A31S134 User's Manual

Cortex-M0+ based 32-bit General-Purpose Microcontroller with 128KB + 4KB FlashROM and 16KB SRAM

UM Rev. 1.01

Introduction

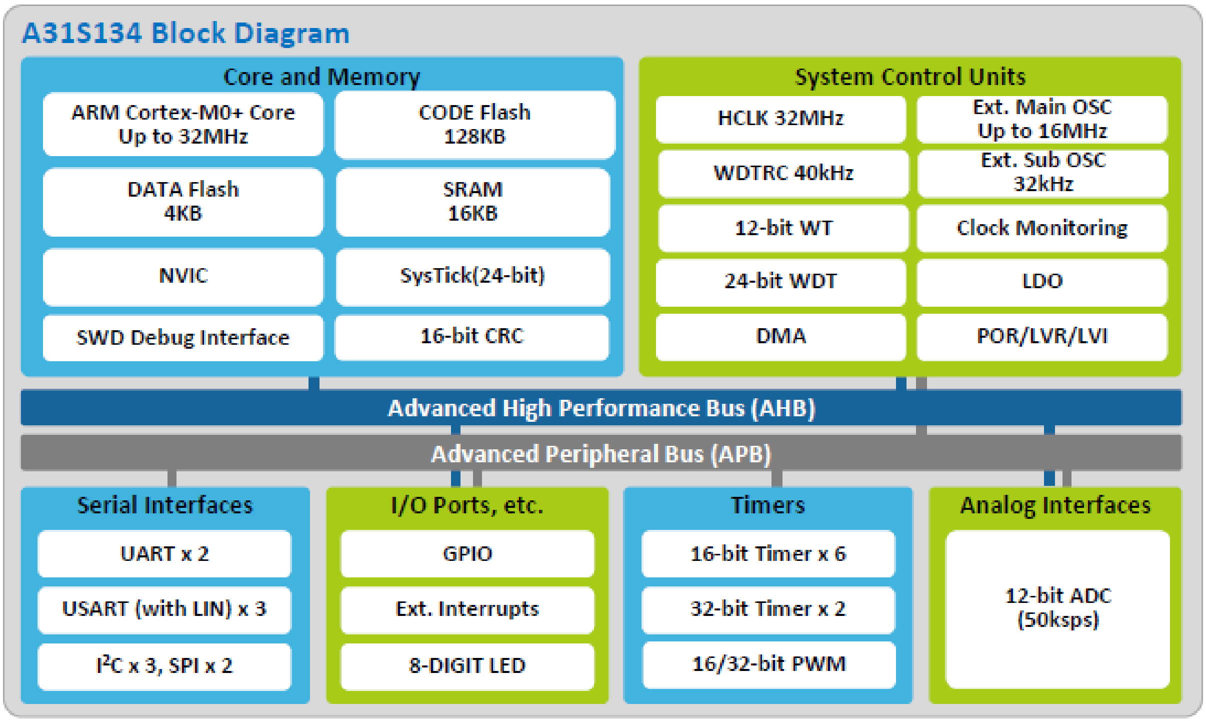
This user's manual contains complete information for application developers who use A31S134 for their specific needs.

The A31S134 is a 32-bit general-purpose microcontroller for various appliances. To meet the requirements for complexity and high performance in consumer electronics, the A31S134 incorporates Arm's high-speed 32-bit Cortex-M0+ core and has up to 128 KB of flash memory, 4 KB of data flash memory, and 16 KB of SRAM.

As shown in the following figure, the A31S134 has various peripherals such as 16-bit timers, 32-bit timers, 12-bit ADC, CRC generator, UART, USART, I2C, SPI, DMA, etc. The A31S134 also has a POR, LVR, LVI, and an internal RC oscillator.

The A31S134 supports sleep and deep sleep modes to reduce power consumption.

Figure 1. A31S134 Block Diagram



Reference Document

- Document ‘DDI 0484C’ is provided by Arm and contains information of the Cortex-M0+.
- A31S134 Datasheet is provided by ABOV Semiconductor and available at www.abovsemi.com.

Contents

List of Tables

List of Figures

1. Description

The A31S134 is a 32-bit high-performance microcontroller with up to 128 KB of flash memory. It is a powerful microcontroller which provides effective solutions to various electrical appliances which require both low power consumption and high performance (Arm Cortex-M0+ core).

1.1 Product Category Definition

Table 1 gives an overview of memory density versus product line. The present document describes the superset of features for each product category. See Table 2 for the list of features per category.

Table 1. A31S134 Memory Density

Memory Density		Category
Flash	RAM	
128 KB	16 KB	A31S134

1.2 Availability of Peripherals

Table 2 summarizes and lists product specific features available in the A31S134 considering the largest package.

For availability of peripherals and their numbers across all sale types, refer to the device's datasheet: **Product selection table** or **Ordering information**.

Table 2. A31S134 Features and Peripherals

Peripherals		Description
Core	CPU	• Maximum operating frequency: 32 MHz • 32-bit Arm Cortex-M0+ core • CPU register set: - Uses general-purpose registers specified by the 16-bit Thumb® instruction set - Main stack pointer (MSP) and process stack pointer (PSP): R13 - Link register (LR): R14 - Program counter (PC): R15 • Data ordering format: Little-Endian • AHB / APB
	Interrupt	• NVIC (Nested-Vectored Interrupt Controller) • Up to 32 peripheral interrupts supported. • Assignable with four different priority levels • 2-bit wide group priority: 4-level priority
Memory	Code Flash	• Capacity: 128 KB • A high-capacity code flash memory built in • 256 bytes page erase and 128 KB bulk erase • Programming in 256-byte sized page and 4-byte sized word • Read protection • Self-programming • CRC code generation and verification for the Flash memory • Endurance: 10,000 cycles • Lifetime: 10 years
	Data Flash	• Capacity: 4 KB • 32 bytes page erase and 4 KB bulk erase • Programming in 32-byte sized page and 4-byte sized word • Read protection • Endurance: 100,000 cycles • Lifetime: 10 years
	SRAM	• Capacity: 16 KB • Usable as a program's work area • Fast execution for time-critical code • Part of the SRAM can be remapped into an interrupt vector area

Table 2. A31S134 Features and Peripherals (continued)

Peripherals		Description
System Control Unit (SCU)	Operating Frequency	• From 2 MHz up to 32 MHz (HIRC, XMOSC)
	Clock	• High-speed internal oscillator (HIRC) - 32 MHz (±1.5% @0°C to +50°C) - 32 MHz (±2% @−40°C to +85°C) - 32 MHz (±3% @−40°C to +105°C) • Low-speed internal oscillator (WDTRC) - 40 kHz (±15% @−40°C to +105°C) • External main oscillator (XMOSC): 2 MHz to 16 MHz • External sub-oscillator (XSOSC): 32.768 kHz
	Clock Monitoring	• System Fail-Safe function by Clock Monitoring - High frequency internal RC oscillator (HIRC) - External main oscillator (XMOSC) - External sub oscillator (XSOSC) - Main system clock (MCLK)
	Operating Mode	• RUN mode • SLEEP mode • DEEP-SLEEP (STOP) mode
	Reset	• nRESET pin reset • Core reset • Software reset • POR (Power-On Reset) • LVR (Low-Voltage Reset) • WDTR (WatchDog Timer Reset) • Clock monitoring reset
	VDC	• Low-dropout (LDO) regulator built in for low-voltage operation
	POR	• The POR generator detects an internal 1.2 V and generates a reset signal
	LVI	• 12 low-voltage detection levels • Supports interrupts • Supports wake-up from SLEEP and DEEP SLEEP modes
	Wake-up	• Wake-up by an external interrupt (GPIO) pin • Wake-up by Watch Timer (WT) • Wake-up by a watchdog timer (WDT) • Wake-up by TIMER20 • Wake-up by a low-voltage indicator (LVI)
	Wake-up Timer	• 16-bit down count timer • Under flow interrupt • Counting stabilization time when returning from DEEP SLEEP mode

Table 2. A31S134 Features and Peripherals (continued)

Peripherals		Description
General-Purpose I/O (GPIO)		• Input/Output (I/O) port for general purposes • 64-LQFP-1010 - I/O pins: 61 • 48-LQFP-0707 - I/O pins: 45 • 44-LQFP-1010 - I/O pins: 41 • 32-LQFP-0707 - I/O pins: 29 • 32-QFN-0505 - I/O pins: 29 • Each pin can be set for one of the following modes: - Push-pull output - Open drain output - Input • The use of each pin can be set by the registers setting. • PB[11:0], PC[7:0], and PE[8:0] pins can be configured as external interrupt sources, either the level-trigger / edge-trigger interrupt or the rising-edge / falling-edge / both-edge interrupts. • Pull-up or pull-down resistor, and debouncing can be set for each pin. • Each pin bit can be individually set or reset. • Wake-up events triggered by external asynchronous inputs
LED Display Driver		• Up to 8 pins to drive sink currents up to 120 mA • PD0 to PD7
Direct Memory Access Controller (DMA)		• 2-ch direct memory access (DMA) support peripherals • 8-/16-/ 32-bit data transfers • Compatible with 20 different types of peripherals - USART10, USART11, USART12 - UART0, UART1 - I2C0, I2C1, I2C2 - SPI0, SPI1
System Timer	WDT	• 24-bit down-count timer • Reset and periodic interrupts • WDTRC or PCLK is selectable for input clock. • Master configurable by configure option • Under flow Interrupt
	WUT	• 16-bit down-count timer • Input clock is HCLK divided by 32. • Underflow interrupt • Counting stabilization time when wake-up from deep sleep mode
	WT	• 12-bit up-count timer • Input clock selection • XSOSC is selectable • WDTRC is selectable • Divided MCLK is selectable • Period interrupt

Table 2. A31S134 Features and Peripherals (continued)

Peripherals		Description
Timer	16-bit Timer	• General-purpose 16-bit up-count timer • Six channels - Timer 1n capture port (T1nCAP) input channels - Timer 1n output port (T1nOUT) output channels • Timer operating modes - Timer/counter mode - Capture mode - PPG one-shot mode - PPG repeat mode • Interrupt events - Timer/counter match interrupt - Timer capture interrupt • Input clock selection - PCLK clock divided by prescaler is selectable - External clock is selectable • Timer signals can be generated through T1nOUT pins • 12-bit prescaler
	32-bit Timer	• General-purpose 32-bit up-count timer • Two channels - Timer 2n capture port (T2nCAP) input channels - Timer 2n output port (T2nOUT) output channels • Timer operating modes - Timer/counter mode - Capture mode - PPG one-shot mode - PPG repeat mode • Interrupt events - Timer/counter match interrupt - Timer capture interrupt • Input clock selection - PCLK clock divided by prescaler is selectable. - External clock is selectable (T20, T21) - XSOSC clock is selectable (T20) • Timer signals can be generated through T2nOUT pins. • 12-bit prescaler

Table 2. A31S134 Features and Peripherals (continued)

Peripherals		Description
Serial Interface	USART	• 3 channels for asynchronous, synchronous, and serial peripheral interfaces • Asynchronous/synchronous modes (UART) • 5- to 9-bit data transfers • Even / Odd / Non-parity generation and checking • 1-bit or 2-bit stop bit generation and checking • 12-bit baud rate generator • Receive time out function from start bit (asynchronous only) • Receive character detection • Auto baud rate detection • 1-wire half-duplex communication • Local interconnection network (LIN) • Tx break transmit and Rx break detection • Serial peripheral interface (SPI) • Master/slave operation • Loop-back function • 8-bit data transmit/receive • Both least significant bit (LSB)-first and most significant bit (MSB)-first modes available
	UART	• Two channels for asynchronous serial communication ports • Configurable standard asynchronous communication bits (start, stop, and parity) • Flexible communication available through programming - 5- to 8-bit data transfers - Even / Odd / Non-parity generation and checking - 1-bit, 1.5-bit, or 2-bit stop bit generation and checking - 8-bit fraction controller and 16-bit baud rate generator
	SPI	• Two synchronous serial communication port channels • Master/slave operation • Loop-back mode • Programmable and flexible communication - 8-bit data transmit and receive - SPI clock speed - Both least significant bit (LSB)-first and most significant bit (MSB)-first modes available
	I2C	• Standard I2C communication protocol • Three channels supported. • Master and slave modes supported each channel. • 7-bit addressing supported for slave mode • SCL signal's high/low periods and SDA signal's hold time settable

Table 2. A31S134 Features and Peripherals (continued)

Peripherals	Description
12-bit A/D Converter	• 14 analog input channels • Software and timer match signal triggers supported • External reference selectable
Cyclic Redundancy Check	• 16-bit CRC generator • CRC operating modes: - CRC-CCITT (0x1021) - CRC-16 (0x8005) • Auto and user mode supported • CRC and checksum generation
Operating Voltage	• 1.8 V to 5.5 V
Operating Temperature	• Commercial grade (−40°C to +85°C) • Industrial Grade (−40°C to +105°C)
Package	• Five types of package options - 64-LQFP-1010 (0.5 mm pitch) - 48-LQFP-0707 (0.5 mm pitch) - 44-LQFP-1010 (0.8 mm pitch) - 32-LQFP-0707 (0.8 mm pitch) - 32-QFN-0505 (0.5 mm pitch)

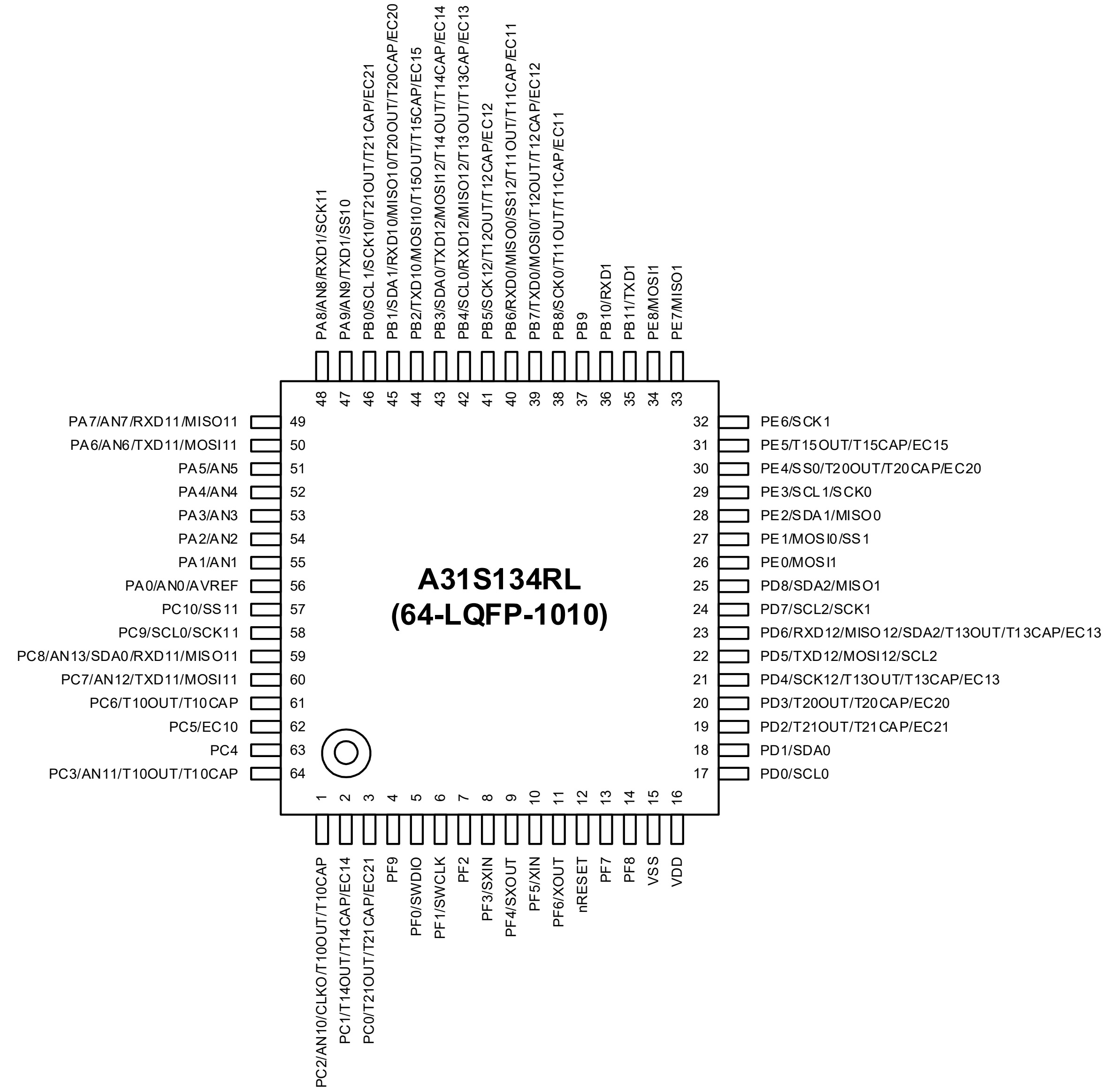
2. Pinouts and Pin Descriptions

In this chapter, pinouts and pin descriptions of the A31S134 are described.

