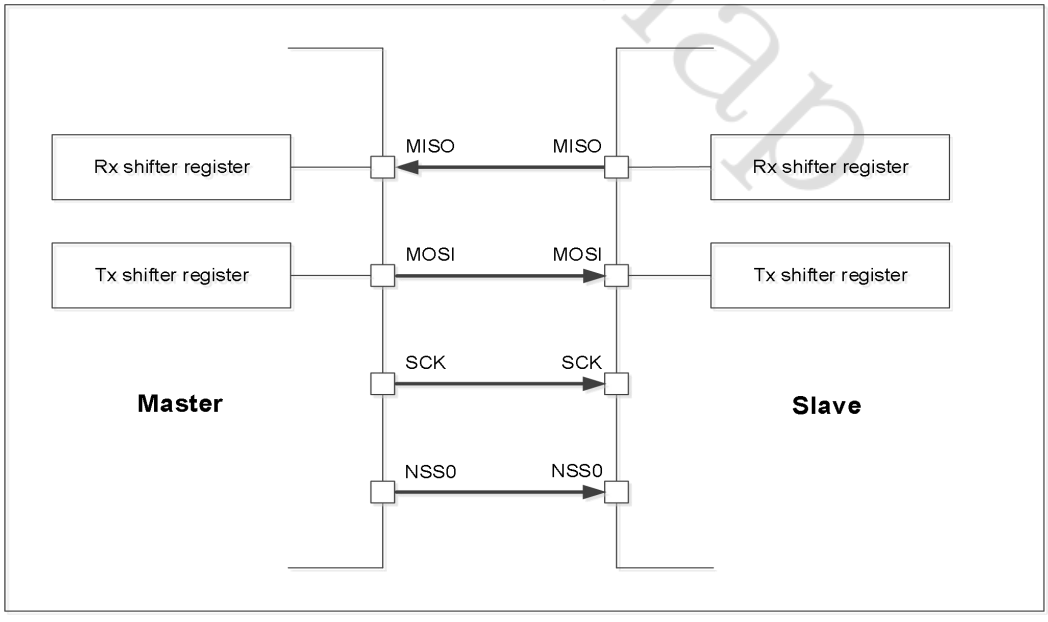
NSS2: Slave2 select pin. Output for SPI master.

NSS3: Slave3 select pin. Output for SPI master.

10.3.2 Communications

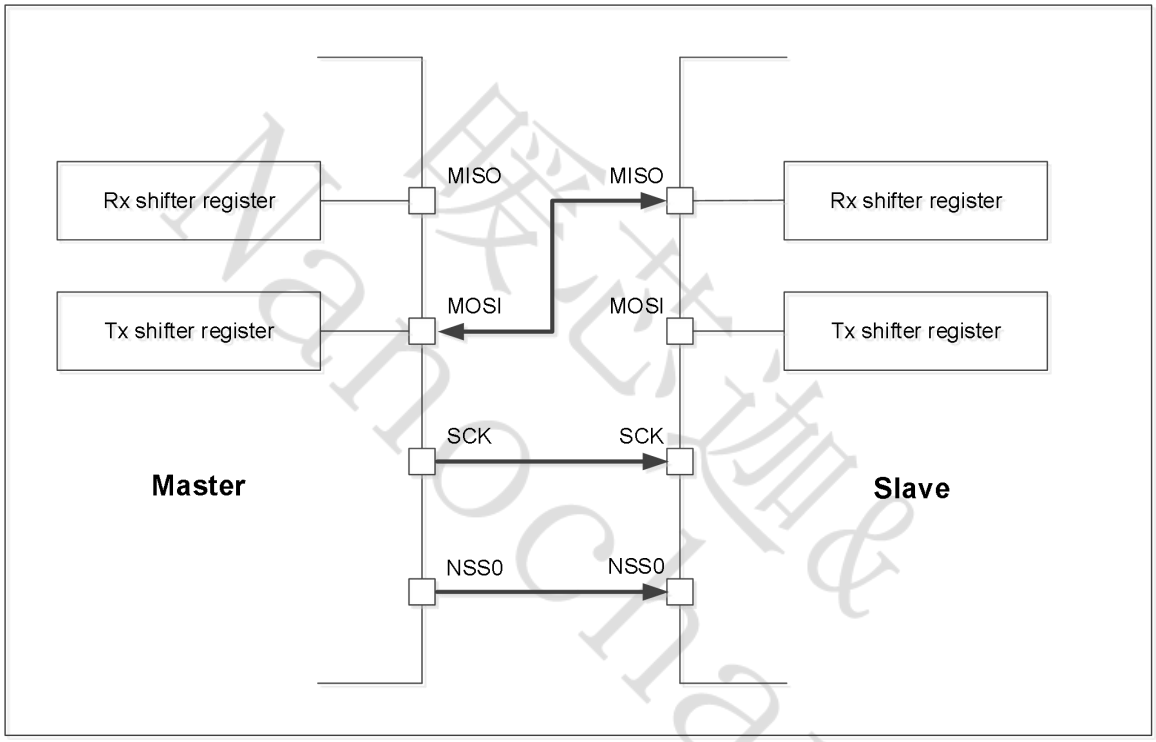
By default, the SPI is configured for full-duplex communication. In this configuration, the shift registers of the master and slave are linked using two unidirectional lines between the MOSI and the MISO pins. During SPI communication, data is shifted synchronously on the SCK clock edges provided by the master. The master transmits the data to be sent to the slave via the MOSI line and receives data from the slave via the MISO line. When the data frame transfer is complete (all the bits are shifted) the information between the master and slave is exchanged.

Figure 7 Full-duplex Application



The SPI can communicate in half-duplex mode by setting the BIDI_EN in the SPI_CTRL1 register. In this configuration, one single cross connection line is used to link the shift registers of the master and slave together. During this communication, the data is synchronously shifted between the shift registers on the SCK clock edge in the transfer direction selected reciprocally by both master and slave with the BIDI_MODE bit in the SPI_CTRL1 registers. In this configuration, the master's MISO pin and the slave's MOSI pin are free for other application used as GPIOs.

Figure 8 Half-duplex Application



The SPI can communicate in simplex mode by setting the SPI in transmit-only or in receive-only using UNIDI_MODE bits in the SPI_CTRL1 register. In this configuration, only one line is used for the transfer between the shift registers of the master and slave. The remaining MISO and MOSI pins pair is not used for communication and can be used as standard GPIOs.

Figure 9 Master in Transmit-only and Slave in Receive-only Mode

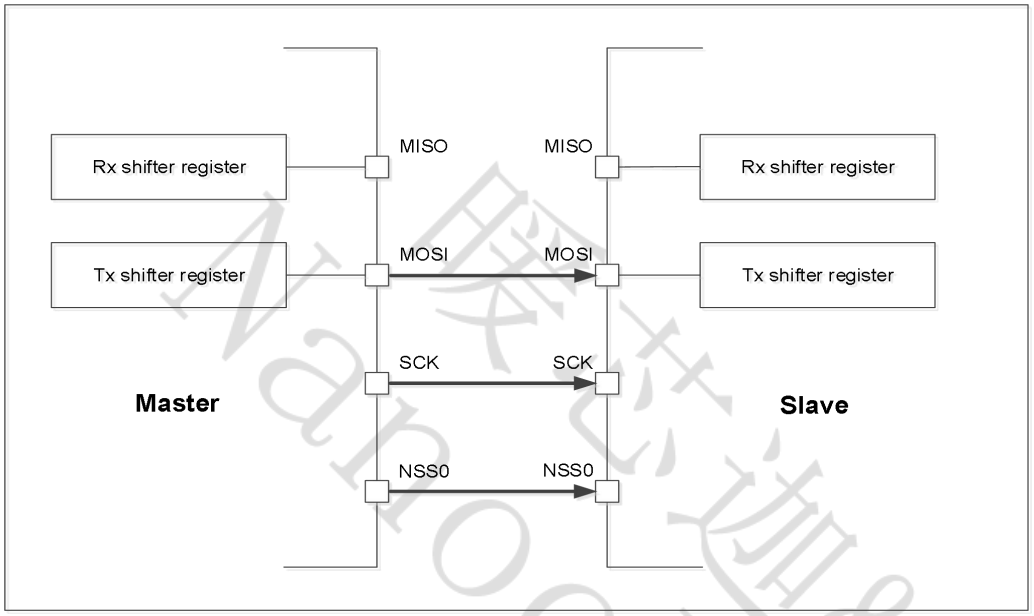
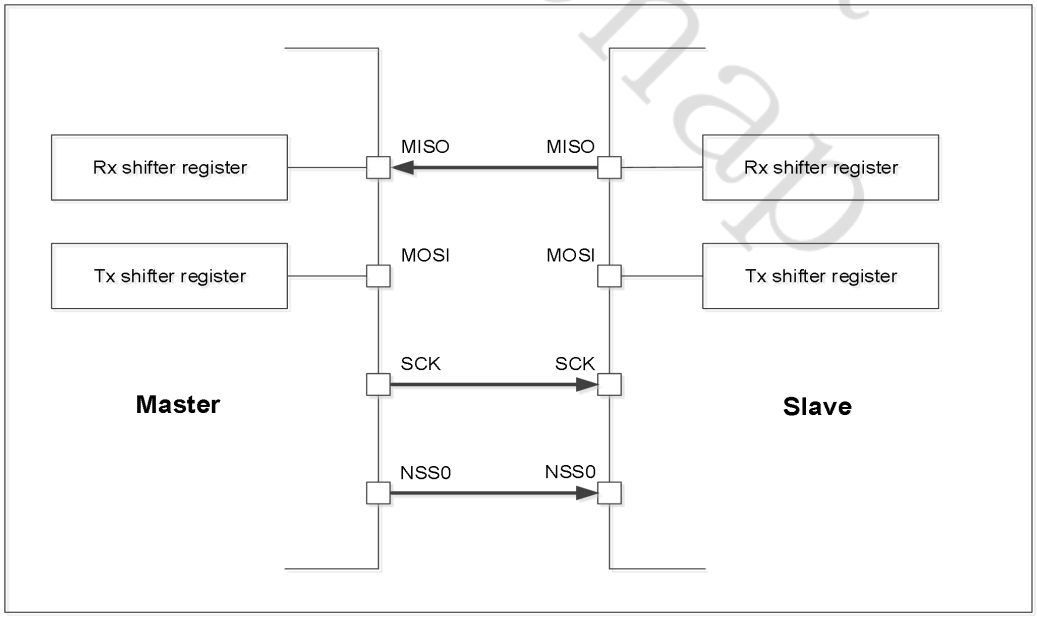
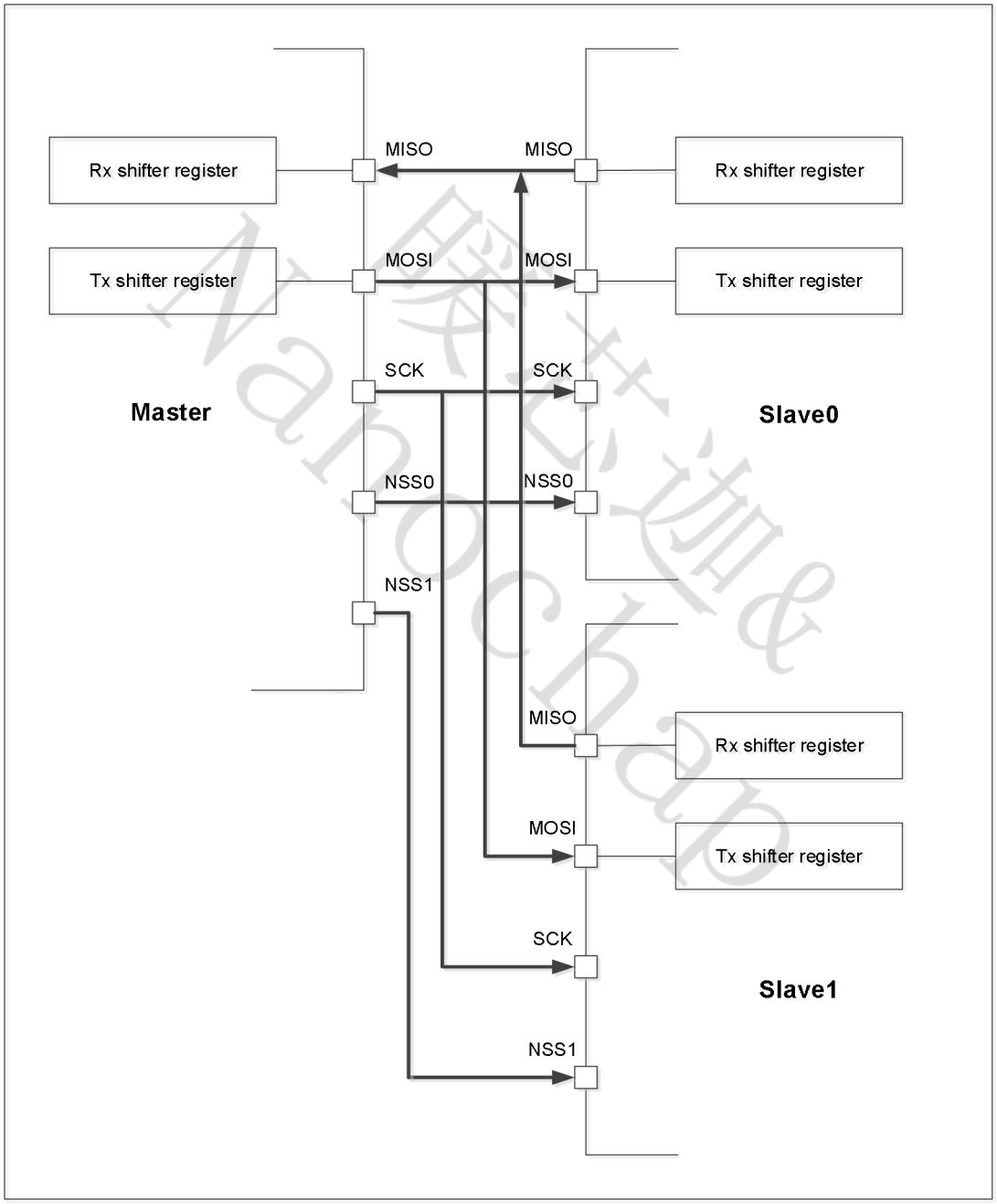


Figure 10 Master in Receive-only and Slave in Transmit-only Mode



In a configuration with up to 4 independent slaves, the master uses nss0~nss3 pins to manage the chip select lines for each slave. The master can select one of the slaves by setting NSS0_EN~NSS3_EN bits in SPI_CTRL2 register. When this is done, a standard master and dedicated slave communication is established.

Figure 11 Multi-slave Application

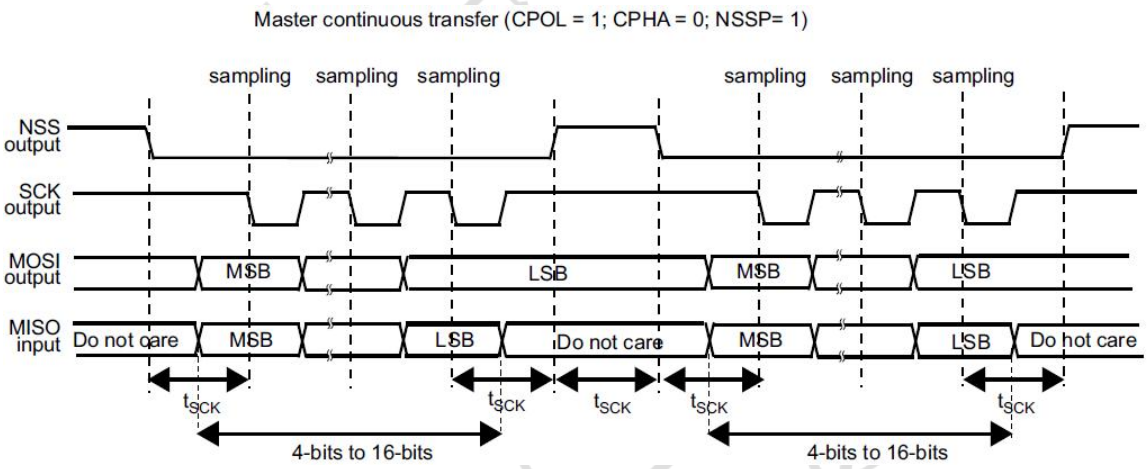


In slave mode, the NSS works as a standard chip select input and lets the slave communicate with the master.

In master mode, NSS is used as output pin. When NSS_MST_CTRL bit in SPI_CTRL1 register is high, NSS is driven by SW setting NSS_MST_SW bit. When NSS_MST_CTRL bit is low, NSS is driven by HW automatically.

NSS pulse mode is activated by the NSS_TOGGLE in SPI Control1 register. When activated, an NSS pulse is generated between two consecutive data frame transfers when NSS stays at high level for the duration of one clock period at least. This mode allows the slave to latch data.

Figure 12 NSS Pulse Mode



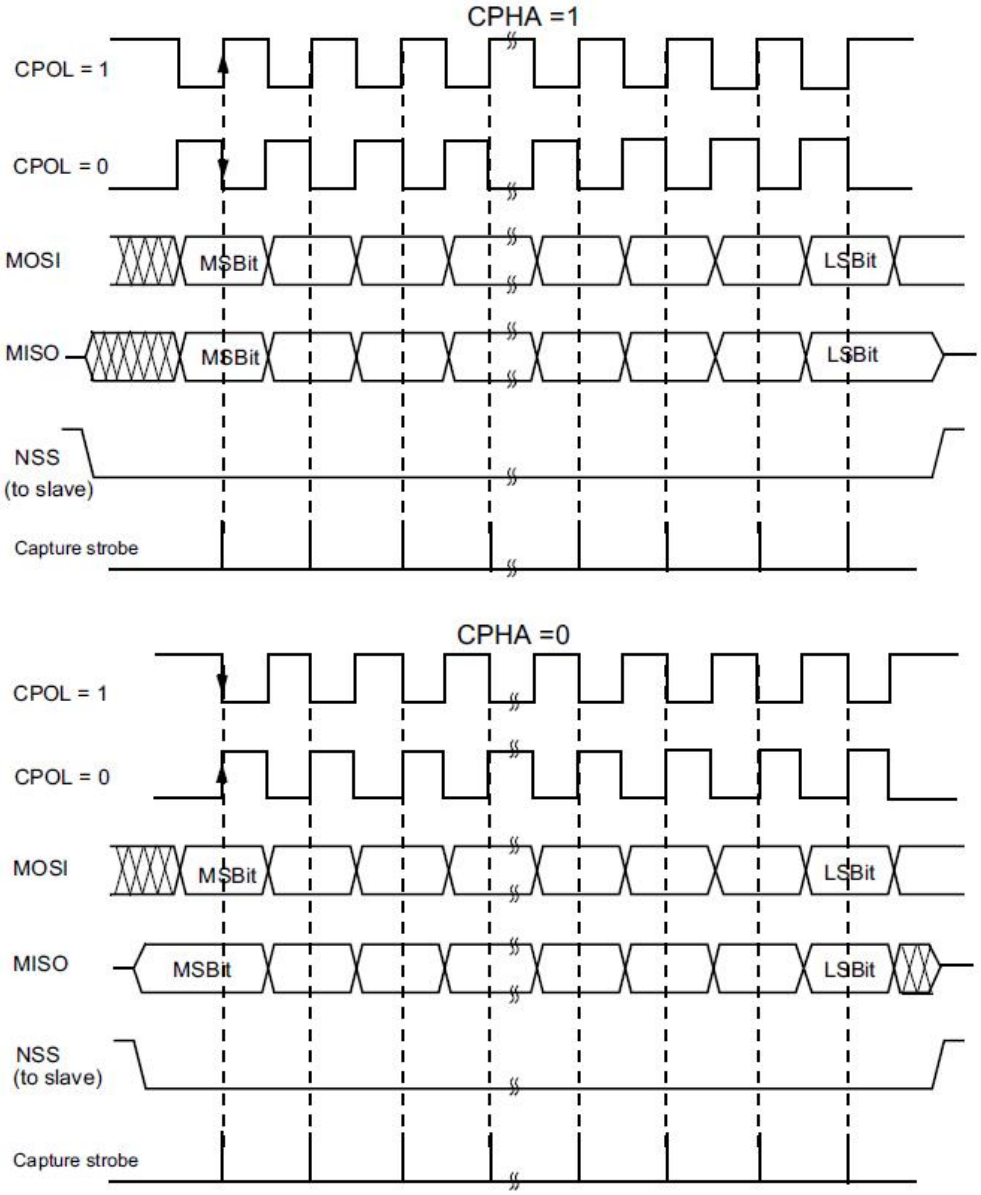
During SPI communication, receive and transmit operations are performed simultaneously. The serial clock (SCK) synchronizes the shifting and sampling of the information on the data lines. The communication format depends on the clock phase, the clock polarity and the data frame format. To be able to communicate together, the master and slaves devices must follow the same communication format.

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits in the SPI_CTRL1 register. The CPOL (clock polarity) bit controls the idle state value of the clock when no data is being transferred. This bit affects both master and slave modes. If CPOL is reset, the SCK pin has a low-level idle state. If CPOL is set, the SCK pin has a high-level idle state.