2.1 Pinouts

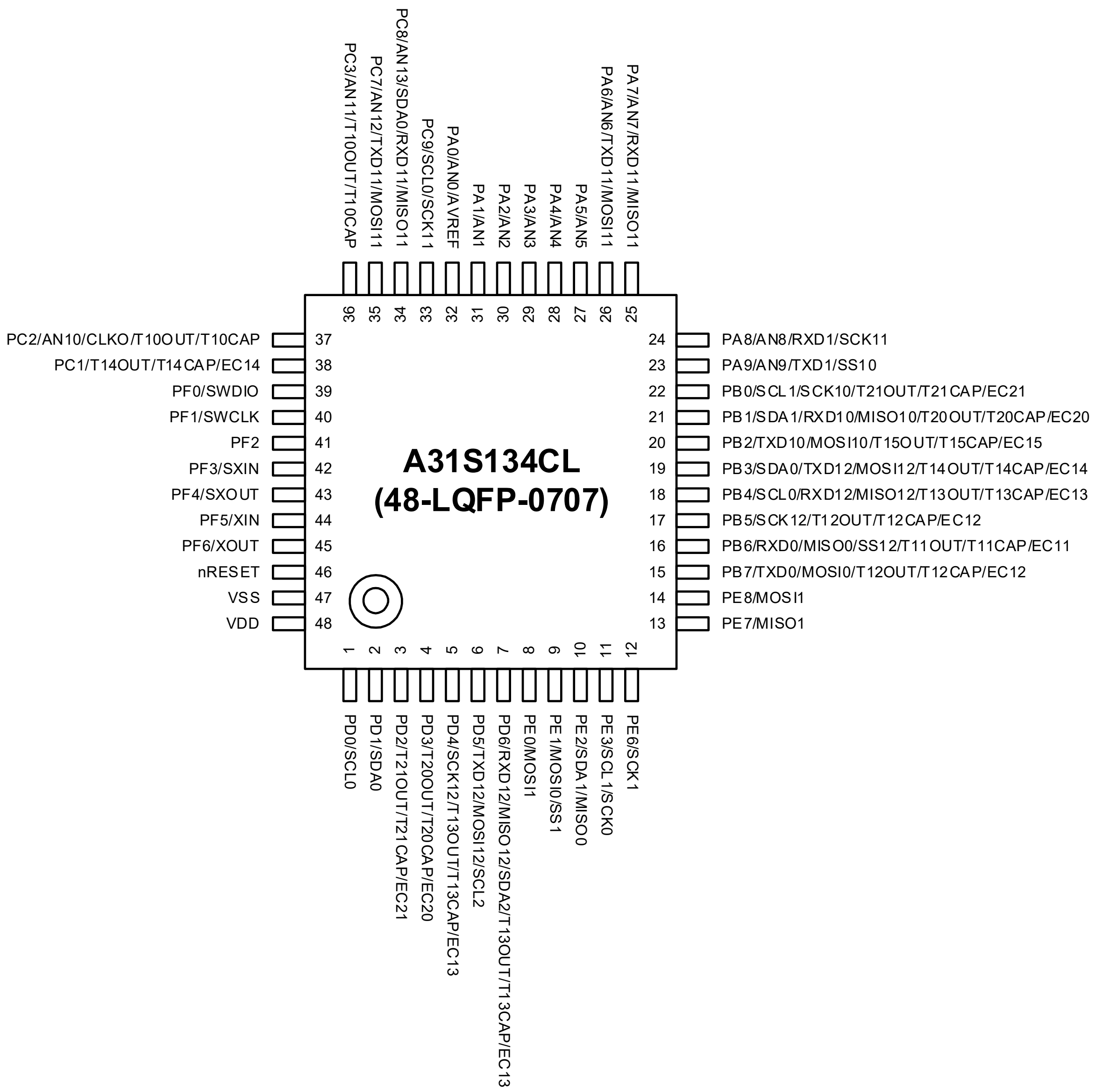
2.1.1 A31S134RL (64-LQFP)

Figure 2. 64-LQFP Pinouts



2.1.2 A31S134CL (48-LQFP)

Figure 3. 48-LQFP Pinouts

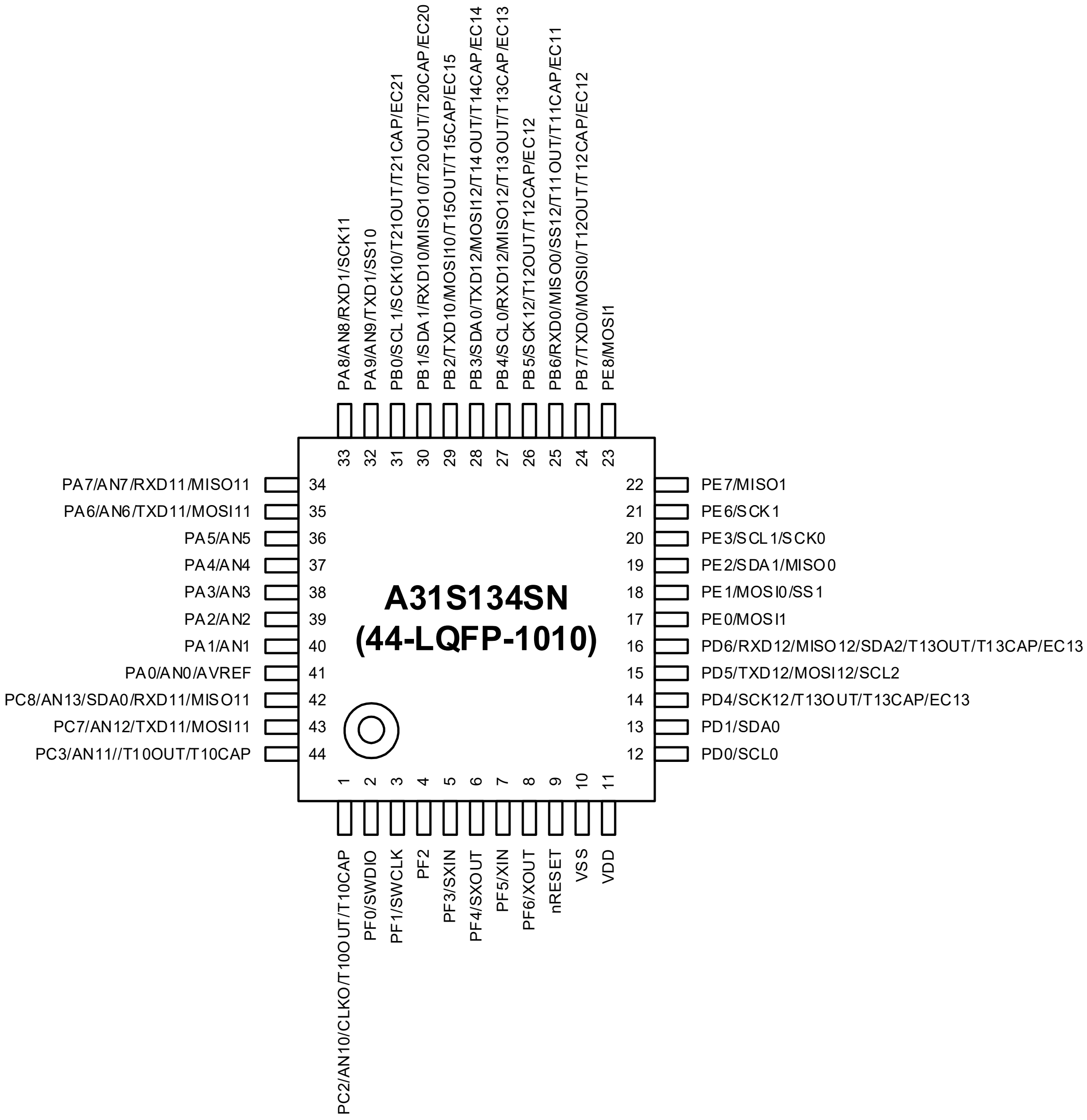


NOTE:

1. Each of the PB[11:8], PC[10, 6:4, 0], PD[8:7], PE[5:4], and PF[9:7] pins should be selected as a push-pull output or an input with pull-up or pull-down resistor by software control when the 48-LQFP package is used.

2.1.3 A31S134SN (44-LQFP)

Figure 4. 44-LQFP Pinouts

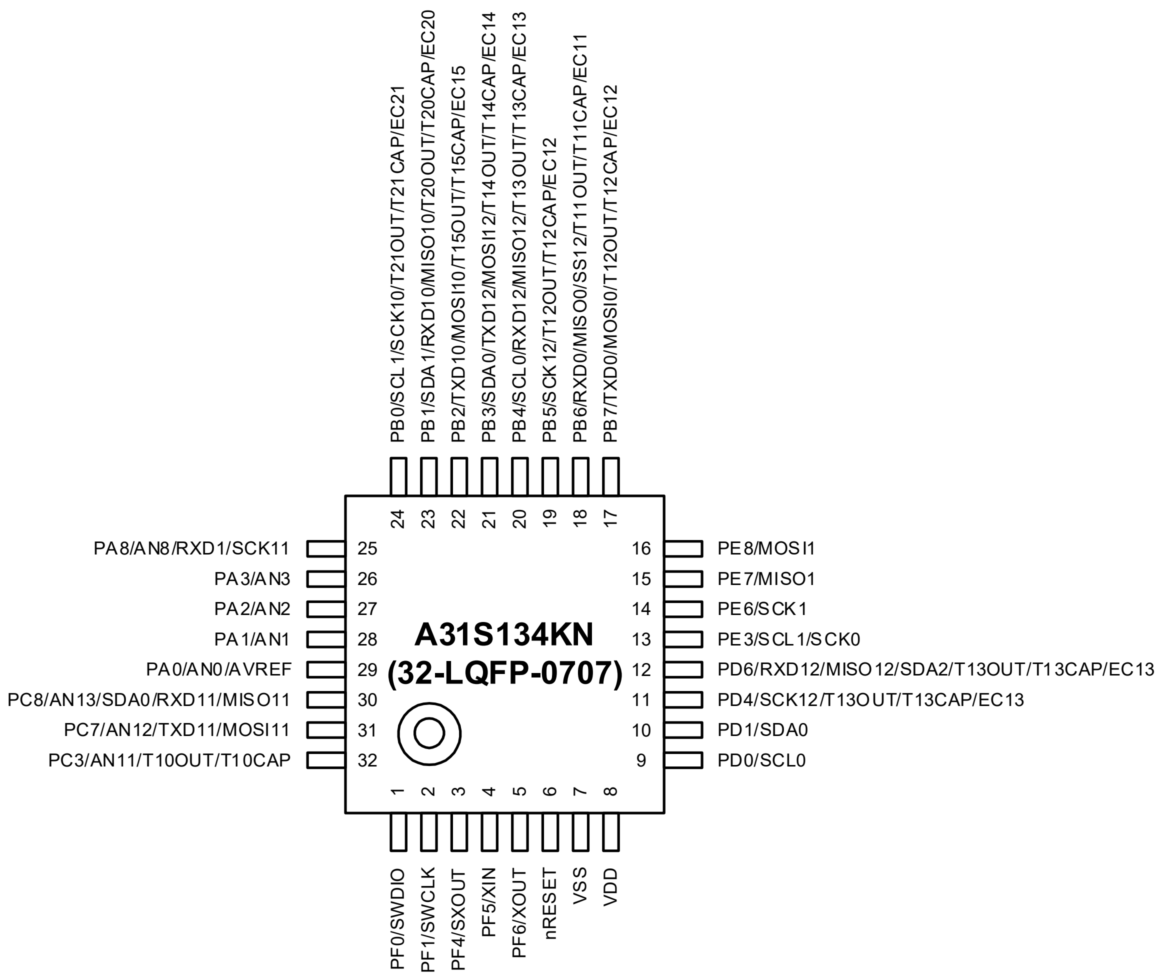

A31S134SN
(44-LQFP-1010)
PA8/AN8/RXD1/SCK11 33
PA9/AN9/TXD1/SS10 32
PB0/SCL1/SCK10/T21OUT/T21CAP/EC21 31
PB1/SDA1/RXD10/MISO10/T20OUT/T20CAP/EC20 30
PB2/TXD10/MOSI10/T15OUT/T15CAP/EC15 29
PB3/SDA0/TXD12/MOSI12/T14OUT/T14CAP/EC14 28
PB4/SCL0/RXD12/MISO12/T13OUT/T13CAP/EC13 27
PB5/SCK12/T12OUT/T12CAP/EC12 26
PB6/RXD0/MISO0/SS12/T11OUT/T11CAP/EC11 25
PB7/TXD0/MOSI0/T12OUT/T12CAP/EC12 24
PE8/MOSI1 23
PA7/AN7/RXD11/MISO11 34
PA6/AN6/TXD11/MOSI11 35
PA5/AN5 36
PA4/AN4 37
PA3/AN3 38
PA2/AN2 39
PA1/AN1 40
PA0/AN0/AVREF 41
PC8/AN13/SDA0/RXD11/MISO11 42
PC7/AN12/TXD11/MOSI11 43
PC3/AN11//T10OUT/T10CAP 44
22 PE7/MISO1
21 PE6/SCK1
20 PE3/SCL1/SCK0
19 PE2/SDA1/MISO0
18 PE1/MOSI0/SS1
17 PE0/MOSI1
16 PD6/RXD12/MISO12/SDA2/T13OUT/T13CAP/EC13
15 PD5/TXD12/MOSI12/SCL2
14 PD4/SCK12/T13OUT/T13CAP/EC13
13 PD1/SDA0
12 PD0/SCL0
1 PC2/AN10/CLKO/T10OUT/T10CAP
2 PF0/SWDIO
3 PF1/SWCLK
4 PF2
5 PF3/SXIN
6 PF4/SXOUT
7 PF5/XIN
8 PF6/XOUT
9 nRESET
10 VSS
11 VDD

NOTE:

1. Each of the PB[11:8], PC[10:9, 6:4, 1:0], PD[8:7, 3:2], PE[5:4], and PF[9:7] pins should be selected as a push-pull output or an input with pull-up or pull-down resistor by software control when the 44-LQFP package is used.

2.1.4 A31S134KN (32-LQFP)

Figure 5. 32-LQFP Pinouts

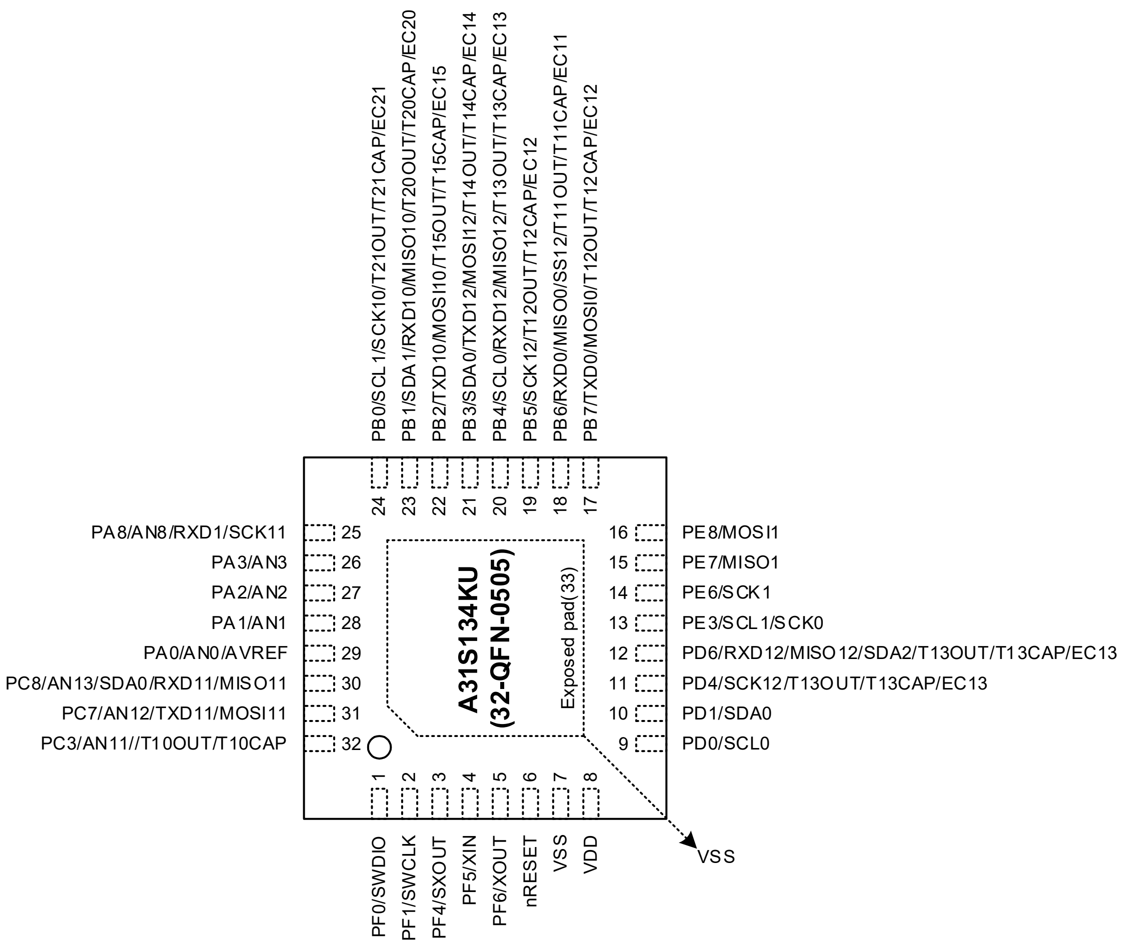

PB0/SCL1/SCK10/T21OUT/T21CAP/EC21
PB1/SDA1/RXD10/MISO10/T20OUT/T20CAP/EC20
PB2/TXD10/MOSI10/T15OUT/T15CAP/EC15
PB3/SDA0/TXD12/MOSI12/T14OUT/T14CAP/EC14
PB4/SCL0/RXD12/MISO12/T13OUT/T13CAP/EC13
PB5/SCK12/T12OUT/T12CAP/EC12
PB6/RXD0/MISO0/SS12/T11OUT/T11CAP/EC11
PB7/TXD0/MOSI0/T12OUT/T12CAP/EC12
24
23
22
21
20
19
18
17
PA8/AN8/RXD1/SCK11
PA3/AN3
PA2/AN2
PA1/AN1
PA0/AN0/AVREF
PC8/AN13/SDA0/RXD11/MISO11
PC7/AN12/TXD11/MOSI11
PC3/AN11/T10OUT/T10CAP
25
26
27
28
29
30
31
32
A31S134KN
(32-LQFP-0707)
16
15
14
13
12
11
10
9
PE8/MOSI1
PE7/MISO1
PE6/SCK1
PE3/SCL1/SCK0
PD6/RXD12/MISO12/SDA2/T13OUT/T13CAP/EC13
PD4/SCK12/T13OUT/T13CAP/EC13
PD1/SDA0
PD0/SCL0
1
2
3
4
5
6
7
8
PF0/SWDIO
PF1/SWCLK
PF4/SXOUT
PF5/XIN
PF6/XOUT
nRESET
VSS
VDD

NOTE:

1. Each of the PA[9, 7:4], PB[11:8], PC[10:9, 6:4, 2:0], PD[8:7, 5, 3:2], PE[5:4, 2:0], and PF[9:7, 3:2] pins should be selected as a push-pull output or an input with pull-up or pull-down resistor by software control when the 32-LQFP package is used.

2.1.5 A31S134KU (32-QFN)

Figure 6. 32-QFN Pinouts


PB0/SCL1/SCK10/T21OUT/T21CAP/EC21
PB1/SDA1/RXD10/MISO10/T20OUT/T20CAP/EC20
PB2/TXD10/MOSI10/T15OUT/T15CAP/EC15
PB3/SDA0/TXD12/MOSI12/T14OUT/T14CAP/EC14
PB4/SCL0/RXD12/MISO12/T13OUT/T13CAP/EC13
PB5/SCK12/T12OUT/T12CAP/EC12
PB6/RXD0/MISO0/SS12/T11OUT/T11CAP/EC11
PB7/TXD0/MOSI0/T12OUT/T12CAP/EC12
24
23
22
21
20
19
18
17
PA8/AN8/RXD1/SCK11 25
PA3/AN3 26
PA2/AN2 27
PA1/AN1 28
PA0/AN0/AVREF 29
PC8/AN13/SDA0/RXD11/MISO11 30
PC7/AN12/TXD11/MOSI11 31
PC3/AN11//T10OUT/T10CAP 32
16 PE8/MOSI1
15 PE7/MISO1
14 PE6/SCK1
13 PE3/SCL1/SCK0
12 PD6/RXD12/MISO12/SDA2/T13OUT/T13CAP/EC13
11 PD4/SCK12/T13OUT/T13CAP/EC13
10 PD1/SDA0
9 PD0/SCL0
A31S134KU
(32-QFN-0505)
Exposed pad(33)
1
2
3
4
5
6
7
8
PF0/SWDIO
PF1/SWCLK
PF4/SXOUT
PF5/XIN
PF6/XOUT
nRESET
VSS
VDD
VSS

NOTE:

1. Each of the PA[9, 7:4], PB[11:8], PC[10:9, 6:4, 2:0], PD[8:7, 5, 3:2], PE[5:4, 2:0], and PF[9:7, 3:2] pins should be selected as a push-pull output or an input with pull-up or pull-down resistor by software control when the 32-QFN package is used.

2.2 Pin Description

Pin configuration information contains one pair of power/ground and other dedicated pins. These multi-function pins have up to six selections of functions including GPIO. Configuration including pin ordering can be changed without notice.

Table 3. Pin Description

Pin No.				Pin Name	Type(1)	Description	Remark
64-pin	48-pin	44-pin	32-pin				
1	37	1	-	PC2*	IOUDS	Port C bit 2 input/output	
				AN10	IA	A/D converter analog input channel	
				CLKO	O	System clock output	
				T10OUT	O	Timer 10 pulse output	
				T10CAP	I	Timer 10 capture input	
2	38	-	-	PC1*	IOUDS	Port C bit 1 input/output	
				T14OUT	O	Timer 14 pulse output	
				T14CAP	I	Timer 14 capture input	
				EC14	I	Timer 14 event count input	
3	-	-	-	PC0*	IOUDS	Port C bit 0 input/output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				EC21	I	Timer 21 event count input	
4	-	-	-	PF9	IOUDS	Port F bit 9 input/output	
5	39	2	1	PF0	IOUDS	Port F bit 0 input/output	
				SWDIO*	I/O	SWD data input / output	Pull-up when reset
6	40	3	2	PF1	IOUDS	Port F bit 1 input/output	
				SWCLK*	I	SWD clock input	Pull-down when reset
7	41	4	-	PF2	IOUDS	Port F bit 2 input/output	
8	42	5	-	PF3*	IOUDS	Port F bit 3 input/output	
				SXIN	IA	Sub Oscillator Input	
9	43	6	3	PF4*	IOUDS	Port F bit 4 input/output	
				SXOUT	OA	Sub Oscillator Output	
10	44	7	4	PF5*	IOUDS	Port F bit 5 input/output	
				XIN	IA	Main Oscillator Input	
11	45	8	5	PF6*	IOUDS	Port F bit 6 input/output	
				XOUT	OA	Main Oscillator Output	
12	46	9	6	nRESET	Input	External Reset Input	Always pull-up
13	-	-	-	PF7	IOUDS	Port F bit 7 input/output	
14	-	-	-	PF8	IOUDS	Port F bit 8 input/output	
15	47	10	7	VSS	P	Ground	
16	48	11	8	VDD	P	VDD	
17	1	12	9	PD0*	IOUDS	Port D bit 0 input/output	
				SCL0	I/O	I2C clock input/output	
18	2	13	10	PD1*	IOUDS	Port D bit 1 input/output	
				SDA0	I/O	I2C data input/output	

Pin No.				Pin Name	Type(1)	Description	Remark
64-pin	48-pin	44-pin	32-pin				
19	3	-	-	PD2*	IOUDS	Port D bit 2 input/output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				EC21	I	Timer 21 event count input	
20	4	-	-	PD3*	IOUDS	Port D bit 3 input/output	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				EC20	I	Timer 20 event count input	
21	5	14	11	PD4*	IOUDS	Port D bit 4 input/output	
				SCK12	I/O	SPI clock input/output	
				T13OUT	O	Timer 13 pulse output	
				T13CAP	I	Timer 13 capture input	
				EC13	I	Timer 13 event count input	
22	6	15	-	PD5*	IOUDS	Port D bit 5 input/output	
				SCL2	I/O	I2C clock input/output	
				TXD12	O	UART data output	
				MOSI12	I/O	SPI master output, slave input	
23	7	16	12	PD6*	IOUDS	Port D bit 6 input/output	
				SDA2	I/O	I2C data input/output	
				RXD12	I	UART data input	
				MISO12	I/O	SPI master input, slave output	
				T13OUT	O	Timer 13 pulse output	
				T13CAP	I	Timer 13 capture input	
				EC13	I	Timer 13 event count input	
24	-	-	-	PD7*	IOUDS	Port D bit 7 input/output	
				SCL2	I/O	I2C clock input/output	
				SCK1	I/O	SPI clock input/output	
25	-	-	-	PD8*	IOUDS	Port D bit 8 input/output	
				SDA2	I/O	I2C data input/output	
				MISO1	I/O	SPI master input, slave output	
26	8	17	-	PE0*	IOUDS	Port E bit 0 input/output	
				MOSI1	I/O	SPI master output, slave input	
27	9	18	-	PE1*	IOUDS	Port E bit 1 input/output	
				MOSI0	I/O	SPI master output, slave input	
				SS1	I	SPI slave select input	
28	10	19	-	PE2*	IOUDS	Port E bit 2 input/output	
				SDA1	I/O	I2C data input/output	
				MISO0	I/O	SPI master input, slave output	
29	11	20	13	PE3*	IOUDS	Port E bit 3 input/output	
				SCL1	I/O	I2C clock input/output	
				SCK0	I/O	SPI clock input/output	

Pin No.				Pin Name	Type[1]	Description	Remark
64-pin	48-pin	44-pin	32-pin				
30	-	-	-	PE4*	IOUDS	Port E bit 4 input/output	
				SS0	I	SPI slave select input	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				EC20	I	Timer 20 event count input	
31	-	-	-	PE5*	IOUDS	Port E bit 5 input/output	
				T15OUT	O	Timer 15 pulse output	
				T15CAP	I	Timer 15 capture input	
				EC15	I	Timer 15 event count input	
32	12	21	14	PE6*	IOUDS	Port E bit 6 input/output	
				SCK1	I/O	SPI clock input/output	
33	13	22	15	PE7*	IOUDS	Port E bit 7 input/output	
				MISO1	I/O	SPI master input, slave output	
34	14	23	16	PE8*	IOUDS	Port E bit 8 input/output	
				MOSI1	I/O	SPI master output, slave input	
35	-	-	-	PB11*	IOUDS	Port B bit 11 input/output	
				TXD1	O	UART data output	
36	-	-	-	PB10*	IOUDS	Port B bit 10 input/output	
				RXD1	I	UART data input	
37	-	-	-	PB9	IOUDS	Port B bit 9 input/output	
38	-	-	-	PB8*	IOUDS	Port B bit 8 input/output	
				SCK0	I/O	SPI clock input/output	
				T11OUT	O	Timer 11 pulse output	
				T11CAP	I	Timer 11 capture input	
				EC11	I	Timer 11 event count input	
39	15	24	17	PB7*	IOUDS	Port B bit 7 input/output	
				TXD0	O	UART data output	
				MOSI0	I/O	SPI master output, slave input	
				T12OUT	O	Timer 12 pulse output	
				T12CAP	I	Timer 12 capture input	
				EC12	I	Timer 12 event count input	
40	16	25	18	PB6*	IOUDS	Port B bit 6 input/output	
				RXD0	I	UART data input	
				MISO0	I/O	SPI master input, slave output	
				SS12	I	SPI slave select input	
				T11OUT	O	Timer 11 pulse output	
				T11CAP	I	Timer 11 capture input	
				EC11	I	Timer 11 event count input	
41	17	26	19	PB5*	IOUDS	Port B bit 5 input/output	
				SCK12	I/O	SPI clock input/output	
				T12OUT	O	Timer 12 pulse output	
				T12CAP	I	Timer 12 capture input	
				EC12	I	Timer 12 event count input	

Pin No.				Pin Name	Type(1)	Description	Remark
64-pin	48-pin	44-pin	32-pin				
42	18	27	20	PB4*	IOUDS	Port B bit 4 input/output	
				SCL0	I/O	I2C clock input/output	
				RXD12	I	UART data input	
				MISO12	I/O	SPI master input, slave output	
				T13OUT	O	Timer 13 pulse output	
				T13CAP	I	Timer 13 capture input	
				EC13	I	Timer 13 event count input	
43	19	28	21	PB3*	IOUDS	Port B bit 3 input/output	
				SDA0	I/O	I2C data input/output	
				TXD12	O	UART data output	
				MOSI12	I/O	SPI master output, slave input	
				T14OUT	O	Timer 14 pulse output	
				T14CAP	I	Timer 14 capture input	
				EC14	I	Timer 14 event count input	
44	20	29	22	PB2*	IOUDS	Port B bit 2 input/output	
				TXD10	O	UART data output	
				MOSI10	I/O	SPI master output, slave input	
				T15OUT	O	Timer 15 pulse output	
				T15CAP	I	Timer 15 capture input	
				EC15	I	Timer 15 event count input	
45	21	30	23	PB1*	IOUDS	Port B bit 1 input/output	
				SDA1	I/O	I2C data input/output	
				RXD10	I	UART data input	
				MISO10	I/O	SPI master input, slave output	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				EC20	I	Timer 20 event count input	
46	22	31	24	PB0*	IOUDS	Port B bit 0 input/output	
				SCL1	I/O	I2C clock input/output	
				SCK10	I/O	SPI clock input/output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				EC21	I	Timer 21 event count input	
47	23	32	-	PA9*	IOUDS	Port A bit 9 input/output	
				AN9	IA	A/D converter analog input channel	
				SS10	I	SPI slave select input	
				TXD1	O	UART data output	
48	24	33	25	PA8*	IOUDS	Port A bit 8 input/output	
				AN8	IA	A/D converter analog input channel	
				SCK11	I/O	SPI clock input/output	
				RXD1	I	UART data input	
49	25	34	-	PA7*	IOUDS	Port A bit 7 input/output	
				AN7	IA	A/D converter analog input channel	
				RXD11	I	UART data input	
				MISO11	I/O	SPI master input, slave output	

Pin No.				Pin Name	Type(1)	Description	Remark
64-pin	48-pin	44-pin	32-pin				
50	26	35	-	PA6*	IOUDS	Port A bit 6 input/output	
				AN6	IA	A/D converter analog input channel	
				TXD11	O	UART data output	
				MOSI11	I/O	SPI master output, slave input	
51	27	36	-	PA5*	IOUDS	Port A bit 5 input/output	
				AN5	IA	A/D converter analog input channel	
52	28	37	-	PA4*	IOUDS	Port A bit 4 input/output	
				AN4	IA	A/D converter analog input channel	
53	29	38	26	PA3*	IOUDS	Port A bit 3 input/output	
				AN3	IA	A/D converter analog input channel	
54	30	39	27	PA2*	IOUDS	Port A bit 2 input/output	
				AN2	IA	A/D converter analog input channel	
55	31	40	28	PA1*	IOUDS	Port A bit 1 input/output	
				AN1	IA	A/D converter analog input channel	
56	32	41	29	PA0*	IOUDS	Port A bit 0 input/output	
				AN0	IA	A/D converter analog input channel	
				AVREF	IA	A/D converter reference input	
57	-	-	-	PC10*	IOUDS	Port C bit 10 input/output	
				SS11	I	SPI slave select input	
58	33	-	-	PC9*	IOUDS	Port C bit 9 input/output	
				SCL0	I/O	I2C clock input/output	
				SCK11	I/O	SPI clock input/output	
59	34	42	30	PC8*	IOUDS	Port C bit 8 input/output	
				AN13	IA	A/D converter analog input channel	
				SDA0	I/O	I2C data input/output	
				RXD11	I	UART data input	
				MISO11	I/O	SPI master input, slave output	
60	35	43	31	PC7*	IOUDS	Port C bit 7 input/output	
				AN12	IA	A/D converter analog input channel	
				TXD11	O	UART data output	
				MOSI11	I/O	SPI master output, slave input	
61	-	-	-	PC6*	IOUDS	Port C bit 6 input/output	
				T10OUT	O	Timer 10 pulse output	
				T10CAP	I	Timer 10 capture input	
62	-	-	-	PC5*	IOUDS	Port C bit 5 input/output	
				EC10	I	Timer 10 event count input	
63	-	-	-	PC4	IOUDS	Port C bit 4 input/output	
64	36	44	32	PC3*	IOUDS	Port C bit 3 input/output	
				AN11	IA	A/D converter analog input channel	
				T10OUT	O	Timer 10 pulse output	
				T10CAP	I	Timer 10 capture input	

NOTES:

1. I = Input, O = Output, U = Pull-up, D = Pull-down, S = Schmitt-Trigger Input Type, C = CMOS Input Type, A = Analog, P = Power
2. * means ‘selected pin function after reset condition’.

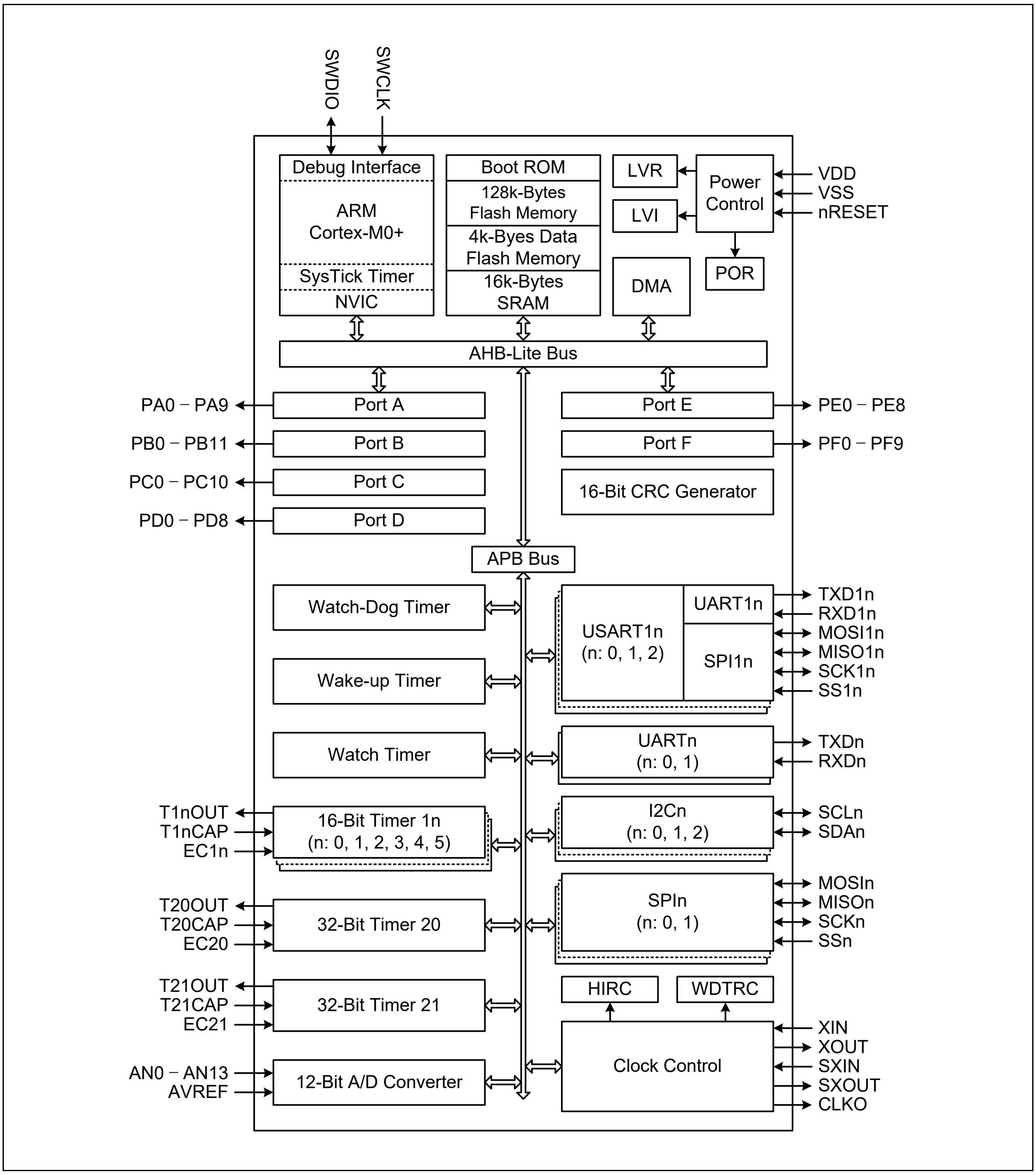
3. Pin order may be changed with revision notice.
4. nRESET and PF0 (SWDIO) are the default pull-up pins. PF1 (SWCLK) is the default pull-down pin.
5. An unused pin should not be configured as an input floating.
6. After a reset, the PF0 and PF1 pins are configured as SWDIO and SWCLK alternative functions, and the internal pull-down for the serial wire clock (SWCLK) and the internal pull-up for the serial wire data I/O (SWDIO) are enabled.
7. The SWCLK and SWDIO pins should not be switched to other functions while they are being used.

3. System and Memory Overview

In this chapter, the system and memory overview of the A31S134 are described.

3.1 System Architecture

Figure 7. System Block Diagram



3.1.1 Cortex-M0+ Core

The A31S134 microcontroller uses the energy efficient Cortex-M0+ core from Arm, which is optimized for low-power consumption and features a highly efficient 32-bit architecture. The Cortex-M0+ is based on the ARMv6-M Thumb instruction set and includes 16-bit instructions with Thumb-2 technology, allowing for improved performance and energy efficiency. It also includes a simple 24-bit system timer (SysTick) that can function as a real-time operating system or a counter. Additionally, the Cortex-M0+ features an integrated Nested-Vectored Interrupt Controller (NVIC) for deterministic interrupt handling, hardware single-cycle multiplication for efficient computation, and supports SWD debugging features.

Refer to the technical reference manual **ARM DDI 0484C** for detailed information on Cortex-M0+.

3.1.2 Interrupt Controller

The Cortex-M0+ processor has an embedded interrupt controller named NVIC (Nested-Vector Interrupt Controller). The A31S134 has an additional interrupt control block for controlling 32 interrupt sources generated by internal peripherals.

To use interrupts from internal peripherals, both the NVIC and the interrupt control block must be configured properly.

This document only describes the peripheral interrupt controller, therefore for more information on NVIC inside the Cortex-M0+ processor, please refer to the technical reference manual “ARM DDI 0484C” on the Arm technical document site.