If the CPHA bit is set, the second edge on the SCK pin captures the first data bit transacted (falling edge if the CPOL bit is reset, rising edge if the CPOL bit is set). Data are latched on each occurrence of this clock transition type. If the CPHA bit is reset, the first edge on the SCK pin captures the first data bit transacted (falling edge if the CPOL bit is set, rising edge if the CPOL bit is reset). Data are latched on each occurrence of this clock transition type.

The combination of CPOL and CPHA bits selects the data capture clock edge.

Figure 13 Data Clock Timing Diagram



The SPI shift register can be set up to shift out MSB-first or LSB-first, depending on the value of LSB_SEL bit in SPI_CTRL1 register. The data frame size is chosen by using the CHAR_LEN bits in SPI_CTRL2 register. It can be set from 1-bit up to 16-bit length and the setting applies for both transmission and reception.

10.3.3 Configurations of SPI

The configuration procedure is almost the same for master and slave. When a standard communication is to be initialized, perform these steps:

1. Write proper GPIO alter registers: configure GPIO for MOSI, MISO ,SCK and NSS pins.

2. Write to the SPI_CTRL1 register:

a. Configure the serial clock baud rate using the BAUD_RATE[2:0] bits.

b. Configure the CPOL and CPHA bits combination to define one of the four relations between the data transfer and the serial clock.

c. Select full-duplex or half-duplex or simplex communication mode by configuring BIDI_EN, BIDI_MODE, UNIDI_MODE bits.

d. Configure the LSB_SEL bit to define the frame format.

e. Select NSS control mode by configuring NSS_TOGGLE,NSS_MST_CTRL, NSS_MST_SW bits.

f. Select master or slave mode by configuring the MST_SLV_SEL bit.

3. Write to the SPI_CTRL2 register:

a. Configure the CHAR_LEN[3:0] bits to select the data length for the transfer.

b. Select NSS port by configuring NSS0_EN, NSS1_EN, NSS2_EN or NSS3_EN bits.

c. Select proper RX data sample phase of master by configuring SAMP_PHASE[1:0] bits.

d. Select proper C2T/T2C delay based on slave device requirement by configuring C2T_DELAY and T2C_DELAY bits.

e. Enable or disable TX/RX DMA in FIFO mode by configuring TXDMA_EN and RXDMA_EN bits.

4. Write to the FIFO Control register:

a. Configure TX_FIFO_TH or RX_FIFO_TH to define the trigger level threshold.

b. Clear TX/RX FIFO by configure TX_FIFO_CLR and RX_FIFO_CLR bits.

c. Enable or disable FIFO mode by configuring FIFO_EN bit.

10.3.4 Sequence Handling

When transmission is enabled, a sequence begins and continues while any data is present in TXFIFO of the master. The clock signal is provided continuously by the master until TXFIFO becomes empty, then it stops waiting for additional data.

In receive-only modes, the master starts the sequence immediately when both SPI is enabled and receive-only mode is activated. The clock signal is provided by the master and it does not stop until either SPO or receive-only mode is disabled by the master. The master receives data frames continuously up to this moment.

While the master can provide all the transactions in continuous mode it has to respect slave capability to handle data flow and its content at anytime. When necessary, the master must slow down the communication and provide either a slower clock or separate frames or data sessions with sufficient delays. There is no underflow error for master, only overflow error for master. And for slave, there are both underflow and overflow error.

Each sequence must be encased by the NSS pulse in parallel with the multi-slave system to select just one the slaves for communication. In a single slave system it is not necessary to control the slave with NSS, but it is often better to provide the pulse here too, to synchronize the slave with the beginning of each data sequence. NSS can be managed by both software and hardware.

Procedure for enabling SPI:

When the SPI_BUSY bit is set it signified and ongoing data frame transaction. When the dedicated frame transaction is finished, the RXNE flag is raised.

It is recommended to enable the SPI slave before the master sends the clock. If not, undesired data transmission might occur. The data register of the slave must already contain data to be sent before starting communication with master.The SCK signal must be settled at an idle state level corresponding to the selected polarity before the SPI slave is enable.

The master at full-duplex (or in any transmit-only mode) starts to communicate when SPI is enabled and TXFIFO is not empty, or with the next write to TXFIFO.

In any master receive only mode, master starts to communicate and the clock starts running immediately after SPI is enabled.

Procedure for disabling SPI:

When SPI is disabled, it is mandatory to follow the disable procedures. It is important to do this before the system enters a low-power mode when the peripheral clock is stopped, Ongoing transactions can be corrupted in this case. In some modes the disable procedure is the only way to stop continuous communication running.

Master in full-duplex or transmit only mode can finish any transaction when it stops providing data for transmission. In this case, the clock stops after the last data transaction.

When master is in any receive only mode, the only way to stop the continuous clock is to disable SPI. This must occur in specific time window within last data frame transaction just between the sampling time of its first bit and before its last bit transfer starts (in order to receive a complete number of expected data frames and to prevent any additional "dummy" data reading after the last valid data frame). Specific procedure must be followed when disabling SPI in this mode.

Data received but not read remains stored in RXFIFO when SPI is disabled, and must be processed the next time the SPI is enabled, before starting a new sequence. To prevent having unread data, ensure that RXFIFO is empty when disabling the SPI, by using the correct disabling procedure.

Standard disable procedure is based on pulling SPI_BUSY status together with TX_FIFO_LEN[4:0] in FIFO status register to check if a transmission session is fully completed.

The correct disable procedure is (except when receive only mode is used):

1. Wait until TX_FIFO_LEN[4:0]=5'd0 (no more data to transmit).

2. Wait until SPI_BUSY=0 (the last data frame is processed).

3. Disable SPI by configuring SPI_EN=0.

4. Read data until RX_FIFO_LEN[4:0]=5'd0 (read all the received data)

The correct disable procedure for certain receive only mode is:

1. Interrupt the receive flow by disabling SPI in the specific time window while the last data frame is ongoing(In FIFO mode, can read RX_FIFO_LEN then decided to disable SPI. In non-FIFO mode, can wait overrun error then disable SPI).

2. Wait until SPI_BUSY=0 (the last data frame is processed).

3. Read data until RX_FIFO_LEN[4:0]=5'd0 (read all the received data).

Busy flag:

The SPI_BUSY flag is set and cleared by hardware.

When SPI_BUSY is set, it indicates that a data transfer is in process on the SPI (SPI bus is busy).

The SPI_BUSY flag can be used in certain modes to detect the end of a transfer so that the software can disable the SPI or its peripheral clock before entering a low-power mode which does not provide a clock for the peripheral.

The SPI_BUSY is cleared under any one of the following conditions:

1. When SPI is correctly disabled

2. In master mode, when it finishes a data transmission and no new data is ready to be sent

3. In slave mode, when NSS is inactive

10.4 **Registers**

The system programmer has access to and control over any of the SPI registers that are listed as below. These registers, which control SPI operations, receive data, and transmit data, are available at 32-bit addresses in the device memory map.

Table 22 SPI Registers

Offset	Acronym	Register Description
00h	RBR	Receiver Buffer Register (read only)
00h	THR	Transmitter Holding Register (write only)
04h	IER	Interrupt Enable Register
08h	ISR	Interrupt Status Register
0Ch	CTRL1	Control1 Register
10h	CTRL2	Control2 Register
14h	FCR	FIFO Control Register
18h	FSR	FIFO Status Register
1Ch	DBG	Debug Signal Register

Offset Address: 03-00h

Receive Buffer Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:0	RBR	RO	0	Any data bytes received by SPI are accessed by reading this register

Offset Address: 03-00h

Transmitter holding Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:0	THR	WO	0	This register is used to buffer outgoing character.

Offset Address: 07-04h

Interrupt Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:5	-	RO	0	Reserved
4	UNDERRUN_INT_EN	RW	0	TX underrun interrupt enable bit 0: UNDERRUN interrupt disabled

				1: UNDERRUN interrupt enabled
3	OVERRUN_INT_EN	RW	0	RX overrun interrupt enable bit 0: OVERRUN interrupt disabled 1: OVERRUN interrupt enabled
2	CMPL_INT_EN	RW	0	TX/RX transaction completed interrupt enable bit 0: CMPL interrupt disable 1: CMPL interrupt enable
1	TXE_INT_EN	RW	0	TX buffer empty interrupt enable bit 0: TXE interrupt disabled 1: TXE interrupt enabled
0	RXNE_INT_EN	RW	0	RX buffer not empty interrupt enable bit 0: RXNE interrupt disabled 1: RXNE interrupt enabled

Offset Address: 0B-08h

Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:5	-	RO	0	Reserved
4	UNDERRUN_INT_STS	RW1C	0	TX underrun interrupt status, this bit can be cleared by writing 1 to this bit
3	OVERRUN_INT_STS	RW1C	0	RX overrun interrupt status, this bit can be cleared by writing 1 to this bit
2	CMPL_INT_STS	RW1C	0	TX/RX transaction completed status, when in master mode, this interrupt will be generated when TX FIFO is empty and RX FIFO is full, when in slave mode, this interrupt will be generated after one data transfer, this bit can be cleared by writing 1 to this bit
1	TXE_INT_STS	RO	1	When in non-FIFO mode 0: Transmit buffer not empty 1: Transmit buffer empty, when write THR register, this status bit is cleared to 0 When in FIFO mode 0: TX FIFO length > TX FIFO Threshold

				1: TX FIFO length <= TX FIFO Threshold
0	RXNE_INT_STS	RO	0	When in non-FIFO mode 0: Receive buffer empty 1: Receive buffer not empty, when read RBR register, this status bit is cleared to 0 When in FIFO mode 0: RX FIFO length < RX FIFO Threshold 1: RX FIFO length >= RX FIFO Threshold

Offset Address: 0F-0Ch

SPI Control1 Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15	BIDI_EN	RW	0	Bidirectional data mode enable bit 0: 2-line unidirectional data mode selected 1: 1-line bidirectional data mode selected
14	BIDI_MODE	RW	0	Bidirectional mode 0: transmit-only 1: receive-only
13:12	UNIDI_MODE	RW	0	Unidirectional mode 00: transmit and receive 01: transmit only 10: receive only 11: reserved
11	NSS_TOGGLE	RW	0	This bit allows the SPI master to generate an NSS pulse between two consecutive data when doing continuous transfers 0: No NSS pulse 1: NSS pulse generated
10	LOOPBACK_EN	RW	0	Internal loop-back test mode, MOSI pin is internally connected to the MISO pin 0: Internal loop-back test mode disabled

				1: Internal loop-back test mode enabled
9	NSS_MST_SW	RW	0	NSS value controlled by SW 0: value 0, NSS is asserted 1: value 1, NSS is deasserted
8	NSS_MST_CTRL	RW	0	NSS master control 0: NSS is controlled by HW 1: NSS is controlled by SW
7	LSB_SEL	RW	0	Frame format 0: data is transmitted / received with the MSB first 1: data is transmitted / received with the LSB first
6:4	BAUD_RATE	RW	0	Baud rate control 000: Fpclk/2 001: Fpclk/4 010: Fpclk/8 011: Fpclk/16 100: Fpclk/32 101: Fpclk/64 110: Fpclk/128 111: Fpclk/256
3	CPOL	RW	0	Clock polarity 0: SCK to 0 when idle 1: SCK to 1 when idle
2	CPHA	RW	0	Clock phase 0: The first clock transition is the first data capture edge 1: The second clock transition is the first data capture edge
1	MST_SLV_SEL	RW	1	Master and slave mode selection 0: Slave configuration 1: Master configuration
0	SPI_EN	RW	0	SPI enable bit

				0: SPI disabled 1: SPI is enabled

Offset Address: 13-10h

SPI Control2 Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:14	T2C_DELAY	RW	00b	Transmit-end-to-chip-inactive delay. T2C_DELAY is used in master mode only. It defines a hold time for the slave device that delays the chip select deactivation by a multiple of SCK cycles after the last bit is transferred. 00: delay 1T SCK 01: delay 2T SCK 10: delay 3T SCK 11: delay 4T SCK
13:12	C2T_DELAY	RW	00b	Chip-select-active-to-transmit-start delay. C2T_DELAY is used in master mode only. It defines a setup time for the slave device that delays the data transmission from the chip select active edge by a multiple of SCK cycles. 00: delay 1T SCK 01: delay 2T SCK 10: delay 3T SCK 11: delay 4T SCK
11	NSS3_EN	RW	0	NSS3 port enable bit 0: disabled 1: enabled
10	NSS2_EN	RW	0	NSS2 port enable bit 0: disabled 1: enabled
9	NSS1_EN	RW	0	NSS1 port enable bit 0: disable 1: enable

8	NSS0_EN	RW	0	NSS0 port enable bit 0: disabled 1: enabled
7:6	SAMP_PHASE	RW	01b	RX sample phase, used only in master mode 00: Pre 1T Pclk period 01: Normal 10: Delay 1T Pclk period 11: Delay 2T Pclk period
5	TXDMA_EN	RW	0	TX DMA enable when in FIFO mode 0: TX DMA disabled 1: TX DMA enabled
4	RXDMA_EN	RW	0	RX DMA enable when in FIFO mode 0: RX DMA disabled 1: RX DMA enabled
3:0	CHAR_LEN	RW	0111b	Character length: 0000, 0001, 0010: 8-bit 0011: 4-bit 0100: 5-bit 0101: 6-bit 0110: 7-bit 0111: 8-bit 1000: 9-bit 1001: 10-bit 1010: 11-bit 1011: 12-bit 1100: 13-bit 1101: 14-bit 1110: 15-bit 1111: 16-bit

Offset Address: 17-14h

FIFO Control Register

Bit	Field Name	Attribute	Default	Field Description
31:14	-	RO	0	Reserved
13:9	TX_FIFO_TH	RW	0	Transmitter FIFO trigger level threshold 00000: 0 character 00001: 1 character ..... 10000: 16 character
8	TX_FIFO_CLR	WO	0	Transmitter FIFO clear. 0 = No effect. 1 = Clears transmitter FIFO and resets the transmitter FIFO counter.
7	-	RO	0	Reserved
6:2	RX_FIFO_TH	RW	0	Receiver FIFO trigger level threshold 00000: 0 character 00001: 1 character ..... 10000: 16 character
1	RX_FIFO_CLR	WO	0	Receiver FIFO clear. 0 = No effect. 1 = Clears receiver FIFO and resets the receiver FIFO counter.
0	FIFO_EN	RW	0	Transmitter and receiver FIFOs mode enable. 0 = Non-FIFO mode. The transmitter and receiver FIFOs are disabled, and the FIFO pointers are cleared. 1 = FIFO mode. The transmitter and receiver FIFOs are enabled.

Offset Address: 1B-18h

FIFO Status Register

Bit	Field Name	Attribute	Default	Field Description
31:21	-	RO	0	Reserved
20:16	RX_FIFO_LEN	RO	0	Receiver FIFO length indicator bit
15:13	-	RO	0	Reserved
12:8	TX_FIFO_LEN	RO	0	Transmitter FIFO length indicator bit.
7:5	-	RO	0	Reserved
4	SPI_BUSY	RO	0	0: SPI is not busy 1: SPI is busy
3	RX_FIFO_FULL	RO	0	Receiver FIFO full indicator bit
2	RX_FIFO_EMPTY	RO	1	Receiver FIFO empty indicator bit

1	TX_FIFO_FULL	RO	0	Transmitter FIFO full indicator bit.
0	TX_FIFO_EMPTY	RO	1	Transmitter FIFO empty indicator bit.

Offset Address: 1F-1Ch

Debug Signal Register

Bit	Field Name	Attribute	Default	Field Description
31:24	APB_IF_DBG[7:0]	RO	0	APB interface module internal signal
23:12	SPIS_DBG[11:1]	RO	0	SPI slave module internal signal
12	SPIS_BUSY	RO	0	SPI slave busy
11:1	SPIM_DBG[11:1]	RO	0	SPI master module internal signal
0	SPIM_BUSY	RO	0	SPI master busy

11 Universal Asynchronous Receiver Transmitter (UART)

11.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) peripheral is based on the industry standard TL16C550 asynchronous communications element, which in turn is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (single character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the CPU of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the CPU. The CPU can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link.

The UART includes a programmable baud generator capable of dividing the UART input clock by divisors from 1 to 65535 and producing a 16× reference clock or a 13× reference clock for the internal transmitter and receiver logic. For detailed timing and electrical specifications for the UART, see the device-specific data manual.

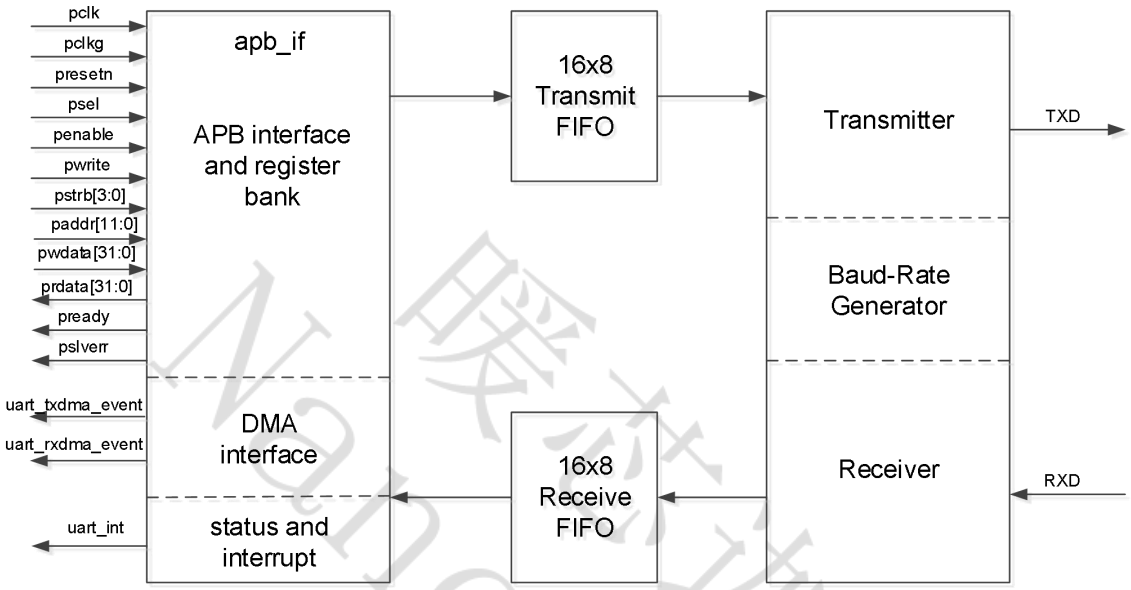
11.1.1 Feature List

- Compliance with AMBA APB specifications
- Support for up to 115200bps baud-rate
- Separate transmit and receive FIFO buffers (16 bytes) to reduce interrupts
- Programmable baud rate generator
- Support auto-flow control
- Standard asynchronous communication bits (start, stop, parity)
- DMA event support
- Support loopback test
- Interrupt support
- Fully-programmable serial interface characteristics:
  - Data can be 5, 6, 7, 8 bits
  - Even, odd or no-parity bit generation and detection
  - 1, 1.5 or 2-stop bit generation

11.2 **Block diagram**

A functional block diagram of the UART is shown as below.

Figure 14 UART Block Diagram



11.3 Function Description

11.3.1 Baud Rate Control

The 16× or 13× reference clock is selected by configuring the OSM_SEL bit in the mode definition register (MDR). The formula to calculate the divisor is:

When OSM_SEL=0, Desired baud rate = input clock frequency/((divisor+1) x 16).

When OSM_SEL=1, Desired baud rate = input clock frequency/((divisor+1) x 13).

When the UART is receiving in the middle bit period, the bit is sampled in the 8th cycle for 16× over sampling mode and on the 6th cycle for 13× oversampling mode.

Two 8-bit register fields (DLH and DLL), called divisor latches, hold this 16-bit divisor. DLH holds the most significant bits of the divisor, and DLL holds the least significant bits of the divisor. These divisor latches must be loaded during initialization of the UART to ensure desired operation of the baud generator.

11.3.2 Protocol Description

Transmission:

The UART transmitter section includes a transmitter hold register (THR) and a transmitter shift register (TSR). When the UART is in the FIFO mode, THR is a 16-byte FIFO. Transmitter section control is a function of the UART line control register (LCR). Based on the settings chosen in the LCR, the UART transmitter sends the following to the receiving device:

- 1 START bit
- 5, 6, 7, or 8 data bits
- 1 PARITY bit (optional)
- 1, 1.5, or 2 STOP bits

THR receives data from the internal data bus, and when TSR is ready, the UART moves the data from THR to TSR. The UART serializes the data in TSR and transmits the data on the uart_txd pin. In the non-FIFO mode, if THR is empty and the THR empty interrupt is enabled in the interrupt enable register (IER), an interrupt is generated. This interrupt is cleared when a character is loaded into THR. In the FIFO mode, the interrupt is generated when the transmitter FIFO is empty, and it is cleared when at least one byte is loaded into the FIFO.

Reception:

The UART receiver section includes a receiver shift register (RSR) and a receiver buffer register (RBR). When the UART is in the FIFO mode, RBR is a 16-byte FIFO. Receiver section control is a function of the UART line control register (LCR).