Table 4. Interrupt Vector Map

Priority	Vector Address	Interrupt Source
-16	0x0000_0000	Stack Pointer
-15	0x0000_0004	Reset Address
-14	0x0000_0008	NMI Exception
-13	0x0000_000C	Hard Fault Exception
-12	0x0000_0010	Reserved
-11	0x0000_0014	
-10	0x0000_0018	
-9	0x0000_001C	
-8	0x0000_0020	
-7	0x0000_0024	
-6	0x0000_0028	
-5	0x0000_002C	SVCall Exception
-4	0x0000_0030	Reserved
-3	0x0000_0034	
-2	0x0000_0038	PenSV Exception
-1	0x0000_003C	SysTick Exception
0	0x0000_0040	LVI Interrupt
1	0x0000_0044	WUT Interrupt

Priority	Vector Address	Interrupt Source
2	0x0000_0048	WDT Interrupt
3	0x0000_004C	EINT0 Interrupt
4	0x0000_0050	EINT1 Interrupt
5	0x0000_0054	EINT2 Interrupt
6	0x0000_0058	EINT3 Interrupt
7	0x0000_005C	TIMER10 Interrupt
8	0x0000_0060	TIMER11 Interrupt
9	0x0000_0064	TIMER12 Interrupt
10	0x0000_0068	I2C0 Interrupt
11	0x0000_006C	USART10 Interrupt
12	0x0000_0070	WT Interrupt
13	0x0000_0074	Reserved
14	0x0000_0078	I2C1 Interrupt
15	0x0000_007C	TIMER20 Interrupt
16	0x0000_0080	TIMER21 Interrupt
17	0x0000_0084	USART11 Interrupt
18	0x0000_0088	ADC Interrupt
19	0x0000_008C	UART0 Interrupt
20	0x0000_0090	UART1 Interrupt
21	0x0000_0094	TIMER13 Interrupt
22	0x0000_0098	TIMER14 Interrupt
23	0x0000_009C	TIMER15 Interrupt
24	0x0000_00A0	DMACH[1:0] Interrupt
25	0x0000_00A4	I2C2 Interrupt
26	0x0000_00A8	USART12 Interrupt
27	0x0000_00AC	Reserved
28	0x0000_00B0	
29	0x0000_00B4	SPI0 Interrupt
30	0x0000_00B8	SPI1 Interrupt
31	0x0000_00BC	Reserved

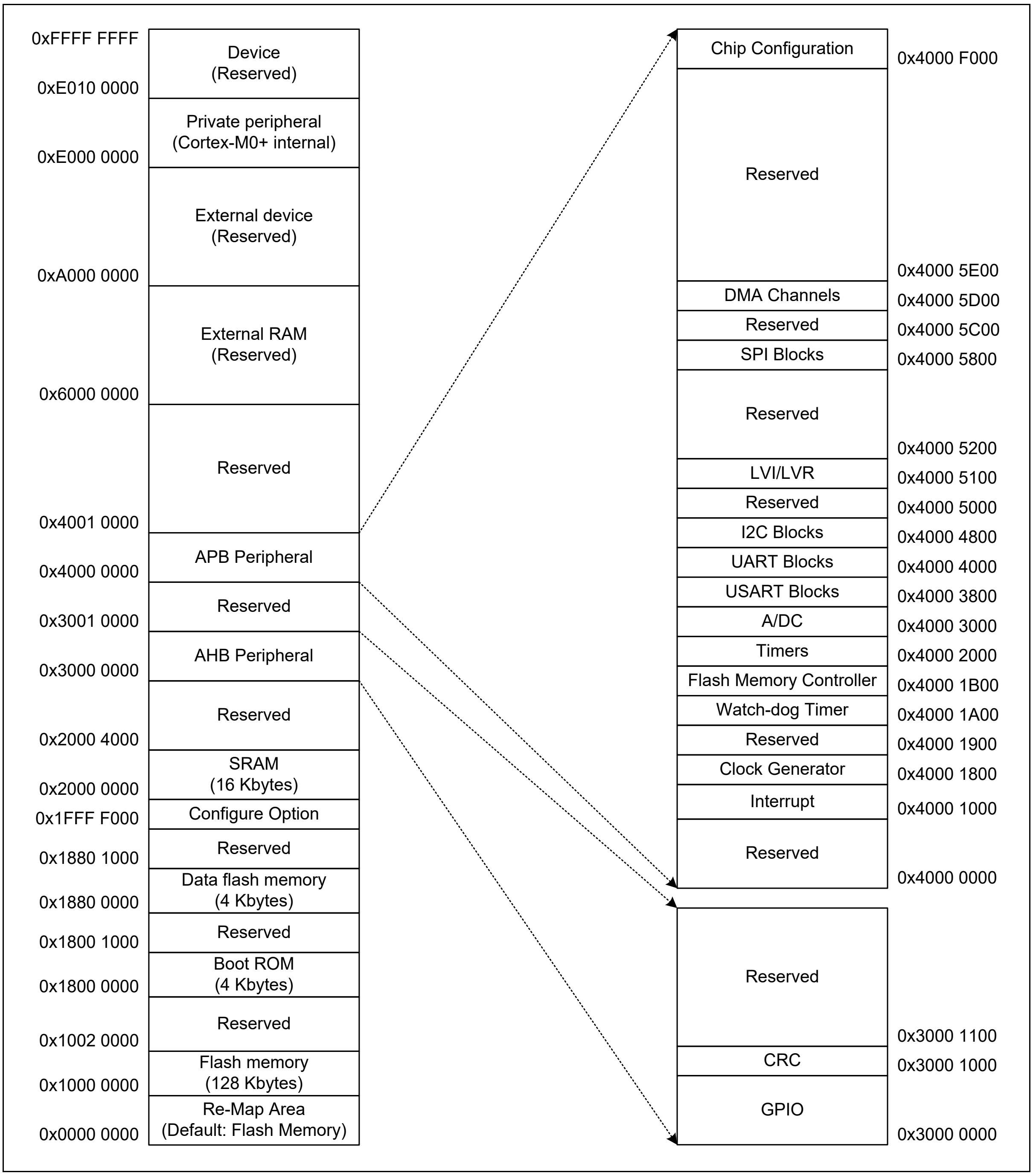
3.2 Memory Organization

3.2.1 Introduction

Program memory, data memory, registers, and I/O ports are organized in the same address space. The bytes are coded in memory in Little Endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte the most significant.

3.2.2 Memory Map and Register Boundary Address

Figure 8. Memory Map



All the memory map areas that are not allocated to on-chip memories and peripherals are considered “Reserved”. For the detailed mapping of available memory and register areas, refer to the following table.

The following table gives the boundary addresses of the peripherals available in the devices.

Table 5. Peripheral Address

Base Address	Peripheral Name
0x1000_0000	Flash memory
0x1880_0000	Data flash memory
0x2000_0000	Internal SRAM
0x3000_0000	GPIO
0x3000_1000	CRC
0x4000_1000	Interrupt
0x4000_1800	SCUCG
0x4000_1A00	WDT
0x4000_1B00	FMC
0x4000_2000	Timers
0x4000_3000	ADC
0x4000_3800	USARTs
0x4000_4000	UARTs
0x4000_4800	I2Cs
0x4000_5100	SCULV
0x4000_5800	SPIs
0x4000_5D00	DMAs
0x4000_F000	SCUCC

3.3 Memory Map

3.3.1 Embedded SRAM

The A31S134 has a block of 0-wait on-chip SRAM. The size of the SRAM is 16 KB, and its base address is 0x2000_0000.

This SRAM memory area is usually used for data memory and stack memory. Sometimes the code is dumped into the SRAM memory for fast operation or Flash erase / programming operation. This device does not support memory remap strategy. So, jump and return are required to perform the code in SRAM memory area.

3.3.2 Flash Memory Overview

The A31S134 provides internal 128 KB code flash memory and a controller. This is enough to control the general system. Self-programming is available and ISP and SWD programming is also supported in debugging mode.

CPU can access flash memory with zero wait state up to 32 MHz bus frequency.

3.3.3 Data Flash Memory Overview

The A31S134 provides internal 4 KB data flash memory and a controller. This area is used to save user data. But the area cannot be used for code. The data flash area is supported for self-programming and ISP.

4. System Control Unit (SCU)

4.1 SCU Introduction

The A31S134 has a built-in intelligent power control block which manages system analog blocks and operating modes. System Control Unit (SCU) module controls an internal reset and clock signals to maintain optimized system performance and power dissipation.

4.2 SCU Main Features

The clock features of A31S134 are as follows:

- Operating frequency: Up to 32 MHz
- High-Speed internal oscillator (HIRC): 2 MHz to 32 MHz
- Watchdog Timer internal oscillator (WDTRC): 40 kHz
- High-Speed External Oscillator (XMOSC): 2 MHz to 16 MHz
- Low-Speed External Oscillator (XSOSC): 32.768 kHz
- The A31S134 has clock monitoring as a system fail-safe function.

The reset features of A31S134 are as follows:

- nRESET pin reset
- CPU reset
- Software reset
- Power-On reset (POR)
- Low-Voltage Detect reset (LVR)
- Watchdog Timer reset (WDT)
- Clock monitoring reset

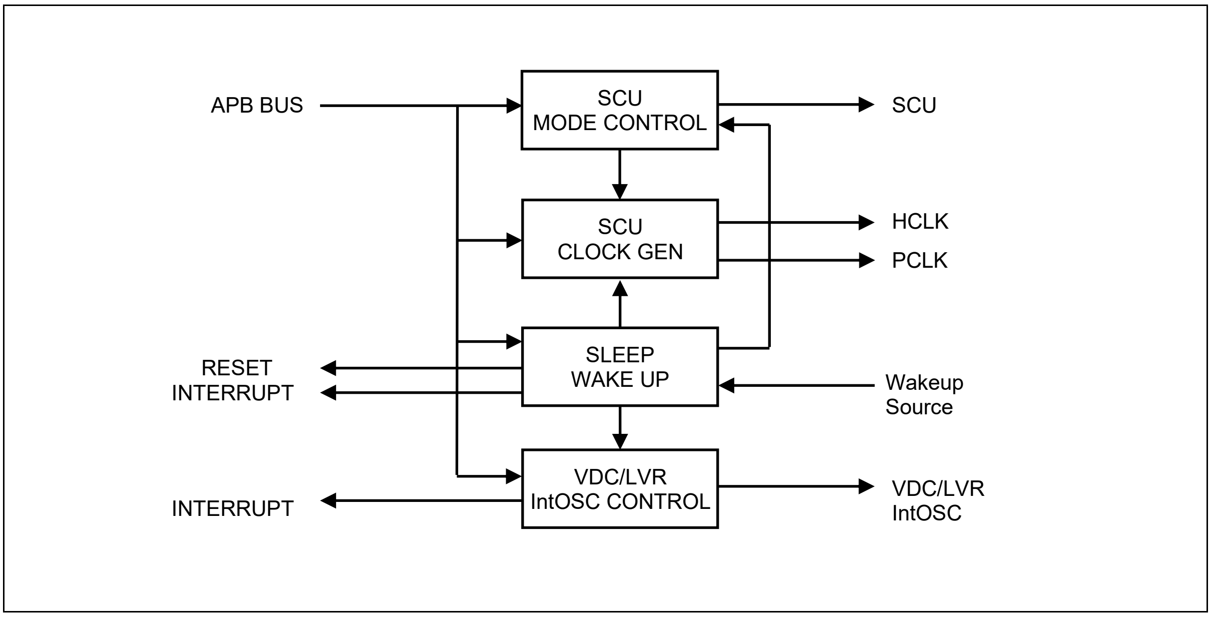
The operating mode features of A31S134 are as follows:

- RUN mode
- SLEEP mode
- DEEP-SLEEP mode

4.3 SCU Block Diagram

Figure 9 shows the SCU block diagram.

Figure 9. SCU Block Diagram

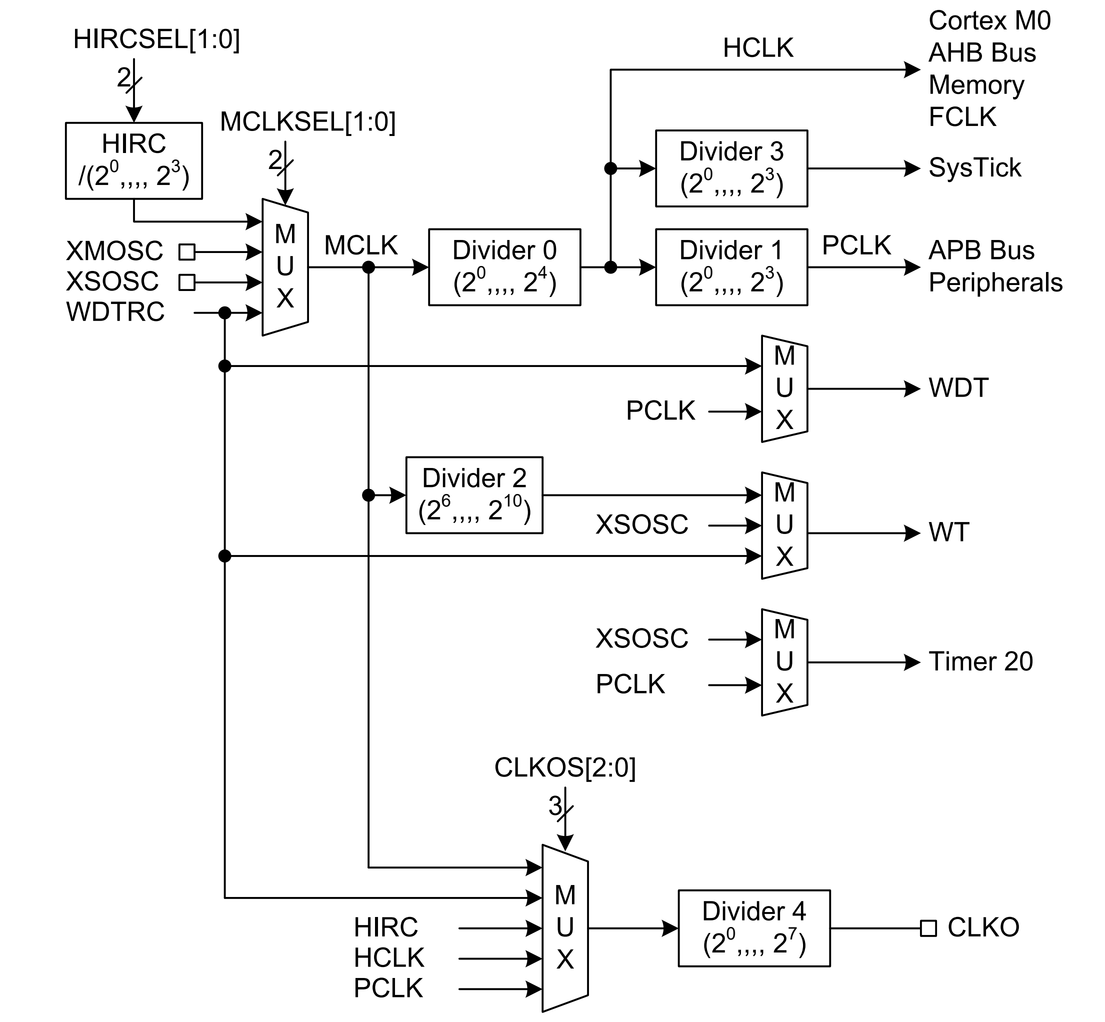


4.4 Clock System

The A31S134 has two main operating clocks. One is the HCLK, which supplies clocks to the CPU and AHB bus system, and the other is the PCLK, which supplies clocks to the peripheral systems.

Users can control the clock system variation using software. Figure 10 shows the clock system of the A31S134 and Table 6 shows the description for clock sources.

Figure 10. Clock Source Configuration



NOTE:

1. If the WRCMF[11:0] bits of configuration are not equal to 0x96D, the WDTRC oscillation can't be disabled by the WDTRCEN bit of SCU_CLKSRCR register.

Each mux that switches the clock source has a glitch-free circuit. So, a clock can be switched without glitch risks. When users change the clock mux control, be sure that both clock sources are alive. If either is not alive, the clock change operation stops, and the system shuts down (and will not recover).

Table 6. Clock Sources

Clock Name	Symbol	Frequency	Description
Main OSC	XMOSC	X-TAL (2 MHz to 16 MHz) External Clock (2 MHz to 32 MHz)	External Main Crystal OSC External Main Clock
Sub OSC	XSOSC	X-TAL (32.768 kHz)	External Sub Crystal OSC
Internal RC OSC	HIRC	2 MHz to 32 MHz	High-Speed Internal RC OSC
WDTRC OSC	WDTRC	40 kHz	Watchdog Timer RC OSC

4.4.1 HCLK Clock Domain

The HCLK clock feeds the clock to the CPU and AHB bus. Cortex-M0+ CPU requires two clocks, FCLK and HCLK. The FCLK is a free running clock and is always running except during power down mode. The HCLK can be stopped during SLEEP mode.

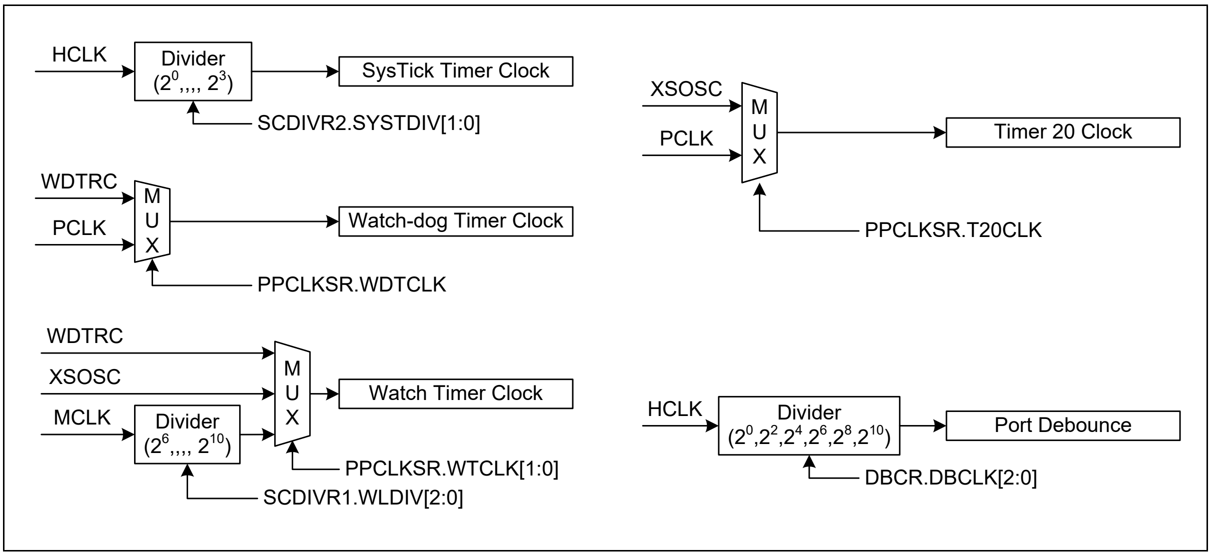
The HCLK clock operates the BUS system and memory systems. The maximum clock speed of the BUS operation is 32 MHz. The HCLK frequency should be limited to 32 MHz or lower.

4.4.2 Miscellaneous Clock Domain

Various clock sources are required for each functional block. The SCU provides clock source selectivity with dedicated pre-scaler for each functional block. The clock selection mux does not support glitch-free function, so the clock is unpredictable during clock selection.

Figure 11 shows the configurations of miscellaneous clocks.

Figure 11. Miscellaneous Clock Configuration



4.4.3 PCLK Clock Domain

The PCLK is the master clock for all peripherals except for the CRC generator and ports. It can shut down during power down modes. Each peripheral clock is generated by the SCU_PPCLKEN1 and SCU_PPCLKEN2 register set.

Figure 10 illustrates the PCLK clock distributions. Peripherals are not accessible until the PCLK clock of each block is enabled, and even by reading its registers.

4.4.4 Clock Configuration Procedure

When the device is powered on, a default system clock is generated by the HIRC (2 MHz) clock. The HIRC is enabled by default during the power up sequence. Other clock sources are enabled by user controls and configuration options with a system clock.

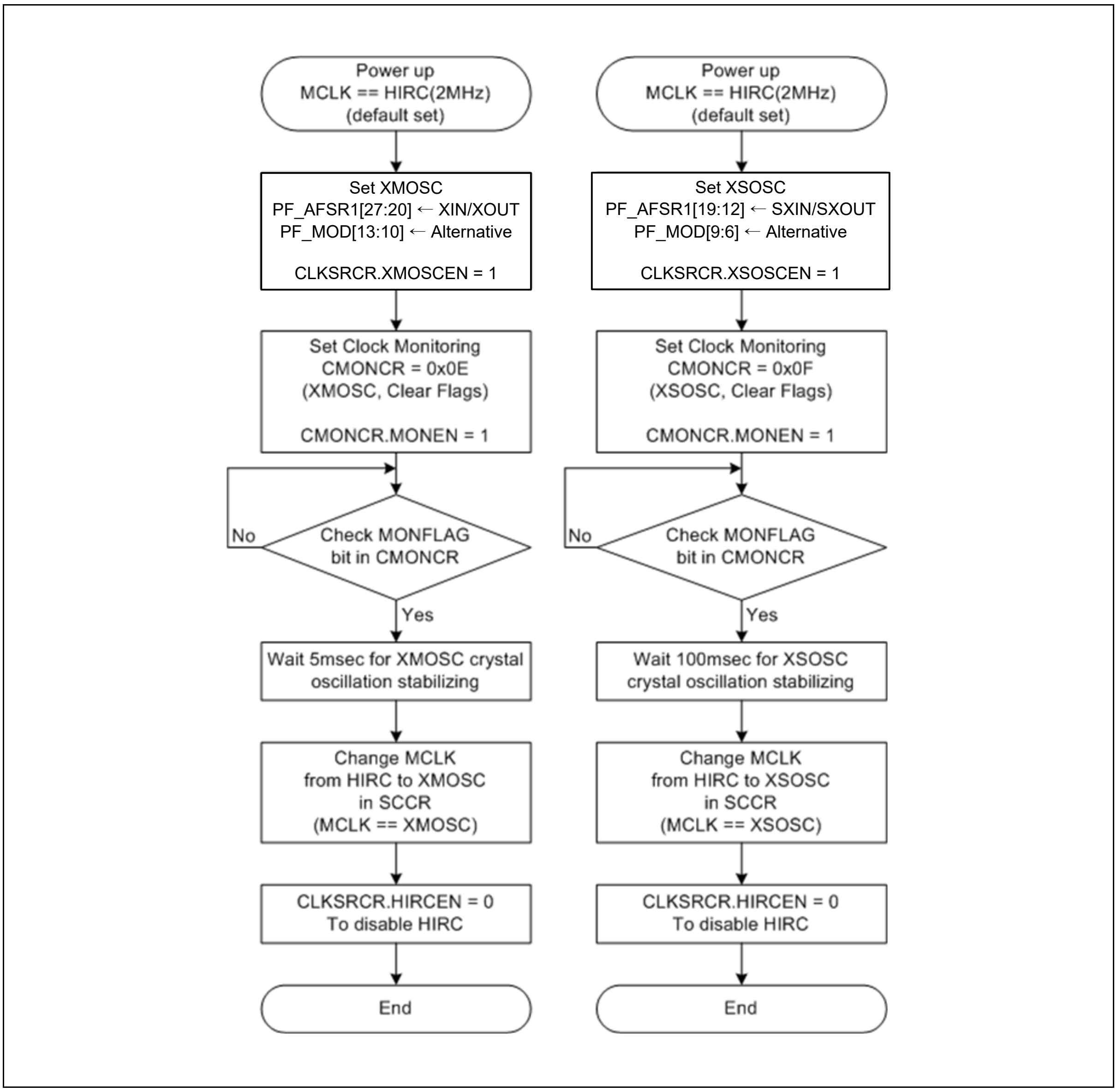
The XMOSC and XSOSC clocks are enabled by the XMOSCEN and XSOSCEN bits of the SCU_CLKSRCR register, respectively. Before enabling XMOSC and XSOSC blocks, the pin mux configuration should be set for XIN/XOUT and SXIN/SXOUT functions.

The PF5/PF6 and PF3/PF4 pins are shared by the XMOSC’s XIN/XOUT function and XSOSC’s SXIN/SXOUT function – The PF_MOD and PF_AFSR1 registers should be configured properly.

After enabling the XMOSC and XSOSC blocks, a user can check stability of crystal oscillation through the clock monitoring control register, SCU_CMONCR. It takes more than 1 ms to ensure stable crystal oscillation before changing the system clock.

Figure 12 shows an example of a flow chart configuring the system clock as the XMOSC and XSOSC clocks.

Figure 12. Clock Configuration Procedure



4.5 Reset

4.5.1 Overview

The A31S134 has two system resets. One is the cold reset by POR, which is effective during power up or down sequence, and the other is the warm reset, which is generated by several reset sources. The reset event makes the device return to its initial state.

The cold reset has a single reset source (POR), and the warm reset has several reset sources listed below:

- nRESET pin
- WDT reset
- LVR reset
- Clock monitoring reset
- Software reset
- CPU request reset

4.5.2 Cold Reset

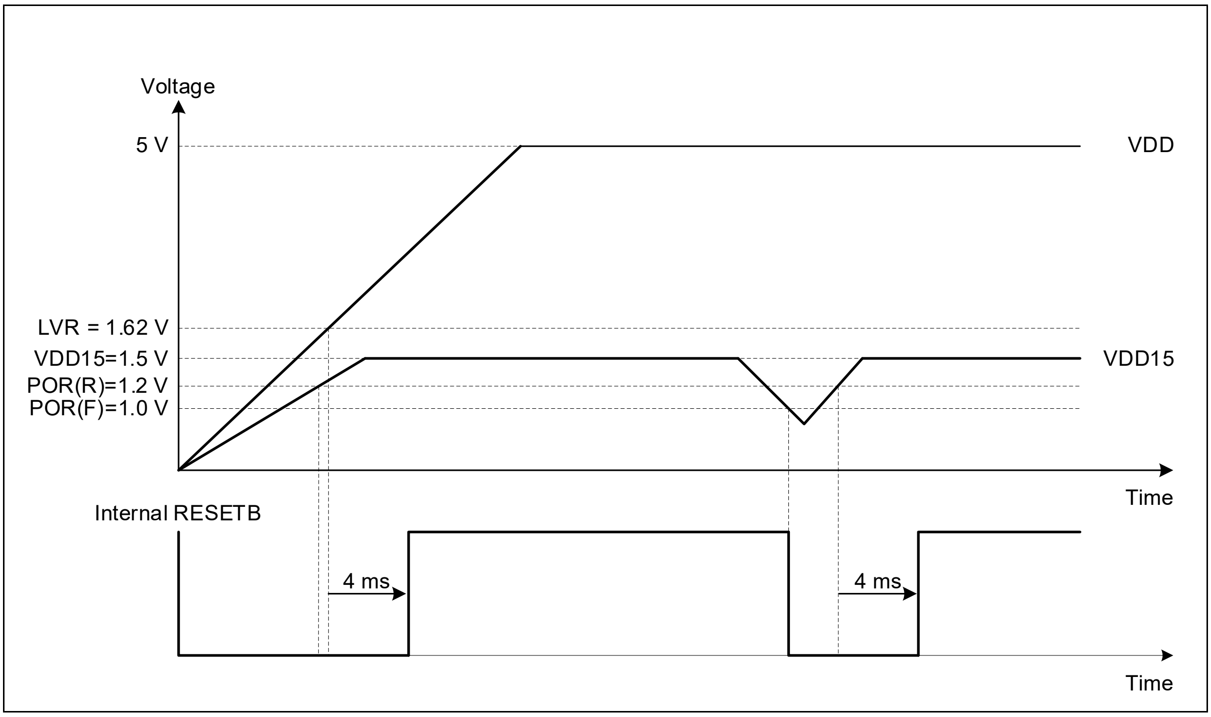
The cold reset is one of the important features of the A31S134 when powered on, which globally affects the system boot.

The internal VDC is enabled when the VDD power is supplied. The internal VDD level slope follows the external VDD power slope.

The internal POR trigger level is at 1.2 V of the internal VDC voltage. At this time, boot operation begins. The internal RC clock turns on and counts 4 ms for the internal VDC level to stabilize. At this time, the external VDD voltage level should be bigger than the initial LVR level (1.62 V). After 4ms of counting, the CPU reset is released, and operation begins.

Figure 13 shows waveform of the power up sequence and internal reset.

Figure 13. Power-up POR Sequence



The register SCU_RSTSSR shows the POR reset status. The last reset comes from the POR.

The PORSTA bit in the SCU_RSTSSR register is set to ‘1’. After power on, this bit is always set to ‘1’ unless it is cleared by software. If an abnormal internal voltage drop is detected during normal operation, the system is reset, and this bit is set to ‘1’.

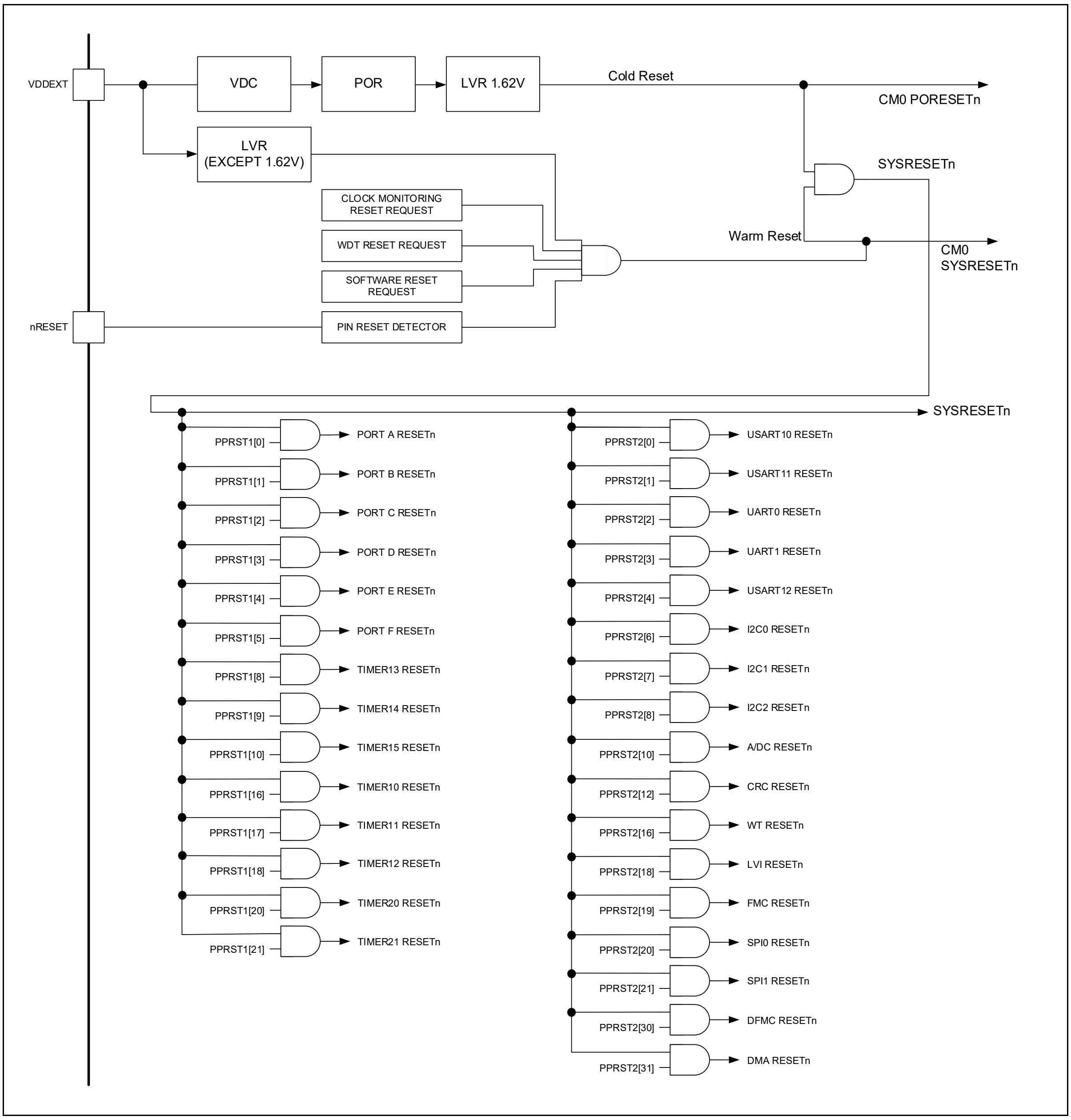
When the cold reset is applied, the entire device returns to its initial state.

4.5.3 Warm Reset

The warm reset event has several reset sources, which make the microcontroller and designated peripherals return to their initial states when the warm reset takes place.

The warm reset status appears in the SCU_RSTSSR register. A reset for each peripheral block is controlled by the register SCU_PPRST. The reset can be masked independently.

Figure 14. Reset Configuration

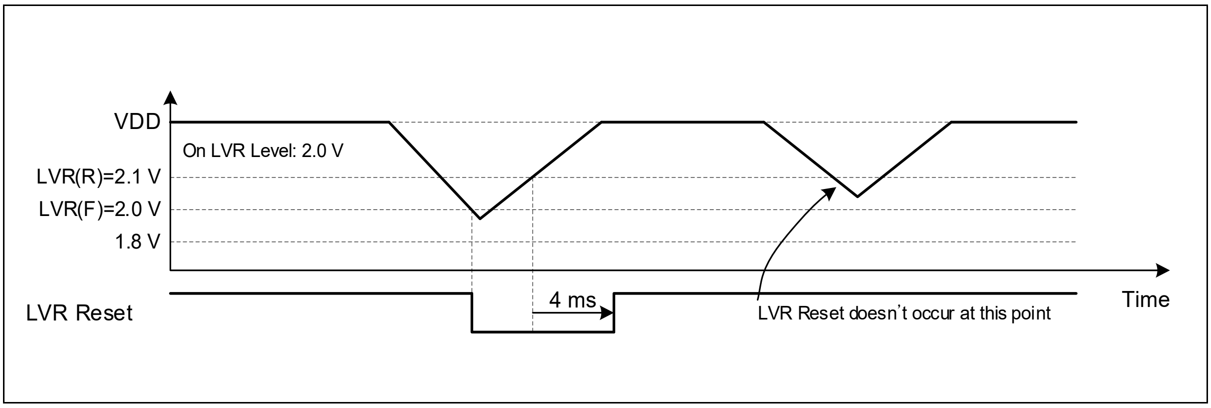


4.5.4 LVR Reset

The LVR voltage level is set by a low voltage reset configuration register (CONF_LVRCNFIG) in the configure option page 1. The LVR reset status appears in the register SCU_RSTSSR.

The LVR reset is controlled by the register SCU_LVRCR. This register is cleared to “0x00” when the POR reset occurs.

Figure 15. LVR Reset Timing Diagram

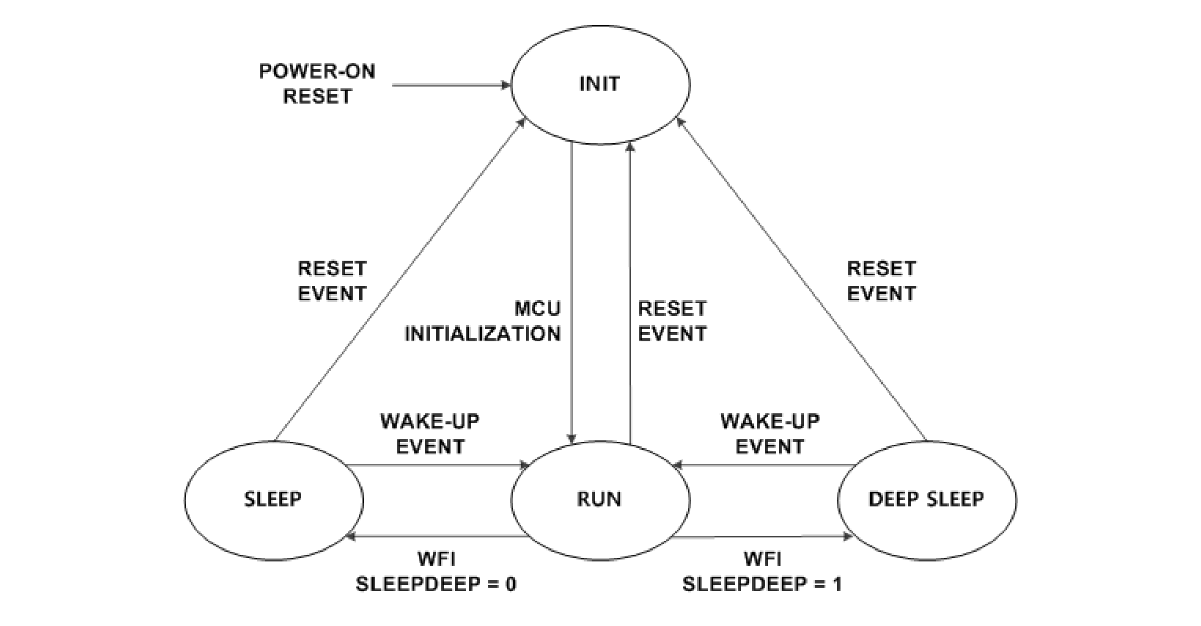


4.6 Operation Mode

INIT mode is the initial state of the device when resetting. At RUN mode, the chip runs at its maximum CPU performance with a high-speed clock system. At SLEEP and DEEP SLEEP mode, the chip runs at a low power consumption mode. The system saves power by halting the processor core and unused peripherals.