Based on the settings chosen in LCR, the UART receiver accepts the following from the transmitting device:

- 1 START bit
- 5, 6, 7, or 8 data bits
- 1 PARITY bit (optional)
- 1 STOP bit (any other STOP bits transferred with the above data are not detected)

RSR receives the data bits from the uart_rxd pin. Then RSR concatenates the data bits and moves the resulting value into RBR (or the receiver FIFO). The UART also stores three bits of error status information next to each received character, to record a parity error, framing error, or break.

In the non-FIFO mode, when a character is placed in RBR and the receiver data-ready interrupt is enabled in the interrupt enable register (IER), an interrupt is generated. This interrupt is cleared when the character is read from RBR. In the FIFO mode, the interrupt is generated when the FIFO is filled to the trigger level selected in the FIFO control register (FCR), and it is cleared when the FIFO contents drop below the trigger level.

Data Format:

The UART transmits in the following format:

1 START bit + data bits (5, 6, 7, 8) + 1 PARITY bit (optional) + STOP bit (1, 1.5, 2).

It transmits 1 START bit; 5, 6, 7, or 8 data bits, depending on the data width selection; 1 PARITY bit, if parity is selected; and 1, 1.5, or 2 STOP bits, depending on the STOP bit selection.

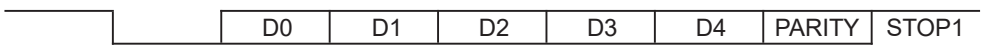
The UART receives in the following format:

1 START bit + data bits (5, 6, 7, 8) + 1 PARITY bit (optional) + 1 STOP bit.

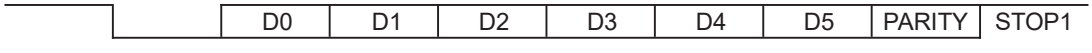
It receives 1 START bit; 5, 6, 7, or 8 data bits, depending on the data width selection; 1 PARITY bit, if parity is selected; and 1 STOP bit.

The protocol formats are shown as below.

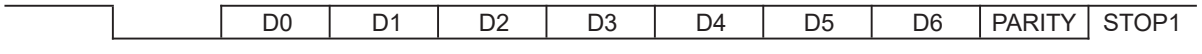
Transmit/Receive for 5-bit data, parity Enable, 1 STOP bit



Transmit/Receive for 6-bit data, parity Enable, 1 STOP bit



Transmit/Receive for 7-bit data, parity Enable, 1 STOP bit



Transmit/Receive for 8-bit data, parity Enable, 1 STOP bit

		D0	D1	D2	D3	D4	D5	D6	D7	PARITY	STOP1

11.3.3 FIFO Modes

The following two modes can be used for servicing the receiver and transmitter FIFOs:

FIFO interrupt mode. The FIFO is enabled and the associated interrupts are enabled. Interrupts are sent to the CPU to indicate when specific events occur.

FIFO poll mode. The FIFO is enabled but the associated interrupts are disabled.

The CPU polls status bits to detect specific events.

Because the receiver FIFO and the transmitter FIFO are controlled separately, either one or both can be placed into the interrupt mode or the poll mode.

FIFO Interrupt Mode:

When the receiver FIFO is enabled in the FIFO control register (FCR) and the receiver interrupts are enabled in the interrupt enable register (IER), the interrupt mode is selected for the receiver FIFO. The following are important points about the receiver interrupts:

- The receiver data-ready interrupt is issued to the CPU when the FIFO has reached the trigger level that is programmed in FCR. It is cleared when the CPU or the DMA controller reads enough characters from the FIFO such that the FIFO drops below its programmed trigger level.
- The receiver line status interrupt is generated in response to an overrun error, a parity error, a framing error, or a break. This interrupt has higher priority than the receiver data-ready interrupt.
- The data-ready (DR) bit in the line status register (LSR) indicates the presence or absence of characters in the receiver FIFO. The DR bit is set when a character is transferred from the receiver shift register (RSR) to the empty receiver FIFO. The DR bit remains set until the FIFO is empty again.
- A receiver time-out interrupt occurs if all of the following conditions exist:
  - At least one character is in the FIFO,
  - The most recent character was received more than four continuous character times ago. A character time is the time allotted for 1 START bit, n data bits, 1 PARITY bit, and 1 STOP bit, where n depends on the word length selected with the WORD_LEN bits in the line control register (LCR).

– The most recent read of the FIFO has occurred more than four continuous character times before.

• Character times are calculated by using the baud rate.

• When a receiver time-out interrupt has occurred, it is cleared and the time-out timer is cleared when the CPU or the DMA controller reads one character from the receiver FIFO. The interrupt is also cleared if a new character is received in the FIFO or if the URRST bit is cleared in the power management register .

• If a receiver time-out interrupt has not occurred, the time-out timer is cleared after a new character is received or after the CPU or DMA reads the receiver FIFO.

When the transmitter FIFO is enabled in FCR and the transmitter holding register empty interrupt is enabled in IER, the interrupt mode is selected for the transmitter FIFO. The transmitter holding register empty interrupt occurs when the transmitter FIFO is empty. It is cleared when the transmitter hold register (THR) is loaded (1 to 16 characters may be written to the transmitter FIFO while servicing this interrupt).

Table 23 UART Character Time

Word Length (n)	Character Time	Four Character Times
5	Time for 8 bits	Time for 32 bits
6	Time for 9 bits	Time for 36 bits
7	Time for 10 bits	Time for 40 bits
8	Time for 11 bits	Time for 44 bits

FIFO Poll Mode:

When the receiver FIFO is enabled in the FIFO control register (FCR) and the receiver interrupts are disabled in the interrupt enable register (IER), the poll mode is selected for the receiver FIFO. Similarly, when the transmitter FIFO is enabled and the transmitter interrupts are disabled, the transmitted FIFO is in the poll mode. In the poll mode, the CPU detects events by checking bits in the line status register (LSR):

• The RXFIFOE bit indicates whether there are any errors in the receiver FIFO.

• The TEMT bit indicates that both the transmitter holding register (THR) and the transmitter shift register (TSR) are empty.

• The THRE bit indicates when THR is empty.

• The BI (break), FE (framing error), PE (parity error), and OE (overrun error) bits specify which error or errors have occurred.

• The DR (data-ready) bit is set as long as there is at least one byte in the receiver FIFO.

Also, in the FIFO poll mode:

- The interrupt identification register (IIR) is not affected by any events because the interrupts are disabled.
- The UART does not indicate when the receiver FIFO trigger level is reached or when a receiver time-out occurs.

11.3.4 Auto Flow Control

The UART can employ autoflow control by connecting the uart_cts_n and uart_rts_n signals. The uart_cts_n input must be active before the transmitter FIFO can transmit data. The uart_rts_n becomes active when the receiver needs more data and notifies the sending device. When uart_rts_n is connected to uart_cts_n, data transmission does not occur unless the receiver FIFO has space for the data. Therefore, with autoflow enabled, overrun errors are eliminated.

RTS Behavior:

When the receiver FIFO level reaches a trigger level of 1, 4, 8, or 14, uart_rts_n is deasserted. The sending UART may send an additional byte after the trigger level is reached(assuming the sending UART has another byte to send), because it may not recognize the deassertion of uart_rts_n until after it has begun sending the additional byte. For trigger level 1, 4 and 8, when register RTS_TRI_MODE is 1, uart_rts_n is automatically reasserted when the receiver FIFO drops below the trigger level, when register RTS_TRI_MODE is 0, uart_rts_n is automatically reasserted after the receiver FIFO is empty. For trigger level 14, uart_rts_n is automatically reasserted when the receiver FIFO drops below the trigger level.

CTS Behavior:

The transmitter checks uart_cts_n before sending the next data byte. If uart_cts_n is active, the transmitter sends the next byte. To stop the transmitter from sending the following byte, uart_cts_n must be released before the middle of the last STOP bit that is currently being sent. When flow control is enabled, uart_cts_n level changes do not trigger interrupts because the device automatically controls its own transmitter. Without autoflow control, the transmitter sends any data present in the transmitter FIFO and a receiver overrun error may result.

11.3.5 Loopback Control

The UART can be placed in the diagnostic mode using the LOOPBACK_EN bit in the modem control register (MCR), which internally connects the UART output back to the UART input. In this mode, the transmit and receive data paths, the transmitter and receiver interrupts, and the modem control interrupts can be verified without connecting to another UART.

11.3.6 Reset

Software Reset

Two bits in the power management register control resetting the parts of the UART:

- The TXRST bit controls resetting the transmitter only. If TXRST = 0 the transmitter is active; if TXRST = 1, the transmitter including Tx FIFO is in reset.
- The RXRST bit controls resetting the receiver only. If RXRST = 0 the receiver is active; if RXRST = 1, the receiver including Rx FIFO is in reset.

In each case, putting the receiver and/or transmitter in reset will reset the state machine of the affected portion but does not affect the UART registers.

Hardware Reset

When the processor RESET pin is asserted, the entire processor is reset and is held in the reset state until the RESET pin is released. As part of a device reset, the UART state machine is reset and the UART registers are forced to their default states.

11.3.7 Initialization

The following steps are required to initialize the UART:

1. Perform the necessary device pin multiplexing setup.

2. Set the desired baud rate by writing the appropriate clock divisor values to the divisor latch registers (DLL and DLH).

3. If the FIFOs will be used, select the desired trigger level and enable the FIFOs by writing the appropriate values to the FIFO control register (FCR). The FIFOEN bit in FCR must be set first, before the other bits in FCR are configured.

4. Choose the desired protocol settings by writing the appropriate values to the line control register (LCR).

5. If autoflow control is desired, write appropriate values to the modem control register (MCR). Note that not all UARTs support autoflow control, see the device-specific data manual for supported features.

6. Choose the desired response to emulation suspend events by configuring the FREE bit and enable the UART by setting the TXRST and RXRST bits in the power management register (PMU).

11.3.8 Interrupt Support

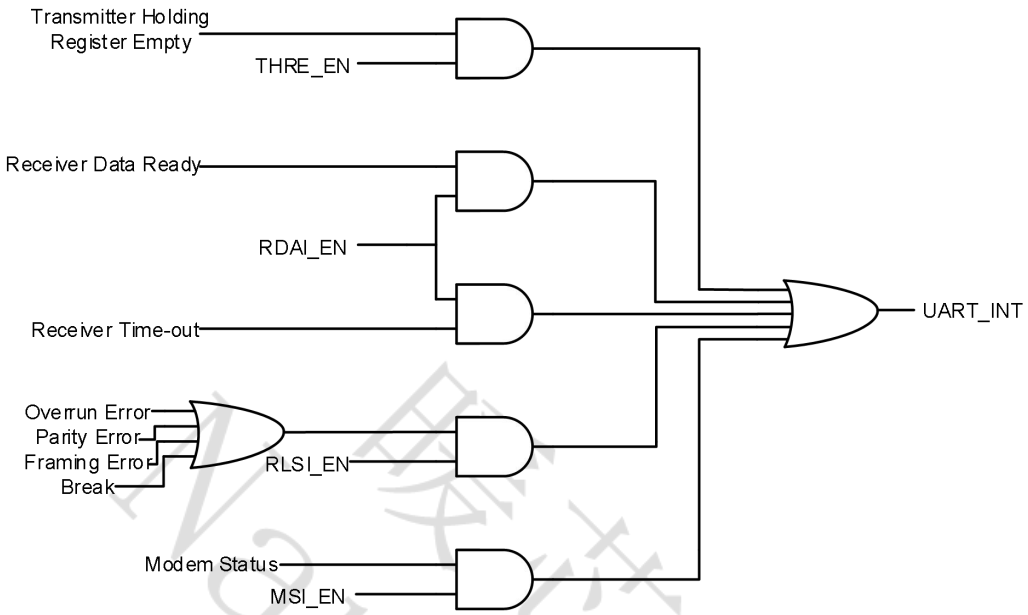
The UART generates the interrupt requests described in blow table. All requests are multiplexed through an arbiter to a single UART interrupt request to the CPU. Each of the interrupt requests has an enable bit in the interrupt enable register (IER) and is recorded in the interrupt identification register (IIR).

If an interrupt occurs and the corresponding enable bit is set to 1, the interrupt request is recorded in IIR and is forwarded to the CPU. If an interrupt occurs and the corresponding enable bit is cleared to 0, the interrupt request is blocked. The interrupt request is neither recorded in IIR nor forwarded to the CPU.

Table 24 UART Interrupt Source

UART Interrupt Request	Interrupt Source	Comment
THREINT	THR-empty condition: The transmitter holding register (THR) or the transmitter FIFO is empty. All of the data has been copied from THR to the transmitter shift register (TSR).	If THREINT is enabled in IER, by setting the THRE_EN bit, it is recorded in IIR. As an alternative to using THREINT, the CPU can poll the THRE
RDAINT	Receive data available in non-FIFO mode or trigger level reached in the FIFO mode.	If RDAINT is enabled in IER, by setting the RDAI_EN bit, it is recorded in IIR. As an alternative to using RDAINT, the CPU can poll the DR bit in the line status register (LSR). In the FIFO mode, this is not a functionally equivalent alternative because the DR bit does not respond to the FIFO trigger level. The DR bit indicates only the presence or absence of unread characters.
RTOINT	Receiver time-out condition (in the FIFO mode only): No characters have been removed from or input to the receiver FIFO during the last four character times, and there is at least one character in the receiver FIFO during this time.	The receiver time-out interrupt prevents the UART from waiting indefinitely when the receiver FIFO level is below the trigger level and, therefore, does not generate a receiver data-ready interrupt. If RTOINT is enabled in IER, by setting the RDAI_EN bit, it is recorded in IIR. There is no status bit to reflect the occurrence of a time-out condition.
RLSINT	Receiver line status condition: An overrun error, parity error, framing error, or break has occurred.	• If RLSINT is enabled in IER, by setting the RLSI_EN bit, it is recorded in IIR. • As an alternative to using RLSINT, the CPU can poll the following bits in the line status register (LSR): overrun error indicator (OE), parity error indicator (PE), framing error indicator (FE), and break indicator (BI).
MSIINT	Modem status: CTS change status(autoflow disable), DSR/RI/DCD change status	• If MSIINT is enabled in IER, by setting the MSI_EN bit, it is recorded in IIR. • As an alternative to using MSINT, the CPU can poll the following bits in the modem status register (MSR): DELTA_CTS_STS, DELTA_DSR_STS, DELTA_RI_STS and DELTA_DCD_STS

Figure 15 UART Interrupt Output



11.4 **Registers**

The system programmer has access to and control over any of the UART registers that are listed as below. These registers, which control UART operations, receive data, and transmit data, are available at 32-bit addresses in the device memory map. See the device-specific data manual for the memory address of these registers.

- RBR, THR, and DLL share one address. When the DLAB bit in LCR is 0, reading from the address gives the content of RBR, and writing to the address modifies THR. When DLAB = 1, all accesses at the address read or modify DLL. DLL can also be accessed with address offset 20h, that is dedicated address, if dedicated address is used, DLAB can be 0, so that the RBR and THR are always selected at the shared address.
- IER and DLH share one address. When DLAB = 0, all accesses read or modify IER. When DLAB = 1, all accesses read or modify DLH. DLH can also be accessed with address offset 24h, that is dedicated address, if dedicated address is used, DLAB can be 0, so that the IER are always selected at the shared address.
- IIR and FCR share one address. Regardless of the value of the DLAB bit, reading from the address gives the content of IIR, and writing modifies FCR.

Table 25 UART Registers

Offset	Acronym	Register Description
00h	RBR	Receiver Buffer Register (read only)
00h	THR	Transmitter Holding Register (write only)
04h	IER	Interrupt Enable Register
08h	IIR	Interrupt Identification Register (read only)
08h	FCR	FIFO Control Register (write only)
0Ch	LCR	Line Control Register
10h	MCR	Modem Control Register
14h	LSR	Line Status Register
18h	MSR	Modem Status Register
1Ch	SCR	Scratch Pad Register
20h	DLL	Divisor LSB Latch
24h	DLH	Divisor MSB Latch
28h	FSR	FIFO Status Register
2Ch	DBG	Debug Signal Register
30h	PMU	Power Management Register
34h	MDR	Mode Definition Register

Offset Address: 03-00h

Receive Buffer Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:0	RBR	RO	0	Any data bytes received by UART are accessed by reading this register

Offset Address: 03-00h

Transmitter holding register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:0	THR	WO	0	This register is used to buffer outgoing character.

Offset Address: 07-04h

Interrupt Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:4	-	RO	0	Reserved
3	MSI_EN	RW	0	Enable modem status interrupt.
2	RLSI_EN	RW	0	Receiver line status interrupt enable. 0 = Receiver line status interrupt is disabled. 1 = Receiver line status interrupt is enabled.
1	THRE_EN	RW	0	Transmitter holding register empty interrupt enable. 0 = Transmitter holding register empty interrupt is disabled. 1 = Transmitter holding register empty interrupt is enabled.
0	RDAI_EN	RW	0	Receiver data available interrupt and character timeout indication interrupt enable. 0 = Receiver data available interrupt and character timeout indication interrupt is disabled. 1 = Receiver data available interrupt and character timeout indication interrupt is enabled.

Offset Address: 0B-08h

Interrupt Identification Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:6	FIFO_USE	RO	00b	FIFO use indicator. 0 = Non-FIFO mode. 1 = Reserved. 2 = Reserved.

				3 = FIFOs are enabled. FIFOEN bit in the FIFO control register (FCR) is set to 1.
5:4	-	RO	0	Reserved
3:1	INT_TYPE	RO	000b	Interrupt type. 0 = Modem status. 1 = Transmitter holding register empty (priority 3). 2 = Receiver data available (priority 2). 3 = Receiver line status (priority 1, highest). 4 = Reserved. 5 = Reserved. 6 = Character timeout indication (priority 2). 7 = Reserved.
0	INT_PEND	RO	1	Interrupt pending. When any UART interrupt is generated and is enabled in IER, IPEND is forced to 0. IPEND remains 0 until all pending interrupts are cleared or until a hardware reset occurs. If no interrupts are enabled, IPEND is never forced to 0. 0 = Interrupts pending. 1 = No interrupts pending.

Interrupt Identification and Interrupt Clearing Information

Priority Level	IIR Bits				Interrupt Type	Interrupt Source	Event That Clears Interrupt
	3	2	1	0			
None	0	0	0	1	None	None	None
1	0	1	1	0	Receiver line status	Overrun error, parity error, framing error, or break is detected.	For an overrun error, reading the line status register (LSR) clears the interrupt. For a parity error, framing error, or break, the interrupt is cleared only after all the erroneous data have been read.
2	0	1	0	0	Receiver data-ready	Non-FIFO mode: Receiver data is ready.	Non-FIFO mode: The receiver buffer register (RBR) is read.
						FIFO mode: Trigger level reached. If four character times pass with no access of the FIFO, the interrupt is asserted again.	FIFO mode: The FIFO drops below the trigger level.
2	1	1	0	0	Receiver time-out	FIFO mode only: No characters have been removed from or input to the receiver FIFO during the last four character times, and there is at least one character in the receiver FIFO during this time.	One of the following events: • A character is read from the receiver FIFO. • A new character arrives in the receiver FIFO. • The URRST bit in the power management register (PMU) is loaded with 1.
3	0	0	1	0	Transmitter holding register empty	• Non-FIFO mode: Transmitter holding register (THR) is empty. • FIFO mode: Transmitter FIFO is empty.	A character is written to the transmitter holding register (THR).

Offset Address: 0B-08h

FIFO Control Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:6	RXFIFOTL	WO	00b	Receiver FIFO trigger level. RXFIFTL sets the trigger level for the receiver FIFO. When the trigger level is reached, a receiver data-ready interrupt is generated (if the interrupt request is enabled). When the FIFO drops below the trigger level, the interrupt is cleared. 0 = 1 byte. 1 = 4 bytes. 2 = 8 bytes. 3 = 14 bytes.
5:4	-	RO	0	Reserved
3	DMA_MODE	WO	0	DMA MODE1 enable if FIFOs are enabled. Always write 1 to DMAMODE1. After a hardware reset, change DMAMODE1 from 0 to 1. DMAMOD1 = 1 is a requirement for proper communication between the UART and the EDMA controller. 0 = DMA MODE1 is disabled. 1 = DMA MODE1 is enabled.
2	TXCLR	WO	0	Transmitter FIFO clear. Write a 1 to TXCLR to clear the bit. 0 = No effect. 1 = Clears transmitter FIFO and resets the transmitter FIFO counter. The shift register is not cleared.
1	RXCLR	WO	0	Receiver FIFO clear. Write a 1 to RXCLR to clear the bit. 0 = No effect. 1 = Clears receiver FIFO and resets the receiver FIFO counter. The shift register is not cleared.
0	FIFOEN	WO	0	Transmitter and receiver FIFOs mode enable. FIFOEN must be set before other FCR bits are written to or the FCR bits are not programmed. Clearing this bit clears the FIFO counters. 0 = Non-FIFO mode. The transmitter and receiver FIFOs are disabled, and the FIFO pointers are cleared. 1 = FIFO mode. The transmitter and receiver FIFOs are enabled.

Offset Address: 0F-0Ch

Line Control Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7	DLAB	RW	0	Divisor latch access bit. The divisor latch registers (DLL and DLH) can be accessed at dedicated addresses or at addresses shared by RBR, THR, and IER. Using the shared addresses requires toggling DLAB to change which registers are selected. If the dedicated addresses are used, DLAB can = 0. 0 = Allows access to the receiver buffer register (RBR), the transmitter holding register (THR), and the interrupt enable register (IER) selected.

				At the address shared by RBR, THR, and DLL, the CPU can read from RBR and write to THR. At the address shared by IER and DLH, the CPU can read from and write to IER. 1 = Allows access to the divisor latches of the baud generator during a read or write operation (DLL and DLH). At the address shared by RBR, THR, and DLL, the CPU can read from and write to DLL. At the address shared by IER and DLH, the CPU can read from and write to DLH.
6	BREAK_EN	RW	0	Break control. 0 = Break condition is disabled. 1 = Break condition is transmitted to the receiving UART. A break condition is a condition in which the uart_txd signal is forced to the spacing (cleared) state.
5	STICK_EN	RW	0	Stick parity enable. The STICK_EN bit works in conjunction with the EVEN_EN and PARITY_EN bits. The relationship between the STICK_EN, EVEN_EN, and PARITY_EN bits is summarized in below table. 0 = Stick parity is disabled. 1 = Stick parity is enabled. • When odd parity is selected (EVEN_EN = 0), the PARITY bit is transmitted and checked as set. • When even parity is selected (EVEN_EN = 1), the PARITY bit is transmitted and checked as cleared.
4	EVEN_EN	RW	0	Even parity select. Selects the parity when parity is enabled (PARITY_EN = 1). The EVEN_EN bit works in conjunction with the STICK_EN and PARITY_EN bits. The relationship between the STICK_EN, EVEN_EN, and PARITY_EN bits is summarized in below table. 0 = Odd parity is selected (an odd number of logic 1s is transmitted or checked in the data and PARITY bits). 1 = Even parity is selected (an even number of logic 1s is transmitted or checked in the data and PARITY bits)
3	PARITY_EN	RW	0	Parity enable. The PARITY_EN bit works in conjunction with the EVEN_EN and STICK_EN bits. The relationship between the SP, EPS, and PEN bits is summarized in below table. 0 = No PARITY bit is transmitted or checked. 1 = Parity bit is generated in transmitted data and is checked in received data between the last data word bit and the first STOP bit.
2	STOP_LEN	RW	0	Number of STOP bits generated. STOP_LEN specifies 1, 1.5, or 2 STOP bits in each transmitted character. When STOP_LEN = 1, the WORD_LEN bit determines the number of STOP bits. The receiver clocks only the first STOP bit, regardless of the number of STOP bits selected. The number of STOP bits generated is summarized in Table 3-10. 0 = One STOP bit is generated. 1 = STOP_LEN bit determines the number of STOP bits: » When STOP_LEN = 0, 1.5 STOP bits are generated. » When STOP_LEN = 1h, 2h, or 3h, 2 STOP bits are generated.
1:0	WORD_LEN	RW	11b	Word length select. Number of bits in each transmitted or received serial character. When STOP_LEN = 1, the WORD_LEN bit determines the number of STOP bits. 0 = 5 bits 1 = 6 bits 2 = 7 bits 3 = 8 bits

STICK_EN	EVEN_EN	PARITY_EN	Parity Option
x	x	0	Parity disabled: No PARITY bit is transmitted or checked.
0	0	1	Odd parity selected: Odd number of logic 1s.
0	1	1	Even parity selected: Even number of logic 1s.
1	0	1	Stick parity selected with PARITY bit transmitted and checked as set.
1	1	1	Stick parity selected with PARITY bit transmitted and checked as cleared.

Offset Address: 13-10h

Modem Control Register

Bit	Field Name	Attribute	Default	Field Description
31:7	-	RO	0	Reserved
6	RTS_TRI_MODE	RW	0	RTS Trigger Mode 0: For trigger level 1, 4 and 8, uart_rts_n is automatically reasserted after the receiver FIFO is empty. 1: For trigger level 1, 4 and 8, uart_rts_n is automatically reasserted when the receiver FIFO drops below the trigger level.
5	AUTOFLOW_EN	RW	0	Autoflow control enable.
4	LOOPBACK_EN	RW	0	Loopback mode enable. LOOPBACK_EN is used for the diagnostic testing using the loopback feature. 0 = Loopback mode is disabled. 1 = Loopback mode is enabled. When LOOPBACK_EN is set, the following occur: » The uart_txd signal is set high. » The uart_rxd pin is disconnected » The output of the transmitter shift register (TSR) is looped back in to the receiver shift register (RSR) input.
3	AUX2	RW	0	AUX2 Control Bit. When in loopback mode, it is connected to DCD
2	AUX1	RW	0	AUX1 Control Bit. When in loopback mode, it is conntected to RI
1	RTS_CTRL	RW	0	When autoflow control is disable, RTS is controled by software 1: RTS is “0” 0: RTS is “1”
0	DTR_CTRL	RW	0	DTR control, when in loopback mode, it is connected to DSR 1: DTR is “0” 0: DTR is “1”

Offset Address: 17-14h

Line Status Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7	RECEIVER_ERRO R	RO	0	Receiver FIFO error. In non-FIFO mode: 0 = There has been no error, or RXFIFOE was cleared because the CPU read the erroneous character from the receiver buffer register

				(RBR). 1 = There is a parity error, framing error, or break indicator in the receiver buffer register (RBR). In FIFO mode: 0 = There has been no error, or RXFIFOE was cleared because the CPU read the erroneous character from the receiver FIFO and there are no more errors in the receiver FIFO. 1 = At least one parity error, framing error, or break indicator in the receiver FIFO.
6	TRANSMITTER_EMPTY	RO	1	Transmitter empty (TEMT) indicator. In non-FIFO mode: 0 = Either the transmitter holding register (THR) or the transmitter shift register (TSR) contains a data character. 1 = Both the transmitter holding register (THR) and the transmitter shift register (TSR) are empty. In FIFO mode: 0 = Either the transmitter FIFO or the transmitter shift register (TSR) contains a data character. 1 = Both the transmitter FIFO and the transmitter shift register (TSR) are empty.
5	THR_EMPTY	RO	1	Transmitter holding register empty (THRE) indicator. If the THRE bit is set and the corresponding interrupt enable bit is set (THRE_EN = 1 in IER), an interrupt request is generated. In non-FIFO mode: 0 = Transmitter holding register (THR) is not empty. THR has been loaded by the CPU. 1 = Transmitter holding register (THR) is empty (ready to accept a new character). The content of THR has been transferred to the transmitter shift register (TSR). In FIFO mode: 0 = Transmitter FIFO is not empty. At least one character has been written to the transmitter FIFO. The transmitter FIFO may be written to if it is not full. 1 = Transmitter FIFO is empty. The last character in the FIFO has been transferred to the transmitter shift register (TSR).
4	BREAK_ERROR_STS	RO	0	Break indicator. The BI bit is set whenever the receive data input was held low for longer than a full-word transmission time. A full-word transmission time is defined as the total time to transmit the START, data, PARITY, and STOP bits. If the BI bit is set and the corresponding interrupt enable bit is set (RLSI_EN = 1 in IER), an interrupt request is generated. In non-FIFO mode: 0 = No break has been detected, or the BI bit was cleared because the CPU read the erroneous character from the receiver buffer register (RBR). 1 = A break has been detected with the character in the receiver buffer register (RBR). In FIFO mode: 0 = No break has been detected, or the BI bit was cleared because the CPU read the erroneous character from the receiver FIFO and the next character to be read from the FIFO has no break indicator. 1 = A break has been detected with the character at the top of the receiver FIFO.
3	FRAME_ERROR_STS	RO	0	Framing error (FE) indicator. A framing error occurs when the received character does not have a valid STOP bit. In response to a framing error, the UART sets the FE bit and waits until the signal on the RX pin goes high. When the RX signal goes high, the receiver is ready to

				detect a new START bit and receive new data. If the FE bit is set and the corresponding interrupt enable bit is set (RLSI_EN = 1 in IER), an interrupt request is generated. In non-FIFO mode: 0 = No framing error has been detected, or the FE bit was cleared because the CPU read the erroneous data from the receiver buffer register (RBR). 1 = A framing error has been detected with the character in the receiver buffer register (RBR). In FIFO mode: 0 = No framing error has been detected, or the FE bit was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no framing error. 1 = A framing error has been detected with the character at the top of the receiver FIFO.
2	PARITY_ERROR_STS	RO	0	Parity error (PE) indicator. A parity error occurs when the parity of the received character does not match the parity selected with the EVEN_EN bit in the line control register (LCR). If the PE bit is set and the corresponding interrupt enable bit is set (RLSI_EN = 1 in IER), an interrupt request is generated. In non-FIFO mode: 0 = No parity error has been detected, or the PE bit was cleared because the CPU read the erroneous data from the receiver buffer register (RBR). 1 = A parity error has been detected with the character in the receiver buffer register (RBR). In FIFO mode: 0 = No parity error has been detected, or the PE bit was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no parity error. 1 = A parity error has been detected with the character at the top of the receiver FIFO.
1	OVERRUN_ERROR_STS	RO	0	Overrun error (OE) indicator. An overrun error in the non-FIFO mode is different from an overrun error in the FIFO mode. If the OE bit is set and the corresponding interrupt enable bit is set (RLSI_EN = 1 in IER), an interrupt request is generated. In non-FIFO mode: 0 = No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR). 1 Overrun error has been detected. Before the character in the receiver buffer register (RBR) could be read, it was overwritten by the next character arriving in RBR. In FIFO mode: 0 = No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR). 1 = Overrun error has been detected. If data continues to fill the FIFO beyond the trigger level, an overrun error occurs only after the FIFO is full and the next character has been completely received in the shift register. An overrun error is indicated to the CPU as soon as it happens. The new character overwrites the character in the shift register, but it is not transferred to the FIFO.
0	DATA_READY	RO	0	Data-ready (DR) indicator for the receiver. If the DR bit is set and the corresponding interrupt enable bit is set (RDAI_EN = 1 in IER), an interrupt request is generated. In non-FIFO mode: 0 = Data is not ready, or the DR bit was cleared because the character was read from the receiver buffer register (RBR). 1 = Data is ready. A complete incoming character has been received

				and transferred into the receiver buffer register (RBR). In FIFO mode: 0 = Data is not ready, or the DR bit was cleared because all of the characters in the receiver FIFO have been read. 1 = Data is ready. There is at least one unread character in the receiver FIFO. If the FIFO is empty, the DR bit is set as soon as a complete incoming character has been received and transferred into the FIFO. The DR bit remains set until the FIFO is empty again.

Offset Address: 1B-18h

Modem Status Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7	MSR_DCD	RO	0	Complement of the Carrier Detect input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 3 (AUX2).
6	MSR_RI	RO	0	Complement of the Ring Indicator input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 2 (AUX1).
5	MSR_DSR	RO	0	Complement of the Data Set Ready input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 0 (DTR).
4	MSR_CTS	RO	0	Complement of the Clear To Send input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 1 (RTS).
3	DELTA_DCD_STS	RO	0	Change in DCD indicator bit. DCD indicates that the DCD input has changed state since the last time it was read by the CPU. When DCD is set and the modem status interrupt is enabled, a modem status interrupt is generated.
2	DELTA_RI_STS	RO	0	Trailing edge of RI (TERI) indicator bit. TERI indicates that the RI input has changed from a low to a high. When TERI is set and the modem status interrupt is enabled, a modem status interrupt is generated.
1	DELTA_DSR_STS	RO	0	Change in DSR indicator bit. DDSR indicates that the DSR input has changed state since the last time it was read by the CPU. When DDSR is set and the modem status interrupt is enabled, a modem status interrupt is generated.
0	DELTA_CTS_STS	RO	0	Change in CTS indicator bit. DCTS indicates that the CTS input has changed state since the last time it was read by the CPU. When DCTS is set (autoflow control is not enabled and the modem status interrupt is enabled), a modem status interrupt is generated. When autoflow control is enabled, no interrupt is generated.

Offset Address: 1F-1Ch

Scratch Pad Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:0	SCR	RW	0	These bits are intended for the programmer's use as a scratch pad in the sense that it temporarily holds the programmer's data without affecting any other UART operation.

Offset Address: 23-20h

Divisor LSB Latch

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:0	DLL	RW	0	The 8 least-significant bits (LSBs) of the 16-bit divisor. When OSM_SEL=0, Desired baud rate = input clock frequency/(({DLH, DLL}+1) x 16). When OSM_SEL=1, Desired baud rate = input clock frequency/(({DLH, DLL}+1) x 13).

Offset Address: 27-24h

Divisor MSB Latch

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:0	DLH	RW	0	The 8 most-significant bits (MSBs) of the 16-bit divisor. When OSM_SEL=0, Desired baud rate = input clock frequency/(({DLH, DLL}+1) x 16). When OSM_SEL=1, Desired baud rate = input clock frequency/(({DLH, DLL}+1) x 13).

Offset Address: 2B-28h

FIFO Status Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	RO	0	Reserved
23:21	-	RO	0	Reserved
20:16	RX_FIFO_LEN	RO	0	Receiver FIFO length indicator bit
15:13	-	RO	0	Reserved
12:8	TX_FIFO_LEN	RO	0	Transmitter FIFO length indicator bit.

7:4	-	RO	0	Reserved
3	RX_FIFO_FULL	RO	0	Receiver FIFO full indicator bit
2	RX_FIFO_EMPTY	RO	1	Receiver FIFO empty indicator bit
1	TX_FIFO_FULL	RO	0	Transmitter FIFO full indicator bit.
0	TX_FIFO_EMPTY	RO	1	Transmitter FIFO empty indicator bit.

Offset Address: 2F-2Ch

Debug Signal Register

Bit	Field Name	Attribute	Default	Field Description
31:13	-	RO	0	Reserved
12	RX_FIFO_TIMEOUT	RO	0	Indicate no access to Receiver FIFO, that leads to timeout
11	RX_BUF_FULL	RO	0	Receiver buffer full indicator bit
10	TX_BUF_EMPTY	RO	1	Transmitter buffer empty indicator bit
9	RX_BAUDRATE	RO	0	Receiver baud-rate
8	TX_BAUDRATE	RO	0	Transmitter baud-rate
7:5	RX_CUR_STATE	RO	0	Receiver FSM
4	RXD_WORK	RO	0	Indicate Receiver is working
3:1	TX_CUR_STATE	RO	0	Transmitter FSM
0	TXD_WORK	RO	0	Indicate Transmitter is working

Offset Address: 33-30h

Power Management Register

Bit	Field Name	Attribute	Default	Field Description
31:2	-	RO	0	Reserved
1	TXRST	RW	0	UART transmitter reset. Resets and enables the transmitter. 1 = Transmitter is disabled and in reset state. 0 = Transmitter is enabled.
0	RXRST	RW	0	UART receiver reset. Resets and enables the receiver. 1 = Receiver is disabled and in reset state. 0 = Receiver is enabled.

Offset Address: 37-34h

Mode Definition Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	OSM_SEL	RW	0	Over-Sampling Mode Select. 0 = 16× oversampling. 1 = 13× oversampling.

12 Inter-integrated Circuit Interface (I2C)

12.1 Overview

I2C (inter-integrated circuit) bus Interface serves as an interface between the microcontroller and the serial I2C bus. It provides multimaster capability, and controls all I2C bus-specific sequencing, protocol, arbitration and timing. It supports the standard mode (Sm, up to 100 kHz) and Fm mode (Fm, up to 400 kHz) and HS mode.

Depending on specific device implementation DMA capability can be available for reduced CPU overload.

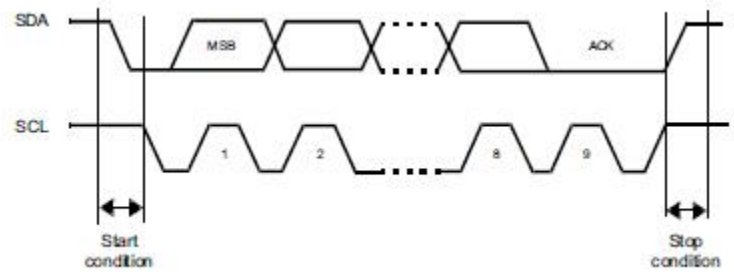
In Master mode, the I2C interface initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated in master mode by software.

In Slave mode, the interface is capable of recognizing its own addresses (7 or 10-bit), and the General Call address. The General Call address detection may be enabled or disabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the start condition contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter.

Figure 16 I2C Bus Protocol



Acknowledge may be enabled or disabled by software. The I2C interface addresses (addressing 7-bit/ 10-bit or general call address) can be selected by software.

12.1.1 Feature List

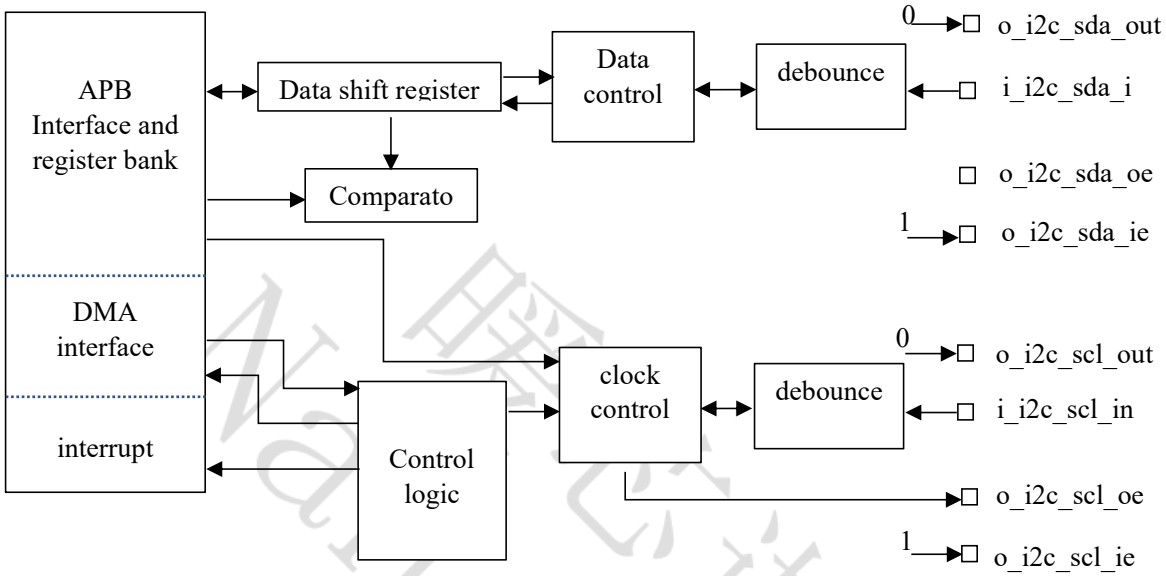
- Compliance with AMBA APB specifications
- Supports I2C Philips standard
- The same interface can act as Master or Slave

- Half-duplex communication (only transmitter or receiver)
- Generation and detection of 7-bit/10-bit addressing
- Supports different communication speeds:
  - Standard Speed (up to 100 kHz)
  - Fast Speed (up to 400 kHz)
- Analog noise filter
- Status flags:
  - Transmitter/Receiver mode flag
  - End-of-Byte transmission flag
  - I2C busy flag
- Data direction is always MSB first
- Error flags:
  - Arbitration lost condition for master mode
  - Acknowledgment failure after address/ data transmission
  - Detection of misplaced start or stop condition
- 1-byte buffer with DMA capability

12.2 **Block Diagram**

The block diagram of the I2C interface is shown as below.

Figure 17 I2C Block Diagram



12.3 Function Description

12.3.1 Supported Mode

The interface can operate in one of the four following modes:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

12.3.2 I2C Slave Mode

By default the I2C interface operates in Slave mode. To switch from default Slave mode to Master mode a Start condition generation is needed.

The peripheral input clock must be programmed in the I2C_CR2 register in order to generate correct timings.

Slave transmitter

(1) Configure interrupt of I2C_CR2 register;

(2) Configure PE of I2C_CR1 to enable I2C interface;

(3) Configure I2C_OAR to set I2C slave address;

(4) Configure ACK of I2C_CR1 to set ack bit;

(5) Read ADDR of I2C_SR to clear addr bit;

(6) Write I2C_DR register;

(7) Read AF and STOPF of I2C_SR ;

(8) Write PE of I2C_CR1 to disable I2C interface.

If TxE is set and some data were not written in the I2C_DR register before the end of the next data transmission, the BTF bit is set and the interface waits until BTF is cleared by a read to I2C_SR followed by a write to the I2C_DR register, stretching SCL low.

Slave receiver

(1) Configure interrupt of I2C_CR2 register;

(2) Configure PE of I2C_CR1 to enable I2C interface;

(3) Configure I2C_OAR to set I2C slave address;

(4) Configure ACK of I2C_CR1 to set ack bit;

(5) Read ADDR of I2C_SR to clear addr bit;

(6) RXNE of I2C_SR is set, read I2C_DR to clear RXNE bit;

(7) Read STOPF of I2C_SR ;

(8) Write PE of I2C_CR1 to disable I2C interface.