Figure 16 shows a diagram of the operation mode transition.

Figure 16. Operation Mode Transition



4.6.1 RUN Mode

This mode is to operate CPU and peripheral hardware with a high-speed clock. The device enters the INIT state after resetting, and then enters the RUN mode.

4.6.2 SLEEP Mode

In this mode, only the CPU is stopped. Each peripheral function is turned on by the function enable bit and the clock enable bit of the register SCU_PPCLKEN.

4.6.3 DEEP SLEEP Mode

In this mode, not only the CPU but also a selected system clock (MCLK) is stopped. The Watch timer with a sub clock and the Watchdog Timer with WDTRC still operate in this mode.

Table 7. Functional Table on Current Mode

IP	Main Run (IDD1)	Main Sleep (IDD2)	Sub Run (IDD3)	Sub Sleep (IDD4)	Deep Sleep (IDD5)
CPU	O	X	O	X	X
FLASH	O	X	O	X	X
SRAM	O	X	O	X	X
FMC/DFMC	Optional	X	Optional	X	X
DMA	Optional	Optional	Optional	Optional	X
CRC	Optional	X	Optional	X	X
POR	O	O	O	O	O
LVR/LVI	Optional	Optional	Optional	Optional	Optional
GPIO	Optional	Optional	Optional	Optional	Optional
SCU	O	O	O	O	O
SPI	Optional	Optional	Optional	Optional	X
I2C	Optional	Optional	Optional	Optional	Optional
USART	Optional	Optional	Optional	Optional	Optional
UART	Optional	Optional	Optional	Optional	X
SysTick	Optional	Optional	Optional	Optional	X
T10 – T15	Optional	Optional	Optional	Optional	X
T20	Optional	Optional	Optional	Optional	Optional
T21	Optional	Optional	Optional	Optional	X
WDT	Optional	Optional	Optional	Optional	Optional
WUT	O	O	O	O	X
A/DC	Optional	X	X	X	X
WT	Optional	Optional	Optional	Optional	Optional
HIRC	Optional	Optional	X	X	X
WDTRC	Optional	Optional	Optional	Optional	Optional
XMOSC	Optional	Optional	X	X	X
XSOSC	Optional	Optional	Optional	Optional	Optional

NOTES:

1. O: Enable, X: Disable, Optional: A function can be disabled/enabled by software.
2. It can be woken up from sleep and deep sleep modes by an interrupt source of the optional peripherals.

4.7 Pins for SCU

Table 8. Pins and External Signals for SCU

Pin Name	Type	Description
nRESET	I	External Reset Input
XIN/XOUT	OSC	External Crystal Oscillator for Main Clock
SXIN/SXOUT	OSC	External Crystal Oscillator for Sub Clock
CLKO	O	Clock Output Monitoring Signal

4.8 SCU Registers

The base address and register map of SCU (chip configuration) are described in the followings:

Table 9. Base Address of SCU (Chip Configuration)

Name	Base Address
SCU (chip configuration)	0x4000_F000

Table 10. SCU (Chip Configuration) Register Map

Name	Offset	Type	Description	Reset Value	Reference
SCU_VENDORID	0x0000	RO	Vendor Identification Register	0x4142_4F56	4.8.1
SCU_CHIPID	0x0004	RO	Chip Identification Register.	0x5331_134x	4.8.2
SCU_REVNR	0x0008	RO	Revision Number Register	0x0000_00xx	4.8.3
SCU_PMREMAP	0x0014	RW	Program Memory Remap Register	0x0000_0000	4.8.4
SCU_RSTSSR	0x001C	RW	Reset Source Status Register	0x0000_00xx	4.8.5
SCU_NMISRCR	0x0020	RW	NMI Source Selection Register	0x0000_0000	4.8.6
SCU_SWRSTR	0x0024	WO	Software Reset Register	0x0000_0000	4.8.7
SCU_SRSTVR	0x0028	RO	System Reset Validation Register	0x0000_0055	4.8.8
SCU_WUTCR	0x002C	RW	Wake-up Timer Control Register	0x0000_0000	4.8.9
SCU_WUTDR	0x0030	RW	Wake-up Timer Data Register	0x0000_00FA	4.8.10
SCU_HIRCTRM	0x00A8	RW	High Frequency Internal RC Trim Register (HIRCNFIG)	0x0000_0xxx	4.8.11
SCU_WDTRCTRM	0x00AC	RW	Watchdog Timer RC Trim Register (WDTRCNFIG)	0x0000_00xx	4.8.12

NOTE: The CHIPID is written by hardware if the proper configure address is read.

The base address and register map of the SCU (clock generation) are described in the followings:

Table 11. Base Address of SCU (Clock Generation)

Name	Base Address
SCU (clock generation)	0x4000_1800

Table 12. SCU Register Map (Clock Generation)

Name	Offset	Type	Description	Reset Value	Ref.
SCU_SCCR	0x0000	RW	System Clock Control Register	0x0000_0000	4.8.13
SCU_CLKSRCR	0x0004	RW	Clock Source Control Register	0x0000_000C	4.8.14
SCU_SCDIVR1	0x0008	RW	System Clock Divide Register 1	0x0000_0000	4.8.15
SCU_SCDIVR2	0x000C	RW	System Clock Divide Register 2	0x0000_0000	4.8.16
SCU_CLKOCR	0x0010	RW	Clock Output Control Register	0x0000_0000	4.8.17
SCU_CMONCR	0x0014	RW	Clock Monitoring Control Register	0x0000_0000	4.8.18
SCU_PPCLKEN1	0x0020	RW	Peripheral Clock Enable Register 1	0x0000_0000	4.8.19
SCU_PPCLKEN2	0x0024	RW	Peripheral Clock Enable Register 2	0x0002_0000	4.8.20
SCU_PPCLKSR	0x0040	RW	Peripheral Clock Selection Register	0x0000_0000	4.8.21
SCU_PPRST1	0x0060	RW	Peripheral Reset Register 1	0x0000_0000	4.8.22
SCU_PPRST2	0x0064	RW	Peripheral Reset Register 2	0x0000_0000	4.8.23
SCU_XTFLSR	0x0080	RW	X-tal Filter Selection Register	0x0000_0005	4.8.24

The base address and register map of the SCU (LVR/LVI) are described in the followings:

Table 13. Base Address of SCU (LVR/LVI)

Name	Base address
SCU (LVR/LVI)	0x4000_5100

Table 14. SCU Register Map (LVR/LVI)

Name	Offset	Type	Description	Reset Value	Reference
SCU_LVICR	0x0000	RW	Low Voltage Indicator Control Register	0x0000_0000	4.8.25
SCU_ LVRCR	0x0004	RW	Low Voltage Reset Control Register	0x0000_0000	4.8.26

4.8.1 SCU_VENDORID: Vendor ID Register

The SCU_VENDORID register shows the vendor identification information. This is a 32-bit read-only register.

SCU_VENDORID=0x4000_F000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
VENDID[31:0]			
0x4142_4F56			
RO			

31 0	VENDID[31:0]	Vendor Identification bits. 0x4142_4F56

4.8.2 SCU_CHIPID: Chip ID Register

The SCU_CHIPID register shows chip identification information. This is a 32-bit read-only register.

SCU_CHIPID=0x4000_F004

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CHIPID[31:0]			
0x5331_1340			
RO			

31 0	CHIPID[31:0]	Chip Identification bits.	
		0x5331_1340	128 KB flash memory for program

4.8.3 SCU_REVNR: Revision Number Register

The SCU_REVNR register is a 32-bit read-only register. This register is available at 32-/ 16-/ 8-bit access.

SCU_REVNR=0x4000_F008

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			REVNO[7:0]
0x000000			xx
-			RO

7 0	REVNO[7:0]	Chip Revision Number. These bits are fixed by manufacturer.

4.8.4 SCU_PMREMAP: Program Memory Remap Register

The SCU_PMREMAP register is a 32-bit write-only register.

SCU_PMREMAP=0x4000_F014

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
WTIDKY[15:0]		nPMREM[7:0]	PMREM[7:0]
0x0000		0x00	0x00
WO		WO	WO

31 16	WTIDKY[15:0]	Write Identification Key When writing, write 0xE2F1 to these bits, or else writing is ignored.	
15 8	nPMREM[7:0]	Write Complement Key When writing, write the complement value of PMREM[7:0], or else writing is ignored.	
7 0	PMREM[7:0]	Program Memory Remap	
		0x69	Boot ROM is re-mapped to address 0x00000000. 0x10001000 of flash memory is re-mapped to address 0x00001000.
		Others	Flash memory is re-mapped to address 0x00000000.
		NOTE: 1. The remapped program memory can be accessed from the original address.	

4.8.5 SCU_RSTSSR: Reset Source Status Register

The SCU_RSTSSR register shows reset source information when reset event occurs. '1' implies a reset event exists, while '0' means a reset event does not exist for a corresponding reset source.

When a reset source is detected, '1' is written into the corresponding bit position and reset status will be cleared.

The SCU_RSTSSR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_RSTSSR=0x4000_F01C

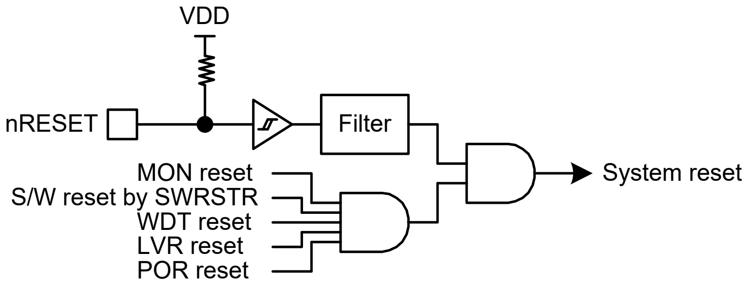
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 1211 10 9 8	7	6	5	4	3	2	1	0
Reserved					MONSTA	SWSTA	EXTSTA	WDTSTA	LVRSTA	PORSTA
-					x	x	x	x	x	x
.					RW	RW	RW	RW	RW	RW

5	MONSTA	Clock Monitoring Reset Status	
		0	Not detected
		1	Clock monitoring reset is detected. The bit is cleared to '0' when '1' is written.
4	SWSTA	Software Reset Status.	
		0	Not detected
		1	Software reset is detected. The bit is cleared to '0' when '1' is written.
3	EXTSTA	External Pin Reset Status.	
		0	Not detected
		1	External pin reset is detected. The bit is cleared to '0' when '1' is written.
2	WDTSTA	Watchdog Timer Reset Status.	
		0	Not detected
		1	Watchdog Timer reset is detected. The bit is cleared to '0' when '1' is written.
1	LVRSTA	LVR Reset Status.	
		0	Not detected
		1	LVR reset is detected. The bit is cleared to '0' when '1' is written.
0	PORSTA	POR Reset Status.	
		0	Not detected
		1	POR reset is detected. The bit is cleared to '0' when '1' is written.

NOTES:

1. The PORSTA bit is set to '1' and the other bits are cleared to '0' when power-on reset occurs.
2. The corresponding reset status bit may be set to '1' if any reset signal is asserted during power-on reset occurs. For example, The EXTSTA bit may be set if the external reset is asserted during POR.

Figure 17. Reset Circuit Diagram



NOTE: A core software reset will reset only CPU.

4.8.6 SCU_NMISRCR: NMI Source Selection Register

The SCU_NMISRCR register is the non-maskable interrupt configuration register, which can be set by software. The SCU_NMISRCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_NMISRCR=0x4000_F020

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7	6	5	4 3 2 1 0
Reserved	NMICON	MONINT	Reserved	NMISRC[4:0]
-	0	0	-	00000
-	RW	RW	-	RW

7	NMICON	Non-Maskable Interrupt (NMI) Control.	
		0	Disable NMI
		1	Enable NMI
6	MONINT	Clock Monitoring Interrupt Selection.	
		0	Non-select clock monitoring interrupt for NMI source
		1	Select clock monitoring interrupt for NMI source
4 0	NMISRC[4:0]	Non-Maskable Interrupt Source Selection.	
		Select one of the interrupt sources 0 to 31 for NMI source.	

NOTE:

1. The interrupt source which is selected for NMI should be disabled in NVIC to avoid both generation of the normal and NMI interrupts.

4.8.7 SCU_SWRSTR: Software Reset Register

The SCU_SWRSTR register is a 32-bit size.

SCU_SWRSTR =0x4000_F024

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
WTIDKY[15:0]	Reserved	SWRST[7:0]
0x0000	-	0x00
WO	-	WO

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0x9EB3 to these bits, or else writing is ignored.	
7 0	SWRST[7:0]	Software Reset (System Reset)	
		0x2D	A software reset will be generated for all peripheral and core.
		Others	No effect

4.8.8 SCU_SRSTVR: System Reset Validation Register

The SCU_SRSTVR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_SRSTVR=0x4000_F028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								VALID[7:0]							
-																								0x55							
-																								RO							

7 0	VALID[7:0]	System Reset Validation	
		0x55	System reset is OK.
		Others	A weak system reset. A system reset must be generated by software.

4.8.9 SCU_WUTCR: Wake-up Timer Control Register

Wake-up timer always works on operating mode. This timer gives a stable time for clock generation during Power on and DEEP SLEEP mode release. The main purpose of this timer is periodical tick timer or a wake-up source.

The SCU_WUTCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_WUTCR=0x4000_F02C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7	6 5 4 3 2	1	0
Reserved	WUTIEN	Reserved	CNTRLD	WUTIFLAG
-	0	-	0	0
-	RW	-	RW	RW

7	WUTIEN	Wake-up Timer Interrupt Enable bit	
		0	Disable wake-up timer interrupt
		1	Enable wake-up timer interrupt
1	CNTRLD	Counter Reload bit	
		0	No effect
		1	Reload data to counter (automatically cleared to '0' after operation)
0	WUTIFLAG	Wake-up Timer Interrupt Flag bit	
		0	No request occurred
		1	Request occurred. The bit is cleared to '0' when '1' is written.
		NOTE: 1. This bit may not be set to '1' if the PCLK frequency is slower than the HCLK frequency. So, for WUT interrupt to occur normally, the SCU_ SCDIVR2.PDIV[1:0] bits must be set to "00" so that the PCLK frequency is the same as the HCLK frequency.	

4.8.10 SCU_WUTDR: Wake-up Timer Data Register

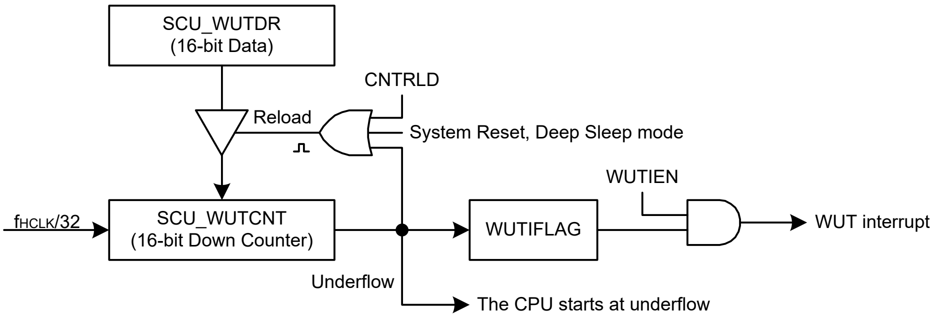
The SCU_WUTDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_WUTDR=0x4000_F030

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		WUTDATA[15:0]	
-		0x00FA	
-		RW	

15 0	WUTDATA[15:0]	Wake-up Timer Data. The range is 0x0001 to 0xFFFF. **NOTE:** 1. Its value should be set to be at least more than 150 µs

Figure 18. Wake-up Timer Circuit Diagram



NOTE:

1. The WUTIFLAG bit may not be set to ‘1’ if the PCLK frequency is slower than the HCLK frequency. So, for WUT interrupt to occur normally, the SCU_SCDIVR2.PDIV[1:0] bits must be set to "00" so that the PCLK frequency is the same as the HCLK frequency.

4.8.11 SCU_HIRCTRM: High Frequency Internal RC Trim Register

The SCU_HIRCTRM register may be used for user trimming of HIRC by software. This register is a 32-bit size.

SCU_HIRCTRM=0x4000_F0A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WTIDKY[15:0]																Reserved							CTRMH[2:0]			FTRMH[5:0]					
0x0000																-							x	x	x	x	x	x	x	x	x
WO																-							RW	RW	RW	RW	RW	RW	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key When writing, write 0xA6B5 to these bits, or else writing is ignored.
8 6	CTRMH[2:0]	Factory HIRC Coarse Trim. These bits are fixed by the manufacturer and read from "Configure Option Page 0" when a system reset occurs. These bits provide a user programmable trimming value on operation. The range is -4 to +3, the CTRMH[2] is sign bit, and the frequency is changed by 1.4MHz step-by-step.
5 0	FTRMH[5:0]	Factory HIRC Fine Trim. These bits are fixed by the manufacturer and read from "Configure Option Page 0" when a system reset occurs. These bits provide a user programmable trimming value on operation. The range is -32 to +31, the FTRMH[5] is sign bit, and the frequency is changed by 70 kHz step-by-step.

4.8.12 SCU_WDTRCTRM: Watchdog Timer RC Trim Register

The SCU_WDTRCTRM register may be used for user trimming of WDTRC by software. This register is a 32-bit size.

SCU_WDTRCTRM=0x4000_F0AC

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
WTIDKY[15:0]		nTRMW[7:0]	CTRMW[3:0]				Reserved	FTRMW[2:0]		
0x0000		xx	x	x	x	x	-	x	x	x
WO		WO	RW	RW	RW	RW	-	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key
		When writing, write 0x4C3D to these bits, or else writing is ignored.
15 8	nTRMW[7:0]	Write Complement Key
		On writes, write the complement value of TRMW[7:0], otherwise the write is ignored.
7 4	CTRMW[3:0]	Factory WDTRC Coarse Trim.
		These bits are fixed by the manufacturer and read from “Configure Option Page 0” when a system reset occurs. These bits provide a user-programmable trimming value on operation. The range is -8 to +7, the CTRMW[3] is sign bit, and the frequency is changed by 4kHz step-by-step.
2 0	FTRMW[2:0]	Factory WDTRC Fine Trim.
		These bits are fixed by the manufacturer and read from “Configure Option Page 0” when a system reset occurs. These bits provide a user-programmable trimming value on operation. The range is -4 to +3, the FTRMW[2] is sign bit, and the frequency is changed by 1.1 kHz step-by-step.