If RxNE is set and the data in the DR register is not read before the end of the next data reception, the BTF bit is set and the interface waits until BTF is cleared by a read from the I2C_DR register, stretching SCL low.

12.3.3 I2C Master Mode

In Master mode, the I2C interface initiates a data transfer and generates the clock signal. A serial data transfer always begins with a Start condition and ends with a Stop condition. Master mode is selected as soon as the Start condition is generated on the bus with a START bit.

Master transmitter

(1) Configure interrupt and clock frequency of I2C_CR2 register;

(2) Configure PE of I2C_CR1 to enable I2C interface;

(3) Configure START of I2C_CR1 to start I2C master;

(4) Read SB of I2C_SR;

(5) Write I2C_DR register with address to clear SB, in 7-bit master transmitter writing slave address, in 10-bit master transmitter writing 10-bit address header;

(6) in 7-bit master transmitter read ADDR of I2C_SR to clear, in 10-bit master transmitter read ADDR10 of I2C_SR and write I2C_DR with slave address to clear ADDR10.

(7) in 7-bit master transmitter write I2C_DR with data, in 10-bit master transmitter read ADDR of I2C_SR to clear and write I2C_DR with data;

(8) Read BTF of I2C_SR;

(9) Write I2C_CR1 to set STOP bit.

If TxE is set and a data byte was not written in the DR register before the end of the last data transmission, BTF is set and the interface waits until BTF is cleared by a write to I2C_DR, stretching SCL low.

Master receiver

(1) Configure interrupt and clock frequency of I2C_CR2 register;

(2) Configure PE of I2C_CR1 to enable I2C interface;

(3) Configure START of I2C_CR1 to start I2C master;

(4) Read SB of I2C_SR;

(5) Write I2C_DR register with address to clear SB, in 7-bit master writing slave address, in 10-bit master writing 10-bit address header;

(6) in 7-bit master read ADDR of I2C_SR to clear, in 10-bit master read ADDR10 of I2C_SR and write I2C_DR with slave address to clear ADDR10 and write I2C_CR1 to set START.

(7) in 7-bit master read RXNE of I2C_SR and read 2C_DR, in 10-bit master read ADDR of I2C_SR to clear and write I2C_CR1 to set START bit;

(8) in 7-bit master if receive last data, RxNE=1 cleared by reading I2C_DR register, write ACK=0 and STOP of I2C_CR1; in 10-bit master read SB of I2C_SR and write I2C_DR register with 10-bit address header to clear SB;

(9) in 10-bit master read ADDR of I2C_SR to clear;

(10) in 10-bit master read RXNE of I2C_SR and read 2C_DR;

(11) in 10-bit master if receive last data, RxNE=1 cleared by reading I2C_DR register, write ACK=0 and STOP of I2C_CR1;

(12) RxNE=1 cleared by reading I2C_DR register;

12.3.4 Error Conditions

The following are the error conditions which may cause communication to fail.

(1) Bus error (BERR)

(2) Acknowledge failure (AF)

(3) Arbitration lost (ARLO)

(4) Overrun/underrun error (OVR)

12.3.5 I2C Interrupt

Table 26 Interrupt Table

Interrupt event	Event Flag	Enable Control bit
Start bit sent (Master)	SB	ITEVFEN
Address sent (Master) or Address matched (Slave)	ADDR	
10-bit header sent (Master)	ADD10	
Stop received (Slave)	STOPF	
Data byte transfer finished	BTF	
Receive buffer not empty	RxNE	ITEVFEN and ITBUFEN
Transmit buffer empty	TxE	

Bus error	BERR	ITERREN
Arbitration loss (Master)	ARLO	
Acknowledge failure	AF	
Overrun/Underrun	OVR	

12.4 **Registers**

The peripheral registers have to be accessed by byte (8 bits), half-words (16 bits) or words (32 bits). These registers are available at 32-bit addresses in the device memory map. See the device-specific data manual for the memory address of these registers.

Table 27 I2C Registers

Offset	Acronym	Register Description
00h	I2C_DR	I2C Data Register
04h	I2C_OAR	I2C Own Address Register
08h	I2C_CR1	I2C Control Register1
0Ch	I2C_CR2	I2C Control Register2
10h	I2C_SR	I2C Status Register
14h	I2C_DBG	I2C Debug Data Register

Offset Address: 03-00h

I2C Data Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7:0	DATA	RW	0	Byte received or to be transmitted to the bus

Offset Address: 07-04h

I2C Own Address Register

Bit	Field Name	Attribute	Default	Field Description
31:10	-	-	0	Reserved
9:8	ADD[9:8]	RW	0	7-bit addressing mode: don't care 10-bit addressing mode: bits9:8 of address
7:1	ADD[7:1]	RW	0	bits 7:1 of address
0	ADD0	RW	0	7-bit addressing mode: don't care 10-bit addressing mode: bit 0 of address

Offset Address: 0B-08h

I2C Control Register 1

Bit	Field Name	Attribute	Default	Field Description
31:16	-	-	0	Reserved
15	SWRST	RW	0	Software reset
14:8	-	-	0	Reserved
7	DBYPASS	RW	0	Debounce bypass 0: bypass disable 1: bypass enable
6	MMASTER	RW	0	Multimaster enable 0: multimaster disable 1: multimaster enable
5	ACK	RW	0	Acknowledge enable 0: No acknowledge returned 1: Acknowledge returned after a byte is received (matched address or data)
4	STOP	RW	0	Stop generation In Master Mode: 0: No Stop generation. 1: Stop generation after the current byte transfer or after the current Start condition is sent. In Slave mode: 0: No Stop generation. 1: Release the SCL and SDA lines after the current byte transfer.
3	START	RW	0	Start generation In Master Mode: 0: No Start generation 1: Repeated start generation In Slave mode: 0: No Start generation

				1: Start generation when the bus is free
2	NOSTRETCH	RW	0	Clock stretching disable (Slave mode) 0: Clock stretching enabled 1: Clock stretching disabled
1	ENGC	RW	0	General call enable 0: General call disabled. Address 00h is NACKed. 1: General call enabled. Address 00h is ACKed
0	PE	RW	0	Peripheral enable 0: Peripheral disable 1: Peripheral enable

Offset Address: 0F-0Ch

I2C Control Register 2

Bit	Field Name	Attribute	Default	Field Description
31:11	-	-	0	Reserved
10	LAST	RW	0	DMA last transfer 0: Next DMA EOT is not the last transfer 1: Next DMA EOT is the last transfer
9	DMAEN	RW	0	DMA requests enable 0: DMA requests disabled 1: DMA request enabled when TxE=1 or RxNE =1
8	ITBUFEN	RW	0	Buffer interrupt enable 0: TxE = 1 or RxNE = 1 does not generate any interrupt. 1: TxE = 1 or RxNE = 1 generates Event Interrupt (whatever the state of DMAEN)
7	ITEVTEN	RW	0	Event interrupt enable 0: Event interrupt disabled 1: Event interrupt enabled This interrupt is generated when: – SB = 1 (Master)

				– ADDR = 1 (Master/Slave) – ADD10= 1 (Master) – STOPF = 1 (Slave) – BTF = 1 with no TxE or RxNE event – TxE event to 1 if ITBUFEN = 1 – RxNE event to 1if ITBUFEN = 1
6	ITERREN	RW	0	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled This interrupt is generated when: – BERR = 1 – ARLO = 1 – AF = 1 – OVR = 1
5:0	FREQ_DIV	RW	0	clock frequency divider Standard-mode 6`b000000: 10 kHz 6`b000001: 20 kHz ... 6`b001001: 100 kHz Fast-mode 6`b001010: 110 kHz 6`b001011: 120 kHz ... 6`b100111: 400 kHz High-speed mode 6`b101000: 500 kHz 6`b101001: 600 kHz ... 6`b101110: 1.5 MHz

Offset Address: 13-10h

I2C Status Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	-	0	Reserved
15	GENCALL	RO	0	General call address (Slave mode) 0: No General Call 1: General Call Address received when ENGC=1
14	TRA	RO	0	Transmitter/receiver 0: Data bytes received 1: Data bytes transmitted
13	BUSY	RO	0	Bus busy 0: No communication on the bus 1: Communication ongoing on the bus
12	MSL	RO	0	Master/slave 0: Slave Mode 1: Master Mode
11	OVR	RC_W0	0	Overrun/Underrun 0: No overrun/underrun 1: Overrun or underrun
10	AF	RC_W0	0	Acknowledge failure 0: No acknowledge failure 1: Acknowledge failure
9	ARLO	RC_W0	0	Arbitration lost (master mode) 0: No Arbitration Lost detected 1: Arbitration Lost detected
8	BERR	RC_W0	0	Bus error 0: No misplaced Start or Stop condition 1: Misplaced Start or Stop condition
7	-	-	0	Reserved
6	TXE	RO	0	Data register empty (transmitters)

				0: Data register not empty 1: Data register empty
5	RXNE	RO	0	Data register not empty (receivers) 0: Data register empty 1: Data register not empty
4	STOPF	RO	0	Stop detection (slave mode) 0: No Stop condition detected 1: Stop condition detected
3	ADD10	RO	0	bit header sent (Master mode) 0: No ADD10 event occurred. 1: Master has sent first address byte (header).
2	BTF	RO	0	Byte transfer finished 0: Data byte transfer not done 1: Data byte transfer succeeded
1	ADDR	RO	0	Address sent (master mode)/matched (slave mode) Address matched (Slave) 0: Address mismatched or not received. 1: Received address matched. Address sent (Master) 0: No end of address transmission 1: End of address transmission
0	SB	RO	0	Start bit (Master mode) 0: No Start condition 1: Start condition generated.

Offset Address: 17-14h

I2C Debug Data Register

Bit	Field Name	Attribute	Default	Field Description

31:22	APB_IF_DBG	RO	0	APB interface debug data
21:11	I2CS_DBG	RO	0	I2C slave debug data
10:0	I2CM_DBG	RO	0	I2C master debug data

13 Real-time Clock (RTC)

13.1 Overview

The Real-Time Clock (RTC) provides an automatic wakeup to manage low-power mode. The RTC is an independent BCD timer/counter. Its main function is to keep track of the current time. The RTC provides a time-of-day clock/calendar with programmable alarm interrupts.

13.1.1 Feature List

- Counts seconds, minutes, hours, day (day of week), date (day of month), month and year
- Compensations for 28-, 29-(leap year), 30-, and 31-day months are performed
- Binary or BCD representation of time, calendar and alarm
- 12- or 24-hour clock with AM and PM in 12-hour mode
- Two interrupts are separately software maskable

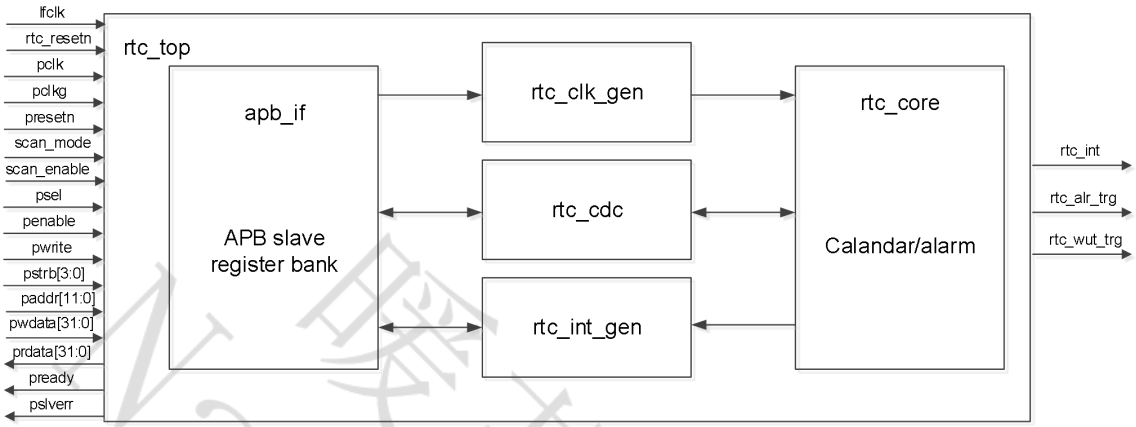
- time-of-day clock/calendar with programmable alarm interrupt

- periodic programmable timer wakeup interrupt

13.2 **Block Diagram**

A functional block diagram of the RTC is shown as below.

Figure 18 RTC Block Diagram



rtc_apb_if as apb slave module is responsible for generating register bank

rtc_clk_gen generates 1Hz rtc clock output rtc_core based on prescaler

rtc_cdc handles clock domain crossing between pclk and rtc clock

rtc_int_gen generates rtc interrupt status including alarm and periodic wakeup

rtc_core is the main module for calculating time/calendar including century, year, day-of-month, day-of-week, hour, minute, second. Each counter can be updated when programming the time/date register.

13.3 Function Description

13.3.1 Clock and Prescaler

The RTC clock source (lfclk) is selected through the clock controller among the RC oscillator clock and the crystal oscillator clock.

A programmable prescaler locates at the offset 0x1C stage generates a 1Hz clock which is used to update the calendar.

The divided RTC clock is given by the following formula:

$$F_{rtc\_clk} = \frac{F_{lfclk}}{(prescaler + 1)}$$

Default prescaler setting is 0x7FFF, when lfclk frequency is 32.768KHz, then the divided RTC clock is 1Hz.

13.3.2 Real-time clock and calendar

The RTC calendar time and date registers are accessed through RTC_TR and RTC_DR registers which are synchronized with pclk (APB clock). RTC_TR for the time, RTC_DR for the date.

Every RTC clock, the current calendar value is copied into RTC_TR and RTC_DR registers. The copy is not performed in stop or standby mode.

When in initial mode, the current calendar will not be updated to RTC_TR and RTC_DR registers. After exiting initial mode, then current calendar will be updated to RTC_TR and RTC_DR registers.

13.3.3 Time/Calendar/Alarm Data Mode

The time, calendar and alarm registers can be access from RTC address offset 0x00 to 0x0D. The contents of time, calendar and alarm bytes can be either Binary or Binary-Coded Decimal (BCD) format. Before writing the internal time, calendar and alarm registers, the INIT bit in Control Register should be written to a logic one, after update finish, then write INIT bit to a logic zero. Once initial finish, the data mode and hour mode cant be changed any more. When 12-hour format is selected, the high order bit of the hours byte represents PM when it is a logic one.

Table 28 RTC Data Mode

Address Location	Function	Decimal Range	Range	
			Binary Data Mode	BCD Data Mode
0x00	Seconds	0-59	00-3B	00-59
0x01	Minutes	0-59	00-3B	00-59
0x02	Hours-12-hr Mode	1-12	01-0C AM, 81-8C PM	01-12AM, 81-92PM
	Hours-24-hr Mode	0-23	00-17	00-23
0x03	Day of the Week	1-7	01-07	01-07
0x04	Day of the Month	1-31	01-1F	01-31
0x05	Month	1-12	01-0C	01-12
0x06	Year	0-99	00-63	00-99
0x07	Century	0-99	00-63	00-99
0x08	Seconds Alarm	0-59	00-3B	00-59
0x09	Minutes Alarm	0-59	00-3B	00-59
0x0A	Hours Alarm-12-hr	1-12	01-0C AM, 81-8C PM	01-12AM, 81-92PM
	Hours Alarm-24-hr	0-23	00-17	00-23
0x0C	Day of the Month Alarm	1-31	01-1F	01-31
0x0D	Month Alarm	1-12	01-0C	01-12

13.3.4 Programmable Alarm

The programmable alarm function is enabled through ALARM_EN bit in the RTC_CR register. The ALARM_STS in RTC_ISR is set to 1 if the calendar seconds, minutes, hours, day-of-month, month match the values in the alarm registers RTC_TAR and RTC_DAR. The ALARM_STS is cleared by writing 1 to this register.

Another use condition is to insert a “don’t care” state in one or more alarm bytes. For seconds, minutes and hours alarm, the “don’t care” code is any hexadecimal value from 0xC0 to 0xFF. For day-of-month alarm, if the alarm setting is not at range between 0x01-0x31 in BCD format and 0x01-0x1F in binary format, treated as “don’t care” code. For month alarm, if the alarm setting is not at range between 0x01-0x12 in BCD format and 0x01-0x0C in binary format, treated as “don’t care” code.

The alarm interrupt is enabled through ALARM_IE in the RTC_IER register.

13.3.5 Periodic Wakeup

The periodic wakeup flag is generated by a 16-bit programmable auto-reload up-counter. The periodic wakeup function is enabled through the WUT_EN bit in the RTC_CR register.

The wakeup timer clock input can be selected by configuring WUT_CLK_SEL in RTC_CR register:

When WUT_CLK_SEL is 0: The wakeup timer clock input is divided by 1 to 16 by configuring WUT_PRES. When LFCLK is 32.768kHz, this allows to configure the wakeup interrupt period from 30us to 32s.

When WUT_CLK_SEL is 1: The wakeup timer clock input is rtc_clk that is 1Hz internal clock, this allows to achieve a wakeup from 1s to around 36 hours with one-second resolution.

The periodic timer value is initialized by writing RTC_PTR register, when read from RTC_PTR, indicate the current periodic timer value.

The WUT_STS in RTC_ISR is set to 1 if the current periodic timer reaches the setting value. The WUT_STS is cleared by writing 1 to this register.

The periodic wakeup interrupt is enabled through WUT_IE in the RTC_IER register.

13.3.6 Initialization and Configuration

Calendar initialization

To program the initial time and data calendar values, including the time format and the prescaler configuration, the following sequence is required

1. To generate a 1Hz clock for the calendar counter, program rtc prescaler.

2. Set INIT bit to 1 and configure DATA_MODE and HOUR_MODE in the RTC_CR register to enter initialization mode. In this mode, the calendar counter is stopped and its value can be updated.

3. Load the initial time and date values in the RTC_TR and RTC_DR register.

4. Exit the initialization mode by clearing the INIT bit. After 2~3 cycle RTC clock(due to synchronization from pclk to rtc_clk), the actual counter value is loaded, then calendar starts counting.

5. Note: When INIT_SYNC_READY bit in RTC_SR register is zero, software cant set INIT bit to trigger another initialization. To read the calendar after initialization, software must check the INIT_SYNC_READY bit is set.

Programming the alarm

A similar procedure must be followed to program or update the programmable alarms.

1. Program the time alarm RTC_TAR and date alarm RTC_DAR registers.
2. Set ALARM_EN in RTC_CR register to enable alarm.

If need to generate interrupt, Set ALARM_IE in RTC_IER register.

Programming the periodic wakeup timer

The following sequence is required to configure or change the wakeup timer value.

1. Program the periodic wakeup input clock select in RTC_CR register
2. Program the periodic wakeup clock prescaler in RTC_WPR register.
3. Program the periodic wakeup timer value in RTC_WTR register
4. Set WUT_EN in RTC_CR register to enable the wakeup timer.
5. If need to generate interrupt, Set WUT_IE in RTC_IER register.

13.4 Registers

Table 29 RTC Registers

Offset	Acronym	Register Description
00h	RTC_TR	RTC Time Register
04h	RTC_DR	RTC Date Register
08h	RTC_TAR	RTC Time Alarm Register
0Ch	RTC_DAR	RTC Date Alarm Register
10h	RTC_IER	RTC Interrupt Enable Register
14h	RTC_ISR	RTC Interrupt Status Register
18h	RTC_CR	RTC Control Register
1Ch	RTC_PR	RTC Prescaler Register
20h	RTC_WTR	RTC Wakeup Timer Register
24h	RTC_WPR	RTC Wakeup Prescaler Register
28h	RTC_SR	RTC Status Register

Offset Address: 03-00h

RTC Time Register

Bit	Field Name	Attribute	Default	Field Description
31:24	REG_DAY_WEEK	RW	01h	When write, the updated day of week will be written in When read, indicates the current day of week
23:16	REG_HOUR	RW	0	When write, the updated hour will be written in When read, indicates the current hour
15:8	REG_MIN	RW	0	When write, the updated minute will be written in When read, indicates the current minute
7:0	REG_SEC	RW	0	When write, the updated second will be written in When read, indicates the current second

Offset Address: 07-04h

RTC Date Register

Bit	Field Name	Attribute	Default	Field Description
31:24	REG_CEN	RW	20h	When write, the updated century will be written in When read, indicates the current century
23:16	REG_YEAH	RW	21h	When write, the updated yeah will be written in When read, indicates the current yeah

15:8	REG_MON	RW	01h	When write, the updated month will be written in When read, indicates the current month
7:0	REG_DAY_MON	RW	01h	When write, the updated day of month will be written in When read, indicates the current day of month

Offset Address: 0B-08h

RTC Time Alarm Register

Bit	Field Name	Attribute	Default	Field Description
31:24	-	R0	0	Reserved
23:16	REG_HOUR_ALM	RW	0	Indicate hours alarm setting
15:8	REG_MIN_ALM	RW	0	Indicate minutes alarm setting
7:0	REG_SEC_ALM	RW	0	Indicate seconds alarm setting

Offset Address: 0F-0Ch

RTC Date Alarm Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:8	REG_MON_ALM	RW	01h	Indicate month alarm setting
7:0	REG_DAY_ALM	RW	01h	Indicate day of month alarm setting

Offset Address: 13-10h

RTC Interrupt Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:2	-	R0	0	Reserved
1	WUT_IE	RW	0	Period wakeup timer interrupt enable bit 0: disabled 1: enabled
0	ALARM_IE	RW	0	Alarm interrupt enable bit 0: disabled 1: enabled

Offset Address: 17-14h

RTC Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:2	-	RO	0	Reserved
1	WUT_STS	RW1C	0	Period wakeup timer interrupt status, this bit can be cleared by writing 1 to this bit
0	ALARM_STS	RW1C	0	Alarm interrupt status, this bit can be cleared by writing 1 to this bit

Offset Address: 1B-18h

RTC Control Register

Bit	Field Name	Attribute	Default	Field Description
31:7	-	RO	0	Reserved
5	WUT_CLK_SEL	RW	0	Periodic waekup timer input clock select: 0: divided clock by 1 to 16 of lfclk 1: 1hz internal clock
4	WUT_EN	RW	0	Periodic wakeup timer enable bit 0: disabled 1: enabled
3	ALARM_EN	RW	0	Alarm enable bit 0: disabled 1: enabled
2	DATA_MODE	RW	0	The Data Mode (DM) bit indicates whether time and calendar information is in binary or BCD format. The DM bit is set by the program to the appropriate format and can be read as required. A one in DM signifies binary data while a zero in DM specifies Binary Coded Decimal (BCD) data.
1	HOUR_MODE	RW	0	HOURS 24/12 Mode The 24/12 control bit establishes the format of the hours byte. A one indicates the 24-hour mode and a zero indicates the 12-hour mode.
0	INIT	RW	0	When the INIT bit is written to a one, the program can initialize the time and calendar bytes. After initialize done, then INIT bit is written to a zero

Offset Address: 1F-1Ch

RTC Prescaler Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:0	PRESCALER	RW	7FFFh	RTC clock prescaler, so when LFCLK is 32.768KHz, default setting is to divide rtc clock to 1Hz Frtc_clk = Flfclk/(PRESCALER+1)

Offset Address: 23-20h

RTC Wakeup Timer Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:0	WUT_VAL	RW	FFFFh	When write, the initial periodic timer value will be written in, default is FFFFh When read, indicates the current period timer. Only when non-zero value is valid, and can generate periodic wakeup interrupt

Offset Address: 27h to 24h

RTC Wakeup Prescaler Register

Bit	Field Name	Attribute	Default	Field Description
31:4	-	RO	0	Reserved
3:0	WUT_PRES	RW	0	Periodic wakeup clock prescaler, divided by 1 to 16 Fwut_clk = Flfclk/(WUT_PRES+1)

Offset Address: 2Bh to 28h

RTC Status Register

Bit	Field Name	Attribute	Default	Field Description
31:4	-	RO	0	Reserved
3	WUT_VAL_SYNC_READY	RO	1	Wakeup timer value synchronize ready. When this bit is 0, software cant update another wut value.
2	WUT_PRES_SYN	RO	1	Wakeup timer clock prescaler synchronize ready. When this bit is 0,

	C_READY			software cant update another wut prescaler.
1	RTC_PRES_SYNC_READY	RO	1	RTC clock prescaler synchronize ready. When this bit is 0, software cant update another prescaler.
0	INIT_SYNC_READY	RO	1	Initialization ready indicator. When this bit is 0, software cant issue another initialization and cant read the calendar.

14 Watchdog Timer (WDT)

14.1 Overview

The watchdog Timer (WDT) is a system function for monitoring the program operation. It helps to recover from error such as, runaway or deadlocked code. The WDT is configured to a predefined time-out period, and constantly runs when enabled. If the WDT is not cleared within the time-out period, it issues a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

14.1.1 Feature List

- Issues a system reset if the watchdog timer is not cleared before its time-out period
- Generates early warning interrupt
- 32-bit free-running down-counter
- Selectable time-out periods
- Software controlled lock to prevent accidental write access

14.2 **Flow Diagram**

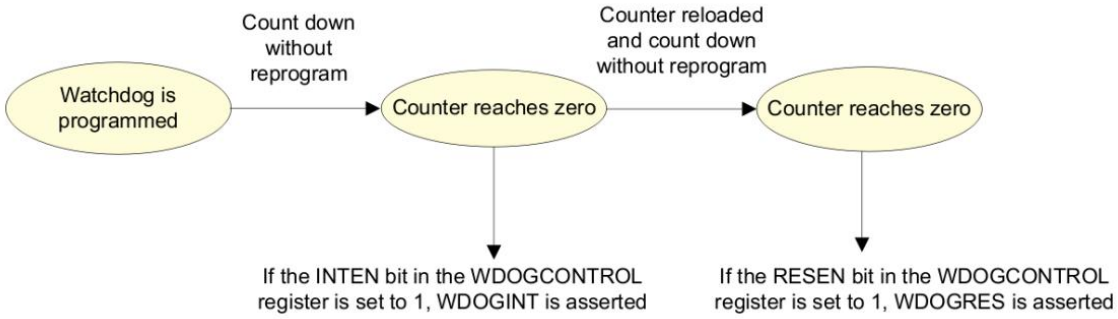
The watchdog is based on a 32-bit down-counter that is initialized from the reload register, WDOGLOAD. The watchdog module generates a regular interrupt, depending on a programmed value. The counter decrements by one on each positive clock edge of watchdog clock.

The watchdog monitors the interrupt and asserts a reset request signal if the interrupt is not cleared before the counter reaches 0 and the counter is stopped. If the interrupt is cleared before it reaches zero, then on the next enabled watchdog clock edge, the counter is reloaded from the WDOGLOAD register and the count down sequence continues.

The watchdog module applies a reset to a system in the event of a software failure, providing a way to recover from software crashes.

Below figure shows the flow diagram for the watchdog operation.

Figure 19 WDT Flow Diagram



14.3 Function Description

14.3.1 Clock Configuration

The MCU free running pclk(divider of hclk) is the input clock for watchdog module. The clock for WDT is enabled by setting WDT_CLKEN in SYSCTRL register.

14.3.2 Operation

- The WDOGLOCK register disables write access to the WDT registers. This prevents software from disabling the watchdog functionality. Writing a value of 0x1ACCE551 enables write access to register. Before writing to the WDT registers, unlock the register by writing 0x1ACCE551 to the WDOGLOCK register.
- The WDT timer is decremented for each pclk. Calculate the counter value for desired interval (us) using the following formular:

  WDOGLOAD = interval (us) * pclk (MHz)

  The interval is loaded by writing to WDOGLOAD[31:0] bits.
- The loaded counter value is reloaded on current counter (WDOGVALUE) register.
- Writing "1" to INTEN bit of the WDOGCONTROL register to enable watchdog and early warning interrupt. When the WDT counter reaches zero, triggers an interrupt. The status of interrupt is indicated by setting WDOGMIS bit. If the WDOGINTCLR register is written by any value, like "1" in the ISR routine, then WDOGMIS bit is cleared. If RESEN bit is "0", then watchdog timer is reloaded with WDOGLOAD value and continues to decrement.
- Once all the configuration is done, lock the registers by writing any value other than 0x1ACCE551 to the WDOGLOCK register.
- If WDT current counter is not reloaded with initial value then when it reaches zero, the WDT triggers an early warning interrupt. Therefore, it is required to reload the WDOGLOAD register with the same or new value before the interrupt is triggered.

14.4 Registers

Table 30 WDT Registers

Offset	Acronym	Register Description
00h	WDOGLOAD	Watchdog Load Register
04h	WDOGVALUE	Watchdog Value Register
08h	WDOGCONTROL	Watchdog Control Register
0Ch	WDOGINTCLR	Watchdog Clear Interrupt Register
10h	WDOGRIS	Watchdog Raw Interrupt Status Register
14h	WDOGMIS	Watchdog Masked Interrupt Status Register
C00h	WDOGLOCK	Watchdog Lock Register
F00h	WDOGITCR	Watchdog Integration Test Control Register
F04h	WDOGITOP	Watchdog Integration Test Output Set Register

Offset Address: 003-000h

Watchdog Load Register

Bit	Field Name	Attribute	Default	Field Description
31:0	WDOGLOAD	RW	FFFFFFFFh	Contain the value from which the counter is to decrement. When the register is written to, the count is immediately restarted from the new value.

Offset Address: 007-004h

Watchdog Value Register

Bit	Field Name	Attribute	Default	Field Description
31:0	WDOGVALUE	RO	FFFFFFFFh	Give the current value of the decrementing counter

Offset Address: 00B-008h

Watchdog Control Register

Bit	Field Name	Attribute	Default	Field Description
31:2	-	RO	0	Reserved
1	RESEN	RW	0	Enable watchdog reset output
0	INTEN	RW	0	Enable the interrupt event

Offset Address: 00F-00Ch

Watchdog Clear Interrupt Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	WDOGINTCLR	WO	0	A write of any value to the WDOGINTCLR register clears the watchdog interrupt, and reloads the counter from the value in WDOGLOAD

Offset Address: 013-010h

Watchdog Raw Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	WDOGRIS	RO	0	Indicate the raw interrupt status from the counter. This value is ANDed with the interrupt enable bit from the control register to create the masked interrupt.

Offset Address: 017-014h

Watchdog Masked Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	WDOGMIS	RO	0	Indicate the masked interrupt status from the counter.

Offset Address: C03-C00h

Watchdog Lock Register

Bit	Field Name	Attribute	Default	Field Description
31:0	WDOGLOCK	RW	0	Writing a value of 0x1ACCE551 enables write access to all other registers. Writing any other value disables write accesses. A read from this register returns only the bottom bit: 0: Indicates that write access is enabled, not locked 1: Indicates that write access is disabled, locked

Offset Address: F03-F00h

Watchdog Integration Test Control Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	WDOGITCR	RW	0	When set to 1, places the watchdog into integration test mode

Offset Address: F07-F04h

Watchdog Integration Test Output Set Register

Bit	Field Name	Attribute	Default	Field Description
31:2	-	RO	0	Reserved
1	TESTINT	RW	0	Value output on WDOGINT when in integration test mode
0	TESTRESET	RW	0	Value output on WDOGRES when in integration test mode

15 Pulse Width Modulation (PWM)

15.1 Overview

The PWM is based on the standard Timer block and inherits all of its features, although only the PWM function is pinned out. The Timer is designed to count cycles of the peripheral clock (PCLK) and optionally generate interrupts or perform other actions when specified timer values occur, based on seven match registers. The PWM function is in addition to these features, and is based on match register events.

The ability to separately control rising and falling edge locations allows the PWM to be used for more applications. For instance, multi-phase motor control typically requires three non-overlapping PWM outputs with individual control of all three pulse widths and positions.

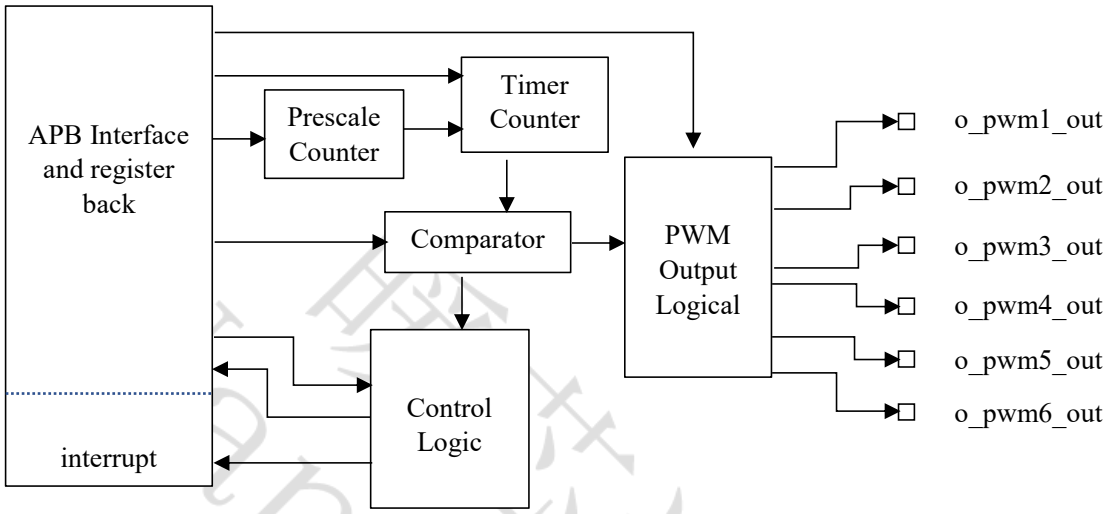
15.1.1 Feature List

- Seven match registers allow up to 6 single edge controlled or 3 double edge controlled PWM outputs, or a mix of both types.
- Supports single edge controlled and/or double edge controlled PWM outputs.
- Pulse period and width can be any number of timer counts.
- Double edge controlled PWM outputs can be programmed to be either positive going or negative going pulses.
- Match register updates are synchronized with pulse outputs to prevent generation of erroneous pulses.
- May be used as a standard timer if the PWM mode is not enabled.
- A 32 bit Timer/Counter with a programmable 32 bit Prescaler.

15.2 **Block Diagram**

A functional block diagram of the PWM is shown as below.

Figure 20 PWM Block Diagram



15.3 Function Description

Two match registers can be used to provide a single edge controlled PWM output. One match register (PWMMR0) controls the PWM cycle rate, by resetting the count upon match. The other match register controls the PWM edge position. Additional single edge controlled PWM outputs require only one match register each, since the repetition rate is the same for all PWM outputs. Multiple single edge controlled PWM outputs will all have a rising edge at the beginning of each PWM cycle, when an PWMMR0 match occurs.

Three match registers can be used to provide a PWM output with both edges controlled. Again, the PWMMR0 match register controls the PWM cycle rate. The other match registers control the two PWM edge positions. Additional double edge controlled PWM outputs require only two match registers each, since the repetition rate is the same for all PWM outputs.

With double edge controlled PWM outputs, specific match registers control the rising and falling edge of the output. This allows both positive going PWM pulses (when the rising edge occurs prior to the falling edge), and negative going PWM pulses (when the falling edge occurs prior to the rising edge).

15.3.1 Single Edge Controlled PWM Outputs

After the corresponding PWM channel enable, the single edge controlled PWM outputs go high at the beginning of a PWM cycle unless their shadow match value is equal to 0.

Each PWM output will go low when its match value is reached. If no match occurs (i.e. the match value is greater than the PWM rate), the PWM output remains continuously

high.

15.3.2 Double Edge Controlled PWM Outputs

The match values for the next PWM cycle are used at the end of a PWM cycle (a time point which is coincident with the beginning of the next PWM cycle), except as noted in rule 3.

A match value equal to 0 or the current PWM rate (the same as the Match channel 0 value) have the same effect, except as noted in rule 3. For example, a request for a falling edge at the beginning of the PWM cycle has the same effect as a request for a falling edge at the end of a PWM cycle.

When match values are changing, if one of the "old" match values is equal to the PWM rate, it is used again once if the neither of the new match values are equal to 0 or the PWM rate, and there was no old match value equal to 0.

If both a set and a clear of a PWM output are requested at the same time, clear takes precedence. This can occur when the set and clear match values are the same as in, or when the set or clear value equals 0 and the other value equals the PWM rate.

If a match value is out of range (i.e. greater than the PWM rate value), no match event occurs and that match channel has no effect on the output. This means that the PWM output will remain always in one state, allowing always low, always high, or "no change" outputs.

15.4 **Registers**

The peripheral registers have to be accessed by byte (8 bits), half-words (16 bits) or words (32 bits). These registers are available at 32-bit addresses in the device memory map. See the device-specific data manual for the memory address of these registers.

Table 31 PWM Registers

Offset	Acronym	Register Description
00h	PWM_TC	PWM Timer Counter Register
04h	PWM_TCR	PWM Timer Control Register
08h	PWM_IR	PWM Interrupt Register
0Ch	PWM_PR	PWM Prescale Register
10h	PWM_PC	PWM Prescale Counter Register
14h	PWM_MCR	PWM Match Control Register
18h	PWM_MR0	PWM Match Register0
1Ch	PWM_MR1	PWM Match Register1
20h	PWM_MR2	PWM Match Register2
24h	PWM_MR3	PWM Match Register3
28h	PWM_MR4	PWM Match Register4
2Ch	PWM_MR5	PWM Match Register5
30h	PWM_MR6	PWM Match Register6
34h	PWM_PCR	PWM Control Register
38h	PWM_LER	PWM Load Enable Register
3Ch	PWM_DBG	PWM Debug Register

Offset Address: 03-00h

PWM Timer Register

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_TC	RW	0	Timer counter

Offset Address: 07-04h

PWM Timer Register

Bit	Field Name	Attribute	Default	Field Description
31:2	-	-	0	Reserved
1	CNT_EN	RW	0	Counter enable 0: The counters are disabled 1: The PWM Timer Counter and PWM Prescale Counter are enabled for counting
0	CNT_RET	RW	0	Counter reset enable

				0: Clear reset 1: The PWM Timer Counter and the PWM Prescale Counter are synchronously reset

Offset Address: 0B-08h

PWM Interrupt Register

Bit	Field Name	Attribute	Default	Field Description
31:7	-	-	0	Reserved
6	PWMMR6_INT	RW1C	0	Interrupt flag for PWM match 6
5	PWMMR5_INT	RW1C	0	Interrupt flag for PWM match 5
4	PWMMR4_INT	RW1C	0	Interrupt flag for PWM match 4
3	PWMMR3_INT	RW1C	0	Interrupt flag for PWM match 3
2	PWMMR2_INT	RW1C	0	Interrupt flag for PWM match 2
1	PWMMR1_INT	RW1C	0	Interrupt flag for PWM match 1
0	PWMMR0_INT	RW1C	0	Interrupt flag for PWM match 0

Offset Address: 0F-0Ch

PWM Prescaler Register

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_PR	RW	0	Prescale register

Offset Address: 13-10h

PWM Prescaler Counter Register

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_PC	RW	0	Prescale counter

Offset Address: 17-14h

PWM Match Control Register

Bit	Field Name	Attribute	Default	Field Description
31:21	-	-	0	Reserved
20	PWMMR6_STP	RW	0	0:This feature is disabled 1:Stop on PWMMR6: the PWMTC and PWMPC will be stopped and PWMTCR[2] will be set to 0 if PWMMR6 matches the PWMTC
19	PWMMR6_RET	RW	0	0:This feature is disabled 1:Reset on PWMMR6: the PWMTC will be reset if PWMMR6 matches it
18	PWMMR6_INT	RW	0	0: This interrupt is disabled 1: Interrupt on PWMMR6: an interrupt is generated when PWMMR6 matches the value in the PWMTC
17	PWMMR5_STP	RW	0	0:This feature is disabled 1:Stop on PWMMR5: the PWMTC and PWMPC will be stopped and PWMTCR[2] will be set to 0 if PWMMR5 matches the PWMTC
16	PWMMR5_RET	RW	0	0:This feature is disabled 1:Reset on PWMMR5: the PWMTC will be reset if PWMMR5 matches it
15	PWMMR5_INT	RW	0	0: This interrupt is disabled 1: Interrupt on PWMMR5: an interrupt is generated when PWMMR5 matches the value in the PWMTC
14	PWMMR4_STP	RW	0	0:This feature is disabled 1:Stop on PWMMR4: the PWMTC and PWMPC will be stopped and PWMTCR[2] will be set to 0 if PWMMR4 matches the PWMTC
13	PWMMR4_RET	RW	0	0:This feature is disabled 1:Reset on PWMMR4: the PWMTC will be reset if PWMMR4 matches it
12	PWMMR4_INT	RW	0	0: This interrupt is disabled 1: Interrupt on PWMMR4: an interrupt is generated when PWMMR4 matches the value in the PWMTC
11	PWMMR3_STP	RW	0	0:This feature is disabled 1:Stop on PWMMR3: the PWMTC and PWMPC will be stopped and PWMTCR[2] will be set to 0 if PWMMR3 matches the PWMTC
10	PWMMR3_RET	RW	0	0:This feature is disabled 1:Reset on PWMMR3: the PWMTC will be reset if PWMMR3 matches it
9	PWMMR3_INT	RW	0	0: This interrupt is disabled 1: Interrupt on PWMMR3: an interrupt is generated when PWMMR3 matches the value in the PWMTC
8	PWMMR2_STP	RW	0	0:This feature is disabled 1:Stop on PWMMR2: the PWMTC and PWMPC will be stopped and PWMTCR[2] will be set to 0 if PWMMR2 matches the PWMTC
7	PWMMR2_RET	RW	0	0:This feature is disabled 1:Reset on PWMMR2: the PWMTC will be reset if PWMMR2 matches it
6	PWMMR2_INT	RW	0	0: This interrupt is disabled 1: Interrupt on PWMMR2: an interrupt is generated when PWMMR2 matches the value in the PWMTC
5	PWMMR1_STP	RW	0	0:This feature is disabled 1:Stop on PWMMR1: the PWMTC and PWMPC will be stopped and