4.8.13 SCU_SCCR: System Clock Control Register

The A31S134 has multiple clock sources to generate internal operating clocks. SCU_SCCR register controls such a clock source. This register is a 32-bit size.

SCU_SCCR=0x4000_1800

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2	1	0
WTIDKY[15:0]	Reserved	MCLKSEL[1:0]	
0x0000	-	0	0
WO	-	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0x570A to these bits, or else writing is ignored.	
1 0	MCLKSEL[1:0]	Main Clock Selection, MCLK	
		00	High frequency Internal RC oscillator (32 MHz), HIRC
		01	External main oscillator (2 to 32 MHz), XMOSC
		10	External sub oscillator (32.768 kHz), XSOSC
		11	Internal watchdog timer RC oscillator (40 kHz), WDTRC

NOTE:

1. If the MCLKSEL bits are “10” or “11”, the HDIV[2:0] bits of SCU_SCDIVR1 register should be “100” for non-divided system clock.

4.8.14 SCU_CLKSRCR: Clock Source Control Register

The A31S134 has multiple clock sources to generate internal operating clocks. SCU_CLKSRCR register controls each clock source. This register is a 32-bit size.

SCU_CLKSRCR=0x4000_1804

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WTIDKY[15:0]	Reserved		HIRCSEL[1:0]		Reserved			XMFRNG	Reserved				WDTRCEN	HIRCEN	XMOSCEN	XSOSCEN
0x0000	-	-	0	0	-	-	-	0	-	-	-	-	1	1	0	0
WO	-	-	RW	RW	-	-	-	RW	-	-	-	-	RW	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0xA507 to these bits, or else writing is ignored.	
13 12	HIRCSEL[1:0]	HIRC Frequency Selection bits	
		00	32 MHz HIRC
		01	16 MHz HIRC
		10	8 MHz HIRC
		11	4 MHz HIRC
8	XMFRNG	Main Oscillator Type and Frequency Range Selection bit	
		0	X-tal for XMOSC, 2 to 16MHz
		1	External clock for XMOSC, 2MHz to 32MHz
3	WDTRCEN	WDTRC Enable bit, Watchdog Timer RC oscillator	
		0	Disable WDTRC
		1	Enable WDTRC
2	HIRCEN	HIRC Enable bit, High frequency internal RC oscillator	
		0	Disable HIRC
		1	Enable HIRC
1	XMOSCEN	XMOSC Enable bit, External main oscillator	
		0	Disable XMOSC
		1	Enable XMOSC
0	XSOSCEN	XSOSC Enable bit, External sub oscillator.	
		0	Disable XSOSC
		1	Enable XSOSC

4.8.15 SCU_SCDIVR1: System Clock Divide Register 1

The SCU_SCDIVR1 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_SCDIVR1=0x4000_1808

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved				WLDIV[2:0]			Reserved	HDIV[2:0]		
-				0	0	0	-	0	0	0
-				RW	RW	RW	-	RW	RW	RW

6 4	WLDIV[2:0]	Clock Divide bits for Watch Timer, Divider 2	
		000	MCLK÷64
		001	MCLK÷128
		010	MCLK÷256
		011	MCLK÷512
		100	MCLK÷1024
		others	Reserved
2 0	HDIV[2:0]	Clock Divide bits for HCLK, Divider 0	
		000	MCLK÷16
		001	MCLK÷8
		010	MCLK÷4
		011	MCLK÷2
		100	MCLK÷1
		others	Reserved (MCLK÷1)

NOTES:

1. If the selected MCLK is XSOSC or WDTRC, the HDIV[2:0] bits should be set to “100”.
2. The frequency range of HCLK should be 2.0 to 32 MHz by software while the HIRC is the system clock.
3. After changing the value of HDIV[2:0] bits for system clock speed, a delay 10 μs is required before changing the value again immediately.

4.8.16 SCU_SCDIVR2: System Clock Divide Register 2

The SCU_SCDIVR2 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_SCDIVR2=0x4000_180C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6	5	4	3	2	1	0
Reserved	SYSTDIV[1:0]		Reserved		PDIV[1:0]	
-	0	0	-	-	0	0
-	RW	RW	-	-	RW	RW

5 4	SYSTDIV[1:0]	Clock Divide bits for SysTick Timer, Divider 3	
		00	HCLK÷1
		01	HCLK÷2
		10	HCLK÷4
		11	HCLK÷8
1 0	PDIV[1:0]	Clock Divide bits for PCLK, Divider 1	
		00	HCLK÷1
		01	HCLK÷2
		10	HCLK÷4
		11	HCLK÷8

NOTE:

1. If the selected MCLK is XSOSC or WDTRC, the PDIV[1:0] should be set to “00”.

4.8.17 SCU_CLKOCR: Clock Output Control Register

The A31S134 can drive the clock from a selected clock (CLKOS) with a dedicated post divider.

The SCU_CLKOCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_CLKOCR=0x4000_1810

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			CLKOEN	POLSEL	CLKODIV[2:0]			CLKOS[2:0]		
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

7	CLKOEN	Clock Output Enable bit	
		0	Disable clock output
		1	Enable clock output
6	POLSEL	Clock Output Polarity Selection bit when disable	
		0	Low level during disabled
		1	High level during disabled
5 3	CLKODIV[2:0]	Output Clock Divide bits, Divider 4	
		000	“Selected clock”÷1
		001	“Selected clock”÷2
		010	“Selected clock”÷4
		011	“Selected clock”÷8
		100	“Selected clock”÷16
		101	“Selected clock”÷32
		110	“Selected clock”÷64
		111	“Selected clock”÷128
2 0	CLKOS[2:0]	Clock Output Selection bits	
		000	MCLK
		001	WDTRC
		010	HIRC
		011	HCLK
		100	PCLK
		others	Reserved (None)

4.8.18 SCU_CMONCR: Clock Monitoring Control Register

Internal clocks can be monitored by using internal WDTRC for security purpose.

The SCU_CMONCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_CMONCR=0x4000_1814

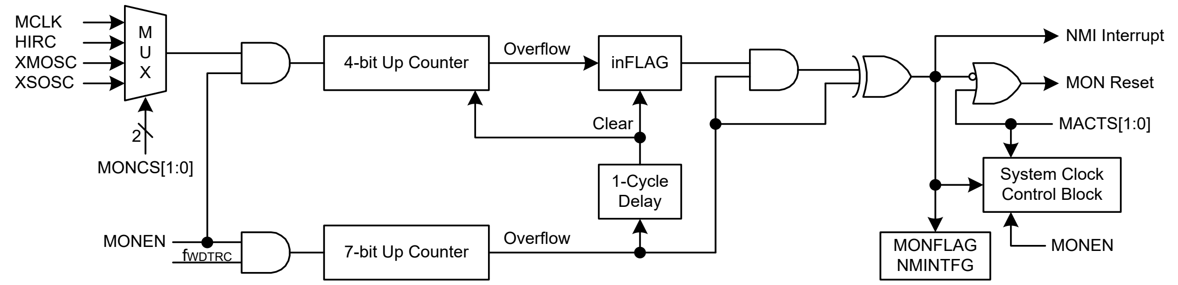
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			MONEN	MACTS[1:0]		Reserved	MONFLAG	NMINTFG	MONCS[1:0]	
-			0	0	0	-	0	0	0	0
-			RW	RW	RW	-	RW	RW	RW	RW

7	MONEN	Clock Monitoring Enable bit	
		0	Disable clock monitoring
		1	Enable clock monitoring
		NOTE: 1. When this bit is reset to ‘0’, the block clears the 4/7-bit counter, inFLAG, and flags.	
6 5	MACTS[1:0]	Clock Monitoring Action Selection bits	
		00	No action by clock monitoring, but flags will be set or cleared on condition
		01	Reset generation by clock monitoring
		10	The system clock will be changed to the WDTRC regardless of MCLKSEL[1:0] bits of system clock control register (SCU_SCCR) only when the MCLK is selected for monitoring.
		11	Not used
3	MONFLAG	Clock Monitoring Result Flag bit	
		0	The clock under monitoring is not ready.
		1	The clock under monitoring is ready. This bit is cleared to ‘0’ when ‘1’ is written.
2	NMINTFG	Clock Monitoring Interrupt Flag bit (only when the MCLK is selected for monitoring)	
		0	No request occurred
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
		NOTE: When the bit is set, the system clock must be switched to WDTRC by software.	
1 0	MONCS[1:0]	Monitored Clock Selection bits	
		00	MCLK
		01	HIRC
		10	XMOSC
		11	XSOSC

NOTES:

1. The block should be enabled after disabled to clear the internal status for new clock monitoring.
2. This block must be disabled by software before entering DEEP SLEEP mode.

Figure 19. Clock Monitoring Circuit Diagram



NOTE:

1. WDTRC should be enabled to use this block since it is the watch clock for monitoring.

4.8.19 SCU_PPCLKEN1: Peripheral Clock Enable Register 1

To use a certain peripheral unit, its clock should be activated by writing ‘1’ to the corresponding bit in the SCU_PPCLKEN1 and SCU_PPCLKEN2 register. Until enabling the clock, the peripheral does not operate properly. To stop the clock of the peripheral unit, write ‘0’ to the corresponding bit in the SCU_PPCLKEN1 and SCU_PPCLKEN2 register.

The SCU_PPCLKEN1 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPCLKEN1=0x4000_1820

31 30 29 28 27 26 25 24 23 22	21	20	19	18	17	16	15 14 13 12 11	10	9	8	7 6	5	4	3	2	1	0
Reserved	T21CLKE	T20CLKE	Reserved	T12CLKE	T11CLKE	T10CLKE	Reserved	T15CLKE	T14CLKE	T13CLKE	Reserved	PFCLKE	PECLKE	PDCLKE	PCCLKE	PBCLKE	PACLKE
-	0	0	-	0	0	0	-	0	0	0	-	0	0	0	0	0	0
-	RW	RW	-	RW	RW	RW	-	RW	RW	RW	-	RW	RW	RW	RW	RW	RW

21	T21CLKE	TIMER21 Clock Enable
20	T20CLKE	TIMER20 Clock Enable
18	T12CLKE	TIMER12 Clock Enable
17	T11CLKE	TIMER11 Clock Enable
16	T10CLKE	TIMER10 Clock Enable
10	T15CLKE	TIMER15 Clock Enable
9	T14CLKE	TIMER14 Clock Enable
8	T13CLKE	TIMER13 Clock Enable
5	PFCLKE	Port F Clock Enable
4	PECLKE	Port E Clock Enable
3	PDCLKE	Port D Clock Enable
2	PCCLKE	Port C Clock Enable
1	PBCLKE	Port B Clock Enable
0	PACLKE	Port A Clock Enable

NOTE:

1. The peripheral registers may not be read or written by software when the peripheral clock is disabled.

4.8.20 SCU_PPCLKEN2: Peripheral Clock Enable Register 2

The SCU_PPCLKEN2 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPCLKEN2=0x4000_1824

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMACLKE	DFMCLKE	Reserved								SPI1CLKE	SPI0CLKE	FMCLKE	LVICLKE	WDTCLKE	WTCLKE	Reserved			CRCLKE	Reserved	ADCLKE	Reserved	I2C2CLKE	I2C1CLKE	I2C0CLKE	Reserved	UST12CLKE	UT1CLKE	UT0CLKE	UST11CLKE	UST10CLKE
0	0	-								0	0	0	0	1	0	-			0	-	0	-	0	0	0	-	0	0	0	0	0
RW	RW	-								RW	RW	RW	RW	RW	RW	-			RW	-	RW	-	RW	RW	RW	-	RW	RW	RW	RW	RW

31	DMACLKE	DMA Clock Enable
30	DFMCLKE	Data Flash Memory Control Clock Enable. Ignored during Data Flash operation.
21	SPI1CLKE	SPI 1 Clock Enable
20	SPI0CLKE	SPI 0 Clock Enable
19	FMCLKE	Flash Memory Control Clock Enable. Ignored during Flash operation.
18	LVICLKE	LVI (Low-Voltage Indicator) Clock Enable
17	WDTCLKE	WDT (Watchdog Timer) Clock Enable. The WDTRC won't be disabled if the clock is enabled by watchdog timer configuration register (CONF_WDTCNFIG) in “Configure Option Page 1”
16	WTCLKE	WT (Watch Timer) Clock Enable
12	CRCLKE	CRC (Cyclic Redundancy Check) Clock Enable
10	ADCLKE	ADC (Analog to Digital Converter) Clock Enable
8	I2C2CLKE	I2C2 (Inter-integrated Circuit) Clock Enable
7	I2C1CLKE	I2C1 (Inter-integrated Circuit) Clock Enable
6	I2C0CLKE	I2C0 (Inter-integrated Circuit) Clock Enable
4	UST12CLKE	USART12 Clock Enable
3	UT1CLKE	UART1 Clock Enable
2	UT0CLKE	UART0 Clock Enable
1	UST11CLKE	USART11 Clock Enable
0	UST10CLKE	USART10 Clock Enable

NOTE:

1. The peripheral registers may not be read/written by software when the peripheral clock is disabled.

4.8.21 SCU_PPCLKSR: Peripheral Clock Selection Register

The SCU_PPCLKSR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPCLKSR=0x4000_1840

31 30 29 28 27 26 25 24 23 22 21	20	19 18 17 16 15 14 13 12 11 10 9 8 7 6 5	4	3	2 1	0
Reserved	T20CLK	Reserved	WTCLK[1:0]		Reserved	WDTCLK
-	0	-	0	0	-	0
-	RW	-	RW	RW	-	RW

20	T20CLK	Timer 20 Clock Selection.	
		0	XSOSC clock
		1	PCLK clock
4 3	WTCLK[1:0]	Watch Timer Clock Selection.	
		00	A clock of the MCLK which is divided by divider 2
		01	XSOSC clock
		10	WDTRC clock
		11	Reserved
		NOTE: 1. These bits should be changed during the WTEN bit of watch timer control register (WT_CR) is ‘0’.	
0	WDTCLK	Watchdog Timer Clock Selection.	
		0	WDTRC clock
		1	PCLK clock

4.8.22 SCU_PPRST1: Peripheral Reset Register 1

The SCU_PPRST1 and SCU_PPRST2 registers can make peripheral resets. If a specific bit in this register is set to ‘1’, the peripheral corresponding with this bit occurs a reset event and the registers of the peripheral are initialized with reset values.

The SCU_PPRST1 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPRST1=0x4000_1860

31 30 29 28 27 26 25 24 23 22	21	20	19	18	17	16	15 14 13 12 11	10	9	8	7 6	5	4	3	2	1	0
Reserved	T21RST	T20RST	Reserved	T12RST	T11RST	T10RST	Reserved	T15RST	T14RST	T13RST	Reserved	PFRST	PERST	PDRST	PCRST	PBRST	PARST
-	0	0	-	0	0	0	-	0	0	0	-	0	0	0	0	0	0
-	RW	RW	-	RW	RW	RW	-	RW	RW	RW	-	RW	RW	RW	RW	RW	RW

21	T21RST	Timer 21 Reset bit	
		0	No effect
		1	Reset Timer 21, Cleared by software
20	T20RST	Timer 20 Reset bit	
		0	No effect
		1	Reset Timer 20, Cleared by software
18	T12RST	Timer 12 Reset bit	
		0	No effect
		1	Reset Timer 12, Cleared by software
17	T11RST	Timer 11 Reset bit	
		0	No effect
		1	Reset Timer 11, Cleared by software
16	T10RST	Timer 10 Reset bit	
		0	No effect
		1	Reset Timer 10, Cleared by software
10	T15RST	Timer 15 Reset bit	
		0	No effect
		1	Reset Timer 15, Cleared by software
9	T14RST	Timer 14 Reset bit	
		0	No effect
		1	Reset Timer 14, Cleared by software
8	T13RST	Timer 13 Reset bit	
		0	No effect
		1	Reset Timer 13, Cleared by software
5	PFRST	Port F Reset bit	
		0	No effect
		1	Reset Port F, Cleared by software
4	PERST	Port E Reset bit	
		0	No effect
		1	Reset Port E, Cleared by software
3	PDRST	Port D Reset bit	
		0	No effect
		1	Reset Port D, Cleared by software

2	PCRST	Port C Reset bit	
		0	No effect
		1	Reset Port C, Cleared by software
1	PBRST	Port B Reset bit	
		0	No effect
		1	Reset Port B, Cleared by software
0	PARST	Port A Reset bit	
		0	No effect
		1	Reset Port A, Cleared by software

4.8.23 SCU_PPRST2: Peripheral Reset Register 2

The SCU_PPRST2 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPRST2=0x4000_1864

31	30	29 28 27 26 25 24 23 22	21	20	19	18	17	16	15 14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMARST	DFMCRST	Reserved	SPI1RST	SPI0RST	FMCRST	LVIRST	Reserved	WTRST	Reserved	CRRST	Reserved	ADRST	Reserved	I2C2RST	I2C1RST	I2C0RST	Reserved	UST12RST	UT1RST	UT0RST	UST11RST	UST10RST
0	0	-	0	0	0	0	-	0	-	0	-	0	-	0	0	0	-	0	0	0	0	0
RW	RW	-	RW	RW	RW	RW	-	RW	-	RW	-	RW	-	RW	RW	RW	-	RW	RW	RW	RW	RW

31	DMARST	DMA Reset bit	
		0	No effect
		1	Reset DMA, Cleared by software
30	DFMCRST	Data Flash Memory Control Reset bit. Ignored during Data Flash operation.	
		0	No effect
		1	Reset Data Flash memory control, Cleared by software.
21	SPI1RST	SPI1 Reset bit	
		0	No effect
		1	Reset SPI1, Cleared by software
20	SPI0RST	SPI0 Reset bit	
		0	No effect
		1	Reset SPI0, Cleared by software
19	FMCRST	Flash Memory Control (FMC) Reset bit. Ignored during Flash operation.	
		0	No effect
		1	Reset Flash memory control, Cleared by software.
18	LVIRST	Low Voltage Indicator (LVI) Reset bit	
		0	No effect
		1	Reset LVI, Cleared by software
16	WTRST	Watch Timer (WT) Reset bit	
		0	No effect
		1	Reset WT, Cleared by software
12	CRRST	Cyclic Redundancy Check (CRC) Reset bit	
		0	No effect
		1	Reset CRC, Cleared by software
10	ADRST	Analog to Digital Converter (ADC) Reset bit	
		0	No effect
		1	Reset ADC, Cleared by software
8	I2C2RST	Inter-integrated Circuit (I2C2) Reset bit	
		0	No effect
		1	Reset I2C2, Cleared by software
7	I2C1RST	Inter-integrated Circuit (I2C1) Reset bit	
		0	No effect
		1	Reset I2C1, Cleared by software
6	I2C0RST	Inter-integrated Circuit (I2C0) Reset bit	
		0	No effect
		1	Reset I2C0, Cleared by software
4	UST12RST	USART12 Reset bit	
		0	No effect

		1	Reset USART12. Cleared by software
3	UT1RST	UART1 Reset bit	
		0	No effect
		1	Reset UART1. Cleared by software
2	UT0RST	UART0 Reset bit	
		0	No effect
		1	Reset UART0. Cleared by software
1	UST11RST	USART11 Reset bit	
		0	No effect
		1	Reset USART11. Cleared by software
0	UST10RST	USART10 Reset bit	
		0	No effect
		1	Reset USART10. Cleared by software

4.8.24 SCU_XTFLSR: Crystal Filter Selection Register

SCU_XTFLSR register is used to improve noise immunity of the main crystal oscillator. This register should be set to a proper value for corresponding crystal oscillator frequency.