				PWMTCR[2] will be set to 0 if PWMMR1 matches the PWMTC
4	PWMMR1_RET	RW	0	0:This feature is disabled 1:Reset on PWMMR1: the PWMTC will be reset if PWMMR1 matches it
3	PWMMR1_INT	RW	0	0: This interrupt is disabled 1: Interrupt on PWMMR1: an interrupt is generated when PWMMR1 matches the value in the PWMTC
2	PWMMR0_STP	RW	0	0:This feature is disabled 1:Stop on PWMMR0: the PWMTC and PWMPC will be stopped and PWMTCR[2] will be set to 0 if PWMMR0 matches the PWMTC
1	PWMMR0_RET	RW	0	0:This feature is disabled 1:Reset on PWMMR0: the PWMTC will be reset if PWMMR0 matches it
0	PWMMR0_INT	RW	0	0: This interrupt is disabled 1: Interrupt on PWMMR0: an interrupt is generated when PWMMR0 matches the value in the PWMTC

Offset Address: 1B-18h

PWM Match Register0

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_MR0	RW	0	Match register0

Offset Address: 1F-1Ch

PWM Match Register1

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_MR1	RW	0	Match register1

Offset Address: 23-20h

PWM Match Register2

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_MR2	RW	0	Match register2

Offset Address: 27-24h

PWM Match Register3

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_MR3	RW	0	Match register3

Offset Address: 2B-28h

PWM Match Register4

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_MR4	RW	0	Match register4

Offset Address: 2F-2Ch

PWM Match Register5

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_MR5	RW	0	Match register5

Offset Address: 33-30h

PWM Match Register6

Bit	Field Name	Attribute	Default	Field Description
31:0	PWM_MR6	RW	0	Match register6

Offset Address: 37-34h

PWM Control Register

Bit	Field Name	Attribute	Default	Field Description
31:11	-	-	0	Reserved
10	PWMEN6	RW	0	0: The PWM6 output disabled 1: The PWM6 output enabled
9	PWMEN5	RW	0	0: The PWM5 output disabled 1: The PWM5 output enabled
8	PWMEN4	RW	0	0: The PWM4 output disabled 1: The PWM4 output enabled
7	PWMEN3	RW	0	0: The PWM3 output disabled 1: The PWM3 output enabled
6	PWMEN2	RW	0	0: The PWM2 output disabled 1: The PWM2 output enabled
5	PWMEN1	RW	0	0: The PWM1 output disabled

				1: The PWM1 output enabled
4	PWMSEL6	RW	0	0: Selects single edge controlled mode for PWM6 1: Selects double edge controlled mode for the PWM6 output
3	PWMSEL5	RW	0	0: Selects single edge controlled mode for PWM5 1: Selects double edge controlled mode for the PWM5 output
2	PWMSEL4	RW	0	0: Selects single edge controlled mode for PWM4 1: Selects double edge controlled mode for the PWM4 output
1	PWMSEL3	RW	0	0: Selects single edge controlled mode for PWM3 1: Selects double edge controlled mode for the PWM3 output
0	PWMSEL2	RW	0	0: Selects single edge controlled mode for PWM2 1: Selects double edge controlled mode for the PWM2 output

Offset Address: 3B-38h

PWM Load Enable Register

Bit	Field Name	Attribute	Default	Field Description
31:7	-	-	0	Reserved
6	PWMML6_EN	RW	0	0: Load disabled 1: The last value written to the PWM Match 6 register to be become effective when the timer is next reset by a PWM Match event.
5	PWMML5_EN	RW	0	0: Load disabled 1: The last value written to the PWM Match 5 register to be become effective when the timer is next reset by a PWM Match event.
4	PWMML4_EN	RW	0	0: Load disabled 1: The last value written to the PWM Match 4 register to be become effective when the timer is next reset by a PWM Match event.
3	PWMML3_EN	RW	0	0: Load disabled 1: The last value written to the PWM Match 3 register to be become effective when the timer is next reset by a PWM Match event.
2	PWMML2_EN	RW	0	0: Load disabled 1: The last value written to the PWM Match 2 register to be become effective when the timer is next reset by a PWM Match event.

1	PWMML1_EN	RW	0	0: Load disabled 1: The last value written to the PWM Match 1 register to be become effective when the timer is next reset by a PWM Match event.
0	PWMML0_EN	RW	0	0: Load disabled 1: The last value written to the PWM Match 0 register to be become effective when the timer is next reset by a PWM Match event.

Offset Address: 3F-3Ch

PWM Debug Data Register

Bit	Field Name	Attribute	Default	Field Description
31:12	-	-	0	Reserved
11:0	PWM_DBG	RO	0	Debug data

16 Timer

16.1 Overview

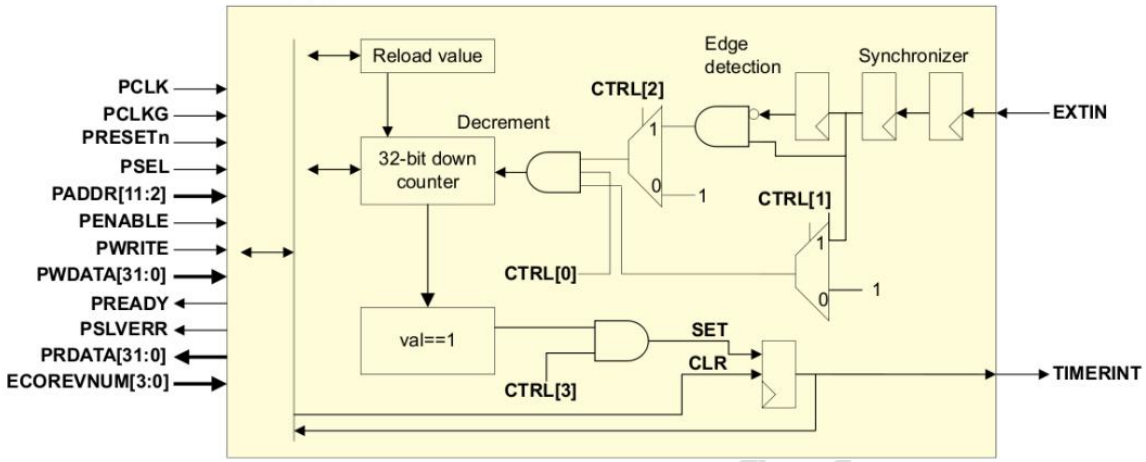
The APB timer is a 32-bit down-counter.

16.1.1 Feature List

- Generate an interrupt when counter reaches 0
- Reload counter when counter reaches 0
- Support external pin as timer enable
- Support external pin as clock

16.2 Block Diagram

Figure 21 Timer Block Diagram



16.3 Function Description

When counter reaches 0, generate an interrupt when interrupt enable is set and counter is updated with reload register TMRRLD. The interrupt status is held until it is cleared by writing to the TMRINT register. When writing to the TMRVAL register, update the current value.

If the APB timer count reaches 0 and, at the same time, the software clears a previous interrupt status, the interrupt status is set to 1.

You can use the zero to one transition of the external input signal, EXTIN, as a timer enable when EXT_EN is set.

When EXT_CLK is set, EXTIN as external clock input, must be slower than half of the peripheral clock because it is sampled by a double flip-flop and then goes through edge-detection logic.

16.4 **Registers**

Table 32 Timer Registers

Offset	Acronym	Register Description
00h	TMRCTRL	Timer Control Register
04h	TMRVAL	Timer Value Register
08h	TMRRLD	Timer Reload Register
0Ch	TMRINT	Timer Interrupt Status Register

Offset Address: 03-00h

Timer Control Register

Bit	Field Name	Attribute	Default	Field Description
31:4	-	RO	0	Reserved
3	INT_EN	RW	0	Timer Interrupt Enable
2	EXT_CLK	RW	0	Select External input as Clock
1	EXT_EN	RW	0	Select External input as Enable
0	TMR_EN	RW	0	Timer Enable

Offset Address: 07-04h

Timer Value Register

Bit	Field Name	Attribute	Default	Field Description
31:0	VALUE	RW	0	When write, update current counter value by this register value When read, given the current counter value

Offset Address: 0B-08h

Timer Reload Register

Bit	Field Name	Attribute	Default	Field Description
31:0	RELOAD	RW	0	Reload value. A write to this register sets the current value after it reaches 0

Offset Address: 0F-0Ch

Timer Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	INT_STS	RW	0	Timer interrupt status. Write one to clear

17 Dual-Timers

17.1 Overview

The dual timer is an APB dual-input timer module consisting of two programmable 32-bit or 16-bit down-counters (TIMER1, TIMER2) that can generate interrupts when they reach zero. The operation of each timer module is identical.

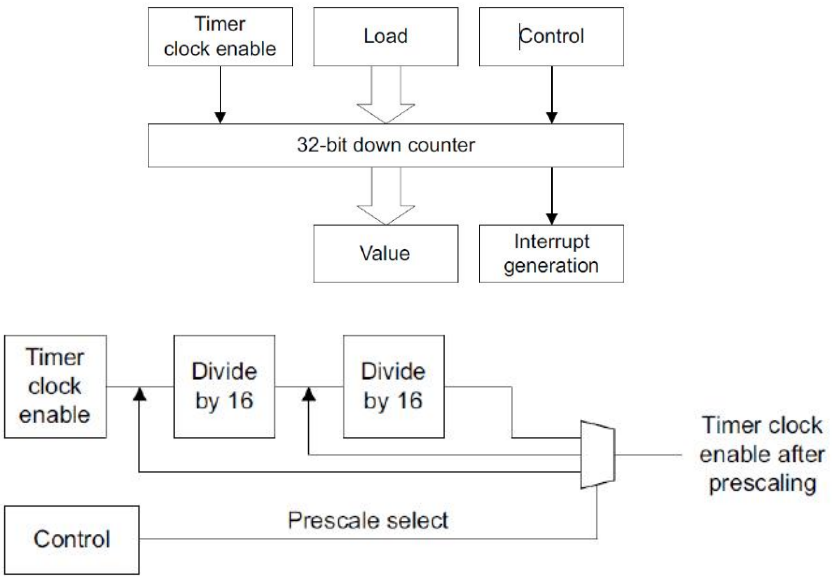
17.1.1 Feature List

- Selectable configuration
  - 16 or 32-bit down counter operation
- Three different modes of operation
  - Free-Running mode
  - One Shot Count mode
  - Periodic mode
- Interrupts on counter underflow
- Internal three selectable prescaler

17.2 Block Diagram

The following are the block diagrams of the dual timer and pre-scale clock enable generation.

Figure 22 Dual-Timers Block Diagram



17.3 Function Description

17.3.1 Free-running Mode

The counter wraps after reaching zero and generates an interrupt every time when it reaches zero. It also continues to count down from the maximum value. This is the default mode.

17.3.2 Periodic Timer Mode

The counter generates an interrupt at a constant interval, reloading the original value after wrapping past zero.

17.3.3 One-shot Timer Mode

The counter generates an interrupt once. When the counter reaches 0, it halts until you reprogram it. You can do this using one of the following:

Clearing the one-shot count bit in the control register, in which case the count proceeds according to the selection of Free-running or Periodic mode.

Writing a new value to the Load Value register.

17.3.4 Operation

Each timer is identical in its operation and has an identical set of register.

- Select the 16-bit or 32-bit timer operation by configuring TIMER_SIZE bit.
- Select the timer mode of operations by configuring ONE_SHOT, TIMER_MODE bits.
- Enable timer interrupt by writing “1” to INT_EN bit.
- Counter is decremented for each PCLK. Calculate the counter value for desired interval (ms) using the following formula:

  TIMn_LOAD = Interval (ms) * PCLK (KHz)
- The interval is loaded by writing to the TIMn_LOAD bits.
- The loaded counter value is reloaded on current counter of TIMn_VAL register.
- Enable the timer by writing “1” to TIMER_EN bit.
- The loaded counter value counts down to 0.
- When 0 is reached, an interrupt is generated. This is indicated through masked interrupt status bit in TIMn_MIS register. Interrupt can be cleared by writing to the TIMn_INTCLR register. If interrupt is not enabled, when the counter reaches 0, the raw interrupt status register TIMn_RIS indicates the counter expiry.

- In the One-Shot mode, the timer halts when it reaches 0. In periodic mode, the counter continues to decrement to 0, and then reloads the TIMn_VAL from the TIMn_LOAD register and continues to decrement. In this mode, the counter effectively generates a periodic interrupt. If the timer is operating in Free-Running mode, it continues to decrement from its maximum value (0xFFFFFFFF for 32-bit mode, 0xFFFF for 16-bit mode).

17.4 Registers

Table 33 Dual-Timers Registers

Offset	Acronym	Register Description
00h	TIM1_LOAD	Timer1 Load Register
04h	TIM1_VAL	Timer1 Current Value Register
08h	TIM1_CTRL	Timer1 Control Register
0Ch	TIM1_INTCLR	Time1 Interrupt Clear Register
10h	TIM1_RIS	Timer1 Raw Interrupt Status Register
14h	TIM1_MIS	Timer1 Masked Interrupt Status Register
18h	TIM1_BGLOAD	Timer1 Background Load Register
20h	TIM2_LOAD	Timer2 Load Register
24h	TIM2_VAL	Timer2 Current Value Register
28h	TIM2_CTRL	Timer2 Control Register
2Ch	TIM2_INTCLR	Timer2 Interrupt Clear Register
30h	TIM2_RIS	Timer2 Raw Interrupt Status Register
34h	TIM2_MIS	Timer2 Masked Interrupt Status Register
38h	TIM2_BGLOAD	Timer2 Background Load Register

Offset Address: 03-00h

Timer1 Load Register

Bit	Field Name	Attribute	Default	Field Description
31:0	TIM1_LOAD	RW	0	This register contains the value from which the counter is to decrement. This is the value that is used to reload the counter when periodic mode is enabled, and the current count reaches 0.

Offset Address: 07-04h

Timer1 Current Value Register

Bit	Field Name	Attribute	Default	Field Description
31:0	TIM1_VAL	RO	32'hFFFF_FFFF	This register provides the current value of the decrementing counter.

Offset Address: 0B-08h

Timer1 Control Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RW	0	Reserved
7	TIMER_EN	RW	0	Enable bit: 0: Timer disabled

				1: Timer enabled
6	TIMER_MODE	RW	0	Mode bit: 0: Timer is in free-running mode 1: Timer is in periodic mode
5	INT_EN	RW	1	Interrupt enable bit: 0: Timer Interrupt disabled 1: Timer Interrupt enabled
4	-	RO	0	Reserved
3:2	TIMER_PRE	RW	00b	Prescale bits: 00: 0 stages of prescale, clock is divided by 1 01: 4 stages of prescale, clock is divided by 16 10, 11: 8 stages of prescale, clock is divided by 256
1	TIMER_SIZE	RW	0	Select 16-bit or 32-bit counter operation: 0: 16-bit counter 1: 32-bit counter
0	ONE_SHOT	RW	0	Select one-shot or wrapping counter mode: 0: Wrapping mode 1: One-shot mode

Offset Address: 0F-0Ch

Timer1 Interrupt Clear Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	TIM1_INTCLR	WO	0	Any write to this register clears the interrupt output from the counter

Offset Address: 13-10h

Timer1 Raw Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	TIM1_RIS	RO	0	This register indicates the raw status from the counter. This value is ANDed with the timer interrupt enable bit from the Timer Control

				register to create the masked interrupt, that is passed to the interrupt output pin.

Offset Address: 17-14h

Timer1 Masked Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	TIM1_MIS	RO	0	This register indicates the masked status from the counter. This value is the logic AND of the raw interrupt status with the timer interrupt enable bit from the Timer Control register, and is the same value that is passed to the interrupt output pin

Offset Address: 1B-18h

Timer1 Background Load Register

Bit	Field Name	Attribute	Default	Field Description
31:0	TIM1_BGLOAD	RW	0	This register contains the value from which the counter is to decrement. This is the value that is used to reload the counter when periodic mode is enabled, and the current count reaches 0. This register provides an alternative method of accessing the TIM1_LOAD register. The difference is that writes to TIM1_BGLOAD do not cause the counter to immediately restart from the new value. Reading from this register returns the same value that is returned from TIM1_LOAD.

Offset Address: 1F-1Ch

Reserved

Bit	Field Name	Attribute	Default	Field Description
31:0	-	RO	0	Reserved

Offset Address: 23-20h

Timer2 Load Register

Bit	Field Name	Attribute	Default	Field Description
31:0	TIM1_LOAD	RW	0	This register contains the value from which the counter is to decrement. This is the value that is used to reload the counter when periodic mode is enabled, and the current count reaches 0.

Offset Address: 27-24h

Timer2 Current Value Register

Bit	Field Name	Attribute	Default	Field Description
31:0	TIM2_VAL	RO	32'hFFFF_FFFF	This register provides the current value of the decrementing counter.

Offset Address: 2B-28h

Timer2 Control Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RW	0	Reserved
7	TIMER_EN	RW	0	Enable bit: 0: Timer disabled 1: Timer enabled
6	TIMER_MODE	RW	0	Mode bit: 0: Timer is in free-running mode 1: Timer is in periodic mode
5	INT_EN	RW	1	Interrupt enable bit: 0: Timer Interrupt disabled 1: Timer Interrupt enabled
4	-	RO	0	Reserved
3:2	TIMER_PRE	RW	00b	Prescale bits: 00: 0 stages of prescale, clock is divided by 1 01: 4 stages of prescale, clock is divided by 16 10, 11: 8 stages of prescale, clock is divided by 256

1	TIMER_SIZE	RW	0	Select 16-bit or 32-bit counter operation: 0: 16-bit counter 1: 32-bit counter
0	ONE_SHOT	RW	0	Select one-shot or wrapping counter mode: 0: Wrapping mode 1: One-shot mode

Offset Address: 2F-2Ch

Timer2 Interrupt Clear Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	TIM2_INTCLR	WO	0	Any write to this register clears the interrupt output from the counter

Offset Address: 33-30h

Timer2 Raw Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	TIM2_RIS	RO	0	This register indicates the raw status from the counter. This value is ANDed with the timer interrupt enable bit from the Timer Control register to create the masked interrupt, that is passed to the interrupt output pin.

Offset Address: 37-34h

Timer2 Masked Interrupt Status Register

Bit	Field Name	Attribute	Default	Field Description
31:1	-	RO	0	Reserved
0	TIM2_MIS	RO	0	This register indicates the masked status from the counter. This value is the logic AND of the raw interrupt status with the timer interrupt enable bit from the Timer Control register, and is the same value that is passed to the interrupt output pin

Offset Address: 3B-38h

Timer2 Background Load Register

Bit	Field Name	Attribute	Default	Field Description
31:0	TIM2_BGLOAD	RW	0	This register contains the value from which the counter is to decrement. This is the value that is used to reload the counter when periodic mode is enabled, and the current count reaches 0. This register provides an alternative method of accessing the TIM2_LOAD register. The difference is that writes to TIM2_BGLOAD do not cause the counter to immediately restart from the new value. Reading from this register returns the same value that is returned from TIM2_LOAD.

18 Analog Control (ANAC)

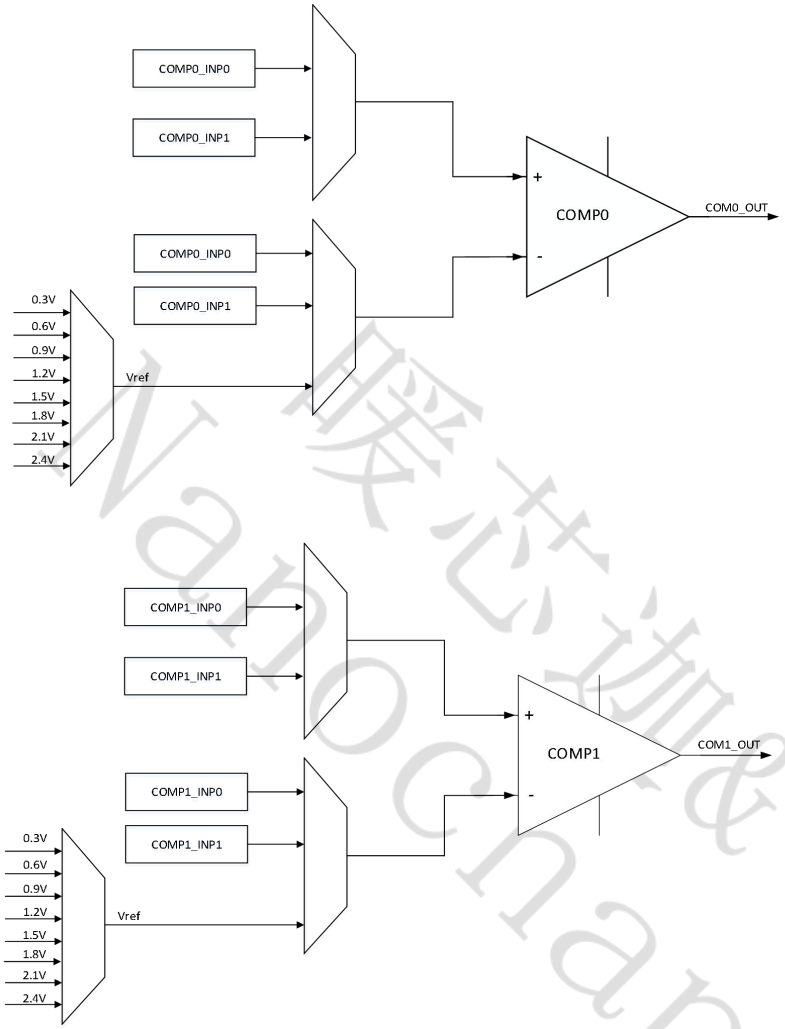
18.1 Comparator

18.1.1 Feature List

- Two rail-to-rail comparators
- Flexible input select
  - Input from IO
  - Internal Vref
- Asynchronous interrupt event which can wakeup MCU
- Comparator output to GPIO

18.1.2 Block Diagram

Figure 23 Comparator Block Diagram



18.1.3 Function Description

Comparator compare the Vin+ and Vin-, when Vin+ > Vin-, output “1”, otherwise output “0”.

Vin+ can be from multi IO input, Vin- can be from multi IO input or internal reference voltage.

Comparator output generate asynchronous interrupt without clock, that can wakeup MCU from low-power mode.

Table 34 COMP0 Pins and Registers

COMP0	Pins	Pin select register	Function select register
COMP0_INP0	GPIO3	GPIO3_ALTF=2’b11 & GPIO_ANAE[3]=1’b1	
COMP0_INP1	GPIO22	GPIO22_ALTF=2’b11 & GPIO_ANAE[22]=1’b1	
COMP0_INN0	GPIO5	GPIO5_ALTF=2’b11 & GPIO_ANAE[5]=1’b1	
COMP0_INN1	GPIO23	GPIO23_ALTF=2’b11 & GPIO_ANAE[23]=1’b1	
COMP0_OUT	GPIO20	GPIO20_ALTF=2’b10	

Table 35 COMP1 Pins and Registers

COMP1	Pins	Pin select register	Function select register
COMP1_INP0	GPIO12	GPIO12_ALTF=2’b11 & GPIO_ANAE[12]=1’b1	
COMP1_INP1	GPIO14	GPIO14_ALTF=2’b11 & GPIO_ANAE[14]=1’b1	
COMP1_INN0	GPIO13	GPIO13_ALTF=2’b11 & GPIO_ANAE[13]=1’b1	
COMP1_INN1	GPIO19	GPIO19_ALTF=2’b11 & GPIO_ANAE[19]=1’b1	
COMP1_OUT	GPIO21	GPIO21_ALTF=2’b10	

18.2 **Registers**

Table 36 Analog Control Registers

Offset	Acronym	Register Description
00h	COMP0_CTRL	Comparator 0 Control Register
04h	COMP1_CTRL	Comparator 1 Control Register
08h	PGA_CTRL	PGA Control Register
0Ch	CHARGE_CTRL	Charge Control Register
10h	PMU_CTRL	PMU Control Register
14h	BOOST_CTRL	Boost Control Register
18h	ANA_BIST	Analog BIST Select Register

Offset Address: 03-00h

Comparator 0 Control Register

Bit	Field Name	Attribute	Default	Field Description
31:9	-	RO	0	Reserved
8	COMP0_OUT	RO	0	Comparator 0 output
7	COMP_REF_EN	RW	0	Comparator 0/1 reference voltage enable bit 0: disabled 1: enabled
6:4	COMP0_VREF_SEL	RW	0	Comparator 0 reference voltage select 000: 0.3V 001: 0.6V 010: 0.9V 011: 1.2V 100: 1.5V 101: 1.8V 110: 2.1V 111: 2.4V
3:2	COMP0_SIGSEL_N	RW	0	Comparator 0 negative input select 00: COMP0_VIN0 01: COMP0_VIN1 10: Vref

				11: Reserved
1	COMP0_SIGSEL_P	RW	0	Comparator 0 positive input select 0: COMP0_VIP0 1: COMP0_VIP1
0	COMP0_EN	RW	0	Comparator 0 enable bit 0: disabled 1: enabled

Offset Address: 07-04h

Comparator 1 Control Register

Bit	Field Name	Attribute	Default	Field Description
31:9	-	RW	0	Reserved
8	COMP1_OUT	RO	0	Comparator 1 output
7	-	RO	0	
6:4	COMP1_VREF_SEL	RW	0	Comparator 1 reference voltage select 000: 0.3V 001: 0.6V 010: 0.9V 011: 1.2V 100: 1.5V 101: 1.8V 110: 2.1V 111: 2.4V
3:2	COMP1_SIGSEL_N	RW	0	Comparator 1 negative input select 00: COMP1_VIN0 01: COMP1_VIN1 10: Vref 11: Reserved
1	COMP1_SIGSEL_P	RW	0	Comparator 1 positive input select

				0: COMP1_VIP0 1: COMP1_VIP1
0	COMP1_EN	RW	0	Comparator 1 enable bit 0: disabled 1: enabled

Offset Address: 0B-08h

PGA Control Register

Bit	Field Name	Attribute	Default	Field Description
31:14	-	RO	0	Reserved
13:12	PGA_VIN_SEL	RW	0	PGA negative input selection 00:PGA_VIN0 01:PGA_VIN1 10: internal VCM 11: external VCM
11	-	RO	0	Reserved
10:8	PGA_VIP_SEL	RW	0	PGA positive input selection 000:PGA_VIP0 001:PGA_VIP1 010: internal VCM 011: VBAT 100: AVDD1P8 101: VLCD[0] 110: VREF1P2V 111: V_temp
7	-	RO	0	Reserved
6:4	PGA_GAIN_SEL	RW	0	PGA gain selection bits For inverting mode, the gain is from 1 to 8 For non-inverting mode, the gain is from 2 to 9

3:2	-	RO	0	
1	PGA_REF_EN	RW	0	PGA reference enable bit 0: disabled 1: enabled
0	PGA_OP_EN	RW	0	PGA enable bit 0: disabled 1: enabled

Offset Address: 0F-0Ch

Charge Control Register

Bit	Field Name	Attribute	Default	Field Description
31:11	-	RO	0	Reserved
10:8	CHARGE_CURRENTSEL	RW	010b	Signals for setting the charge current value 000: 133mA 001: 145mA 010: 158mA 011: 174mA 100: 193mA 101: 219mA 110: 250mA 111: 292mA
7	-	RO	0	Reserved
6:4	CHARGE_LVSEL	RW	100b	Signals for setting low voltage detection value: 000: 300mV 001: 600mV 010: 900mV 011: 1200mV 100: 1500mV 101: 1800mV

				110: 2100mV 111: 2400mV
3	CHARGER_END	RO	0	When charge is finished, this signal turns from “0” to “1”
2	CHARGER_OK	RO	0	When the power supply for charge is ready, the signal turns from “0” to “1”
1	CHARGE_BATON	RW	0	Enable bit for turning on the charge function 0: disabled 1: enabled
0	CHARGE_SYSON	RW	0	Enable bit for turning on the LDO in charger 0: disabled 1: enabled

Offset Address: 13-10h

PMU Control Register

Bit	Field Name	Attribute	Default	Field Description
31:8	-	RO	0	Reserved
7	TEMP_150C_TRIG	RO	0	When the chip temperature is 150C, the signal turns from 0 to 1
6	LVD_OUT	RO	0	When he supply is lower than than the threshold value, the signal turns from 0 to 1
5	TEMP_EN	RW	0	Enable bit for turning on the temperature sensor 0: disabled 1: enabled
4	BG_BUFFER_EN	RW	0	Enable bit for turning on the bandgap buffer 0: disabled 1: enabled
3:1	LVD_INSEL	RW	0	Signals for selecting different voltage for LVD 000: 4.2V 001: 3.9V 010: 3.6V 011: 3.3V

				100: 3.0V 101: 2.7V 110: 2.4V 111: 2.1V
0	LVD_EN	RW	0	Enable bit for turning on the low voltage detector 0: disabled 1: enabled

Offset Address: 17-14h

Reserved

Offset Address: 1B-18h

Analog BIST Select Register

Bit	Field Name	Attribute	Default	Field Description
31:16	-	RO	0	Reserved
15:8	ANA_RSV	RO	0	Reserved signals for analog
7	-	RO	0	-
6:4	BIST_SEL	RW	0	Selecting different signals to the analog bist output 000: bandgap current 001: VBG 010: 1.2V for comparator 011: vcm (vdd/2) for pga 100: vcm (vdd/2) for adc 101: temperature voltage of DC-DC boost 110: reserved 111: reserved
3:1	-	RO	0	
0	BIST_EN	RW	0	Enable bit for analog bist 0: disabled 1: enabled

19 Liquid Crystal Display Controller (LCD)

19.1 Overview

The LCD controller is a digital controller/driver for monochrome passive liquid crystal display (LCD) with up to 4 common terminals and up to 16 segment terminals.

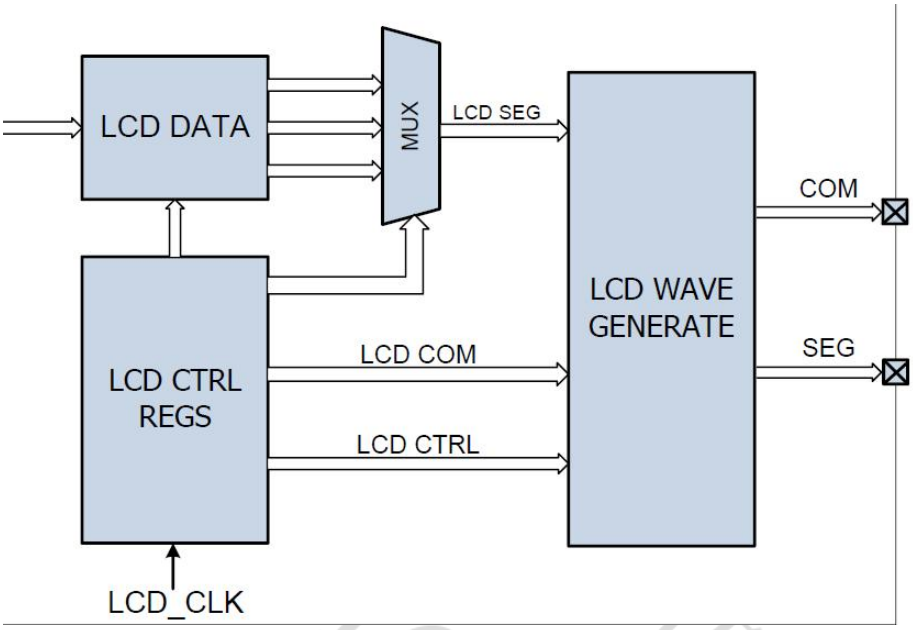
LCD needs total 20 I/O pads including 4 COM and 16 SEG and mux wih GPIO. Before enable LCD, set the related GPIO_ALTF[1:0] to 2'b11, then GPIO is for analog function. And then enable LCD and related COMEN and SEGEN.

19.1.1 Feature List

- Support up to 4 common terminals and 16 segment terminals
- Support 1/3 bias
- Support 1/4 duty
- Support 16 gray degree
- Support flicker and the flicker frequency is configurable
- Support intermittent lighten up
- Support all screen lighten up and all screen extinguish
- LCD driver can work at active mode ,sleep mode and stop mode
- Support TYPE A driver wave
- Typical refresh frequency is 64Hz

19.2 **Block Diagram**

Figure 24 LCD Block Diagram



20 APPENDIX

20.1 Stimulation parameters

20.1.1 Stimulation channel

Item	Value	Unit
Supply Voltage	5~60	V
Charge Imbalance	<10	uC/sec
	<0.75uA	mm^2
Current Array	2 drivers with current and switches (2 current sources,2 current sinks)	
Current Range	0~67, 8bits	mA
Stimulation Frequency	1~250K	Hz
Pulse Width	2~infinity	us
Unit Current	33~264, 33steps	uA
Resistance Measurement	0.1~10	k

20.1.2 Waveform example

Figure 25 Square waveform

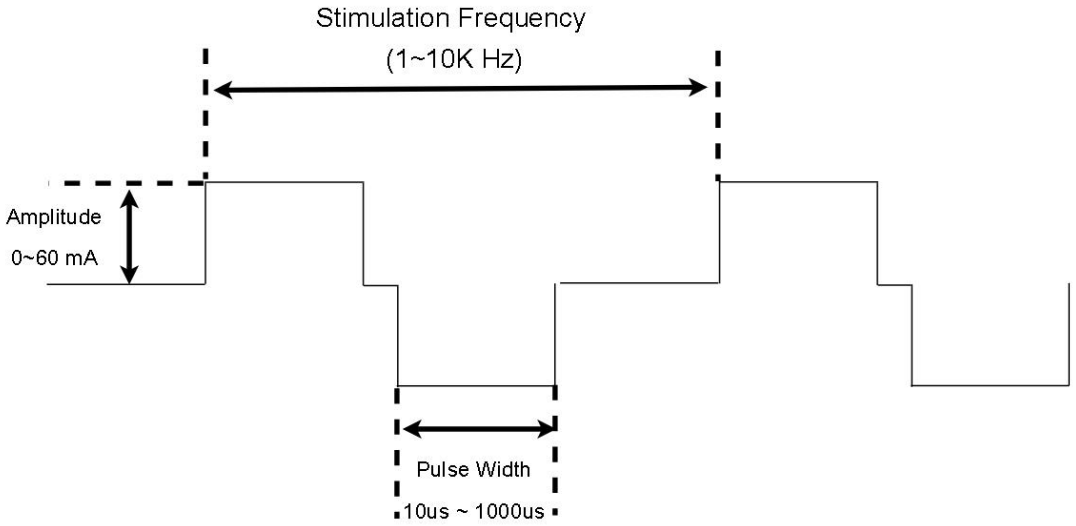


Figure 26 Sine waveform

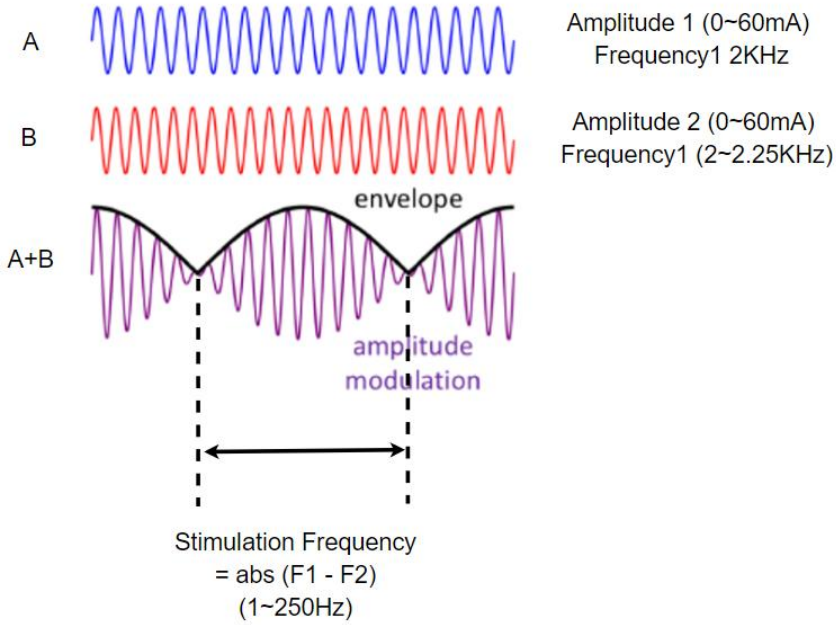
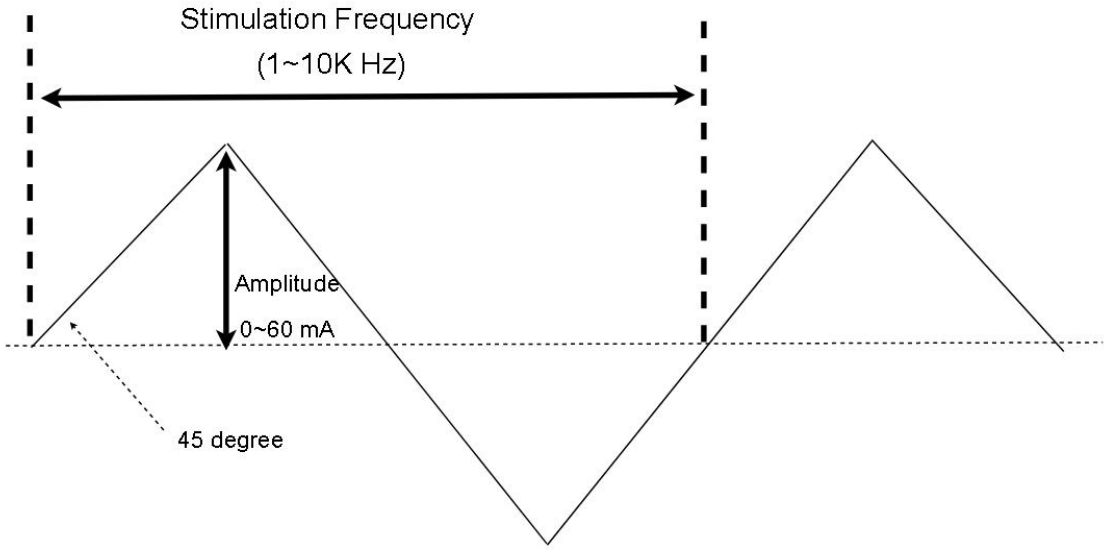


Figure 27 Triangle waveform



20.1.3 Waveform registers

Name	Bits range	Access	Definition
ADDR_WG_DRV_CONFIG_REG	<7:0>	RD/WR	bit 0: rest enable, 1: negative enable, 2: silent enable, 3: source B enable, 4: alternating (+/-) the positive side, 5: continue repeating the waveform if one works together with interrupt if zero, 6: multi-electrode
ADDR_WG_DRV_CTRL_REG	<7:0>	RD/WR	Bit 0 only: enabling the wave gen block; rest not used
ADDR_WG_DRV_REST_T_REG	<7:0>	WR	resting time (in microseconds) between the positive side and the negative side of the wave in a period
ADDR_WG_DRV_SILENT_T_REG	<31:0>	WR	silent time (in microseconds) before the next wave period
ADDR_WG_DRV_HLF_WAVE_PRD_REG	<31:0>	WR	Half of the period of the arbitrary (e.g., sine or square) wave (in microseconds)
ADDR_WG_DRV_NEG_HLF_WAVE_PRD_REG	<31:0>	WR	Negative half of the period of the arbitrary (e.g., sine or square) wave (in microseconds)
ADDR_WG_DRV_CLK_FREQ_REG	<7:0>	WR	Frequency of the clock in MHz
ADDR_WG_DRV_IN_WAVE_ADDR_REG	<7:0>	WR	Address for writing the next 8 bit of the wave form value to the register file of APB where the wave values are stored
ADDR_WG_DRV_IN_WAVE_REG	<7:0> <15:8>	WR	Next half wave point value to be written to the address specified by ADDR_WG_DRV_IN_WAVE_ADDR_REG. Total half wave is 64 points, each point 8 bit between 0 and 255. If Driver B is used, user can

			choose which channel of Driver B is activated, while sending out the half wave point <7:0>. That is, for each value of half wave, a unique channel can be chosen (between 0 and 23). This value (channel number) is saved in bits <15:8> (only applicable for Driver B).
ADDR_WG_DRV_ALT_LIM_REG	<15:0>	WR	Number of clocks for a period of alternating signal
ADDR_WG_DRV_ALT_SILENT_LIM_REG	<15:0>	WR	Number of clocks for each silent duration for the alternating frequency
ADDR_WG_DRV_DELAY_LIM_REG	<15:0>	WR	Number of clocks for initial delay after the reset is disabled and before the waves are generated. This delay is for situations where there are multiple drivers like A and C so that we can put delay on each of them before they start to prevent them from overlapping. In driver b, however, there is only one driver (with 24 channels). Delaying will only delay the start which we is not useful as we would delay the start of sound wave generation for hearing aid while it should start as quick as possible
ADDR_WG_DRV_NEG_SCALE_REG	<7:0>	WR	Scale the negative side of the waveform by this unsigned value (multiply by this value)
ADDR_WG_DRV_NEG_OFFSET_REG	<7:0>	WR	Offset (shift) the negative side of the waveform by this unsigned value
ADDR_WG_DRV_INT_REG	<31:0>	RD/WR	Write access: Bit 0: enable interrupting

			process. Bit 1: clear first address interrupt when enabled Bit 2: clear second address interrupt when enabled Bits <15:8>: first address interrupt. Enable the APB interrupt signal when wave gen arrives at this wave form address (there are 64 points wave form, hence, 64 addresses to be used as first address interrupt). Bits <23:16>: second address interrupt. Enable the APB interrupt signal when wave gen arrives at this wave form address (there are 64 points wave form, hence, 64 addresses to be used as second address interrupt). Rest of bits reserved. Read Access: Bits <7:0>: wave generator's number that we are reading Bit <8>: interrupt enabled Bit <9>: first address interrupt happened. Enabled when wave gen arrives at first wave form address. Bit <10>: second address interrupt happened. Enabled when wave gen arrives at second wave form address. Bits <23:16>: reporting the first address for the interrupt. Bits <31:24>: reporting the second address for the interrupt.
ADDR_WG_DRV_ISEL_REG	<2:0	WR	Current select value

	>		
ADDR_WG_DRV_SW_CONFIG_REG	<7:0>	WR	Switch selection for the electrode. Which switch(es) should be used for this electrode. Bit 0: switch 0 is used for the electrode. Bit 1: switch 1 is used for the electrode and so on. Combination of bits can be used.