The SCU_XTFLSR register is a 32-bit size.

SCU_XTFLSR =0x4000_1880

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3	2	1	0
WTIDKY[15:0]	Reserved	XRNS[2:0]		
0x0000	-	1	0	1
WO	.	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0x9B37 to these bits, or else writing is ignored	
2 0	XRNS[2:0]	External Main Oscillator Filter Selection.	
		000	Crystal ≤ 4.5MHz
		001	4.5MHz < crystal ≤ 6.5MHz
		010	6.5MHz < crystal ≤ 8.5MHz
		011	8.5MHz < crystal ≤ 10.5MHz
		100	10.5MHz < crystal ≤ 12.5MHz
		101	12.5MHz < crystal ≤ 16.5MHz
		Others	Reserved

NOTE:

1. The External Main Oscillator range (XRNS) should be changed when the IRC is selected as the system clock.

4.8.25 SCU_LVICR: Low-Voltage Indicator Control Register

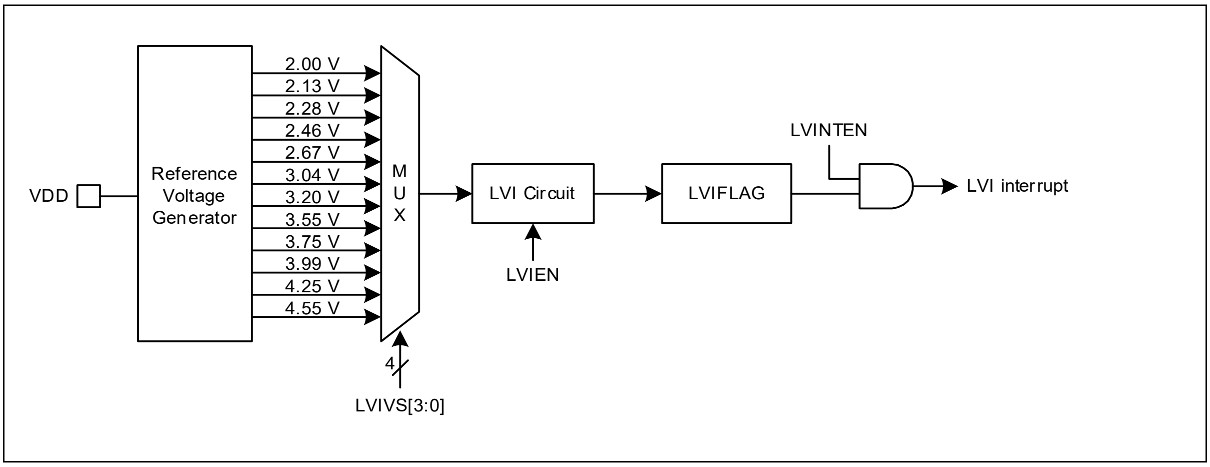
The SCU_LVICR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_LVICR=0x4000_5100

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			LVIEN	Reserved	LVINTEN	LVIFLAG	LVIVS[3:0]			
-			0	-	0	0	0	0	0	0
-			RW	-	RW	RW	RW	RW	RW	RW

7	LVIEN	LVI Enable.	
		0	Disable low-voltage indicator.
		1	Enable low-voltage indicator.
5	LVINTEN	LVI Interrupt Enable.	
		0	Disable low-voltage indicator interrupt.
		1	Enable low-voltage indicator interrupt.
4	LVIFLAG	LVI Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
3 0	LVIVS[3:0]	LVI Voltage Selection.	
		0100	2.00 V
		0101	2.13 V
		0110	2.28 V
		0111	2.46 V
		1000	2.67 V
		1001	3.04 V
		1010	3.20 V
		1011	3.55 V
		1100	3.75 V
		1101	3.99 V
		1110	4.25 V
		1111	4.55 V
		Other values	Not available

Figure 20. LVI Block Diagram



4.8.26 SCU_LVRCR: Low-Voltage Reset Control Register

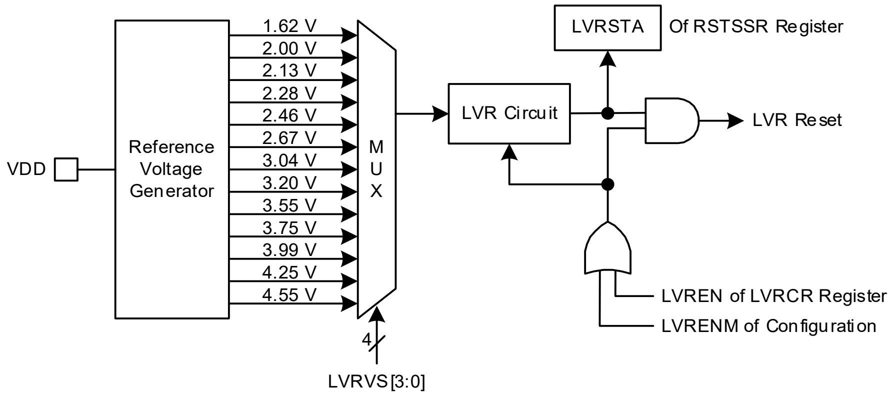
The SCU_LVRCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_LVRCR=0x4000_5104

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			LVREN[7:0]
-			0x00
.			RW

7 0	LVREN[7:0]	LVR Enable. These bits are cleared to 0x00 by POR only and retained by other reset signals.	
		0x55	Disable low-voltage reset.
		Others	Enable low-voltage reset.

Figure 21. LVR Block Diagram



NOTE:

1. Voltage level of LVR is set by the low voltage reset configuration register (CONF_LVRCNFIG) in the Configure Option Page 1.

4.8.27 SCU Register Map Summary

Table 15. SCU CHIPCONFIG Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SCU_VENDORID	VENDID[31:0]																															
	Reset value	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	1	0	1	1	0
0x04	SCU_CHIPID	CHIPID[31:0]																															
	Reset value	0	1	0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	x
0x08	SCU_REVNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVNO[7:0]							
	Reset value																									x	x	x	x	x	x	x	x
0x14	SCU_PMREMAP	WTIDKY[15:0]																nPMREM[7:0]								PMREM[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	SCU_RSTSSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MONSTA	SWSTA	EXTSTA	WDTSTA	LVRSTA	PORSTA
	Reset value																											x	x	x	x	x	x
0x20	SCU_NMISRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NMICON	MONINT	Res.	NMISRC[4:0]				
	Reset value																									0	0		0	0	0	0	0
0x24	SCU_SWRSTR	WTIDKY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWRST[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x28	SCU_SRSTVR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VALID[7:0]							
	Reset value																									0	1	0	1	0	1	0	1
0x2C	SCU_WUTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUTIEN	Res.	Res.	Res.	Res.	Res.	CNTRLD	WUTIFLAG
	Reset value																									0						0	0
0x30	SCU_WUTDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0
0xA8	SCU_HIRCTRM	WTIDKY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRMH[2:0]			FTRMH[4:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								x	x	x	x	x	x	x	x	x
0xAC	SCU_WDTRCTRM	WTIDKY[15:0]																nTRMW[7:0]								CTRMW[3:0]				Res.	FTRMW[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x

Table 16. SCU Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SCU_SCCR	WTIDKY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCLKSEL[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															0	0
0x04	SCU_CLKSRCR	WTIDKY[15:0]																Res.	Res.	HIRCSEL[1:0]		Res.	Res.	Res.	XMFRNG	Res.	Res.	Res.	Res.	WDTRCEN	HIRCEN	XMOSCEN	XSOSCEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0				0					1	1	0	0
0x08	SCU_SCDIVR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WLDIV[2:0]			Res.	HDIV[2:0]		
	Reset value																										0	0	0		0	0	0
0x0C	SCU_SCDIVR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSTDIV[1:0]		Res.	Res.	PDIV[1:0]	
	Reset value																											0	0			0	0
0x10	SCU_CLKOCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLKOEN	POLSEL	CLKODIV[2:0]			CLKOS[2:0]		
	Reset value																									0	0	0	0	0	0	0	0
0x14	SCU_CMONCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MONEN	MACTS[1:0]		Res.	MONFLAG	NMINTFG	MONCS[1:0]	
	Reset value																									0	0	0		0	0	0	0

Table 16. SCU Register Map Summary (continued)

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x20	SCU_PPCLKEN1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T21CLKE	T20CLKE	Res.	T12CLKE	T11CLKE	T10CLKE	Res.	Res.	Res.	Res.	Res.	T15CLKE	T14CLKE	T13CLKE	Res.	Res.	PFCLKE	PECLKE	PDCLKE	PCCLKE	PBCLKE	PACLKE
	Reset value											0	0		0	0	0						0	0	0			0	0	0	0	0	0
0x24	SCU_PPCLKEN2	DMACLKE	DFMCLKE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI1CLKE	SPI0CLKE	FMCLKE	LVICLKE	WDTCLKE	WTCLKE	Res.	Res.	Res.	CRCLKE	Res.	ADCLKE	Res.	I2C2CLKE	I2C1CLKE	I2C0CLKE	Res.	UST12CLKE	UT1CLKE	UT0CLKE	UST11CLKE	UST10CLKE
	Reset value	0	0									0	0	0	0	1	0				0		0		0	0	0		0	0	0	0	0
0x40	SCU_PPCLKSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T20CLK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WTCLK[1:0]		Res.	Res.	WDTCLK
	Reset value												0																0	0			0
0x60	SCU_PPRST1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T21RST	T20RST	Res.	T12RST	T11RST	T10RST	Res.	Res.	Res.	Res.	Res.	T15RST	T14RST	T13RST	Res.	Res.	PFRST	PERST	PDRST	PCRST	PBRST	PARST
	Reset value											0	0		0	0	0						0	0	0			0	0	0	0	0	0
0x64	SCU_PPRST2	DMARST	DFMCRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI1RST	SPI0RST	FMCRST	LVIRST	Res.	WTRST	Res.	Res.	Res.	CRRST	Res.	ADRST	Res.	I2C2RST	I2C1RST	I2C0RST	Res.	UST12RST	UT1RST	UT0RST	UST11RST	UST10RST
	Reset value	0	0									0	0	0	0		0				0		0		0	0	0		0	0	0	0	0
0x80	SCU_XTFLSR	WTIDKY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XRNS[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														1	0	1

Table 17. SCU LVI/LVR Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SCU_LVICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVIEN	Res.	LVINTEN	LVIFLAG	LVIVS[3:0]			
	Reset value																									0		0	0	0	0	0	0
0x04	SCU_LVRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVREN[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

5. Port Control Unit (PCU) and GPIO

5.1 Introduction

5.1.1 PCU Introduction

The A31S134 has a Port Control Unit (PCU) module that controls the external input and output (I/O) ports. By setting the PCU registers, users can configure the pins' uses, input/output direction, pull-up/pull-down resistors, and debouncing for their applications as needed.

5.1.2 GPIO Introduction

Pins except the VDD, GND, and certain specific-purpose pins can be used as General-Purpose Input/Output (GPIO) pins.

The GPIO module controls the general I/O ports. Output pins can be set to generate high- or low-level signals by configuring the corresponding bits of the GPIO control register, while logic input pins can be monitored for their input status in the control registers.

5.2 Main Features

5.2.1 PCU Features

The Port Control Unit (PCU) configures and controls external I/Os as listed below:

- MOD registers define the use of each pin.
  - Input / Output / Alternative function
- Internal pull-up and pull-down resistors and push-pull/open-drain mode can be configured for each pin.
- Interrupts listed below can be set for each pin:
  - Input level interrupt
  - Input rising-edge interrupt
  - Input falling-edge interrupt
  - Input both-edge interrupt
- Up to four GPIO interrupts are supported, including GPIOn (n = B, C, E, and F).
- Each pin can be set for debouncing.

5.2.2 GPIO Features

The GPIO module controls the GPIO ports as listed below:

- Selects the output signal level.
- Enables or disables the pull-up and pull-down resistors for pins.

5.3 GPIO Functional Description

Depending on the specific hardware characteristics of each I/O port listed in the DC characteristics of the datasheet, each port bit on a general purpose I/O (GPIO) port can be individually set to several modes by software:

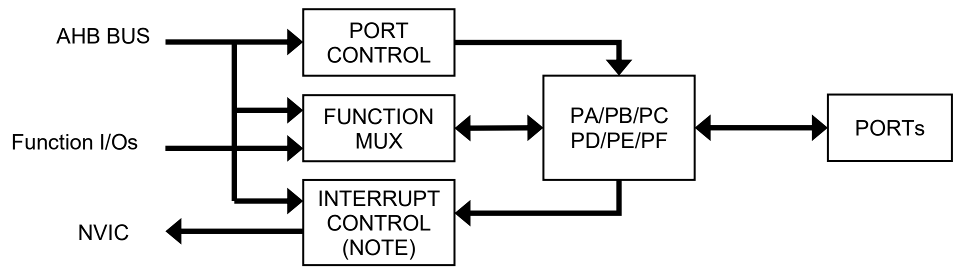
- Input floating
- Input Pull-up
- Input Pull-down
- Analog
- Output Open-drain with Pull-up or Pull-down capability
- Output Push-pull with Pull-up or Pull-down capability
- Alternative function

During and just after reset, the alternate functions are not active and most of the I/O ports are configured in input mode. The debug pins are configured in alternative functions and pull-up/pull-down resistor after reset:

- PF1: SWCLK with pull-down resistor
- PF0: SWDIO with pull-up resistor

Following figures (Figure 22, Figure 23) show block diagrams of GPIO and external interrupt I/O pins, respectively.

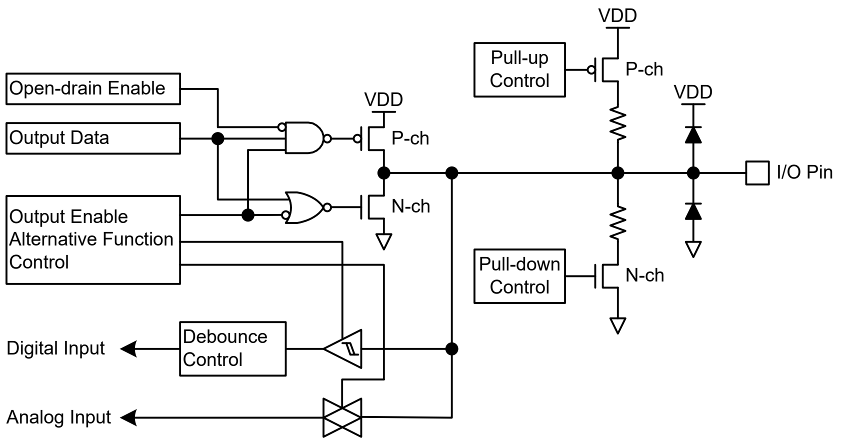
Figure 22. PCU/GPIO Block Diagram



NOTE:

1. Refer to section 3.1.2 Interrupt Controller.

Figure 23. I/O Port Block Diagram (GPIO Pins)



5.3.1 GPIO Pins and Internal Signals

Table 18 describes PCU and GPIO internal signal.

Table 18. PCU and GPIO Internal Signal

Signal Name	Signal Type	Description
HCLK	clock	AHB clock

Table 19 describes six pins assigned for PCU and GPIO module.

Table 19. PCU and GPIO Pins

Pin Name	Type	Description
PA	IO	PA0 to PA9
PB	IO	PB0 to PB11
PC	IO	PC0 to PC10
PD	IO	PD0 to PD8
PE	IO	PE0 to PE8
PF	IO	PF0 to PF9

5.3.2 I/O Port Control Registers

Each GPIO controls up to sixteen I/Os by using three 32-bit memory mapping control registers such as the Pn_MOD, Pn_TYP, Pn_AFSR, and Pn_PUPD registers as shown below:

- The Pn_MOD register selects the I/O mode from the Input, Output, and Alternative modes.
- The Pn_TYP register selects the push-pull/open-drain output of each port.
- The Pn_PUPD register controls the pull-up/pull-down resistors of all ports I/Os.

5.3.3 I/O Port Data Registers

Each GPIO has two 16-bit memory mapping data registers that operate as shown below:

- The Pn_OUTDR register can be read and written when it has output data.
- The Pn_INDR register stores input data entered via the I/O pin. It can only be read.

5.3.4 I/O Data Bitwise Handling

The bit set register (Pn_BSR) and the bit clear register (Pn_BCR) can be set or clear by the application that controls corresponding bit of the Pn_OUTDR register. These registers support bit banding.

5.3.5 I/O Alternative Function

Each GPIO has four 32-bit memory mapping alternative function registers that operate as shown below:

- The Pn_AFSR1 register selects an alternative function for each port I/O pin (pin 0 to 7).
- The Pn_AFSR2 register selects an alternative function for each port I/O pin (pin 8 to 15).

The direction of the alternative function is automatically determined according to the purpose, whether it is input or output.

Table 20 describes the GPIO alternative functions.

Table 20. GPIO Alternative Function

Port	Pin	Alternate Function					
		AF0	AF1	AF2	AF3	AF4	AF5
PA	0	AN0	AVREF	–	–	–	–
	1	AN1	–	–	–	–	–
	2	AN2	–	–	–	–	–
	3	AN3	–	–	–	–	–
	4	AN4	–	–	–	–	–
	5	AN5	–	–	–	–	–
	6	AN6	–	–	MOSI11	TXD11	–
	7	AN7	–	–	MISO11	RXD11	–
	8	AN8	–	–	SCK11	RXD1	–
	9	AN9	–	–	–	TXD1	SS10
PB	0	T21OUT	T21CAP	EC21	SCK10	–	SCL1
	1	T20OUT	T20CAP	EC20	MISO10	RXD10	SDA1
	2	T15OUT	T15CAP	EC15	MOSI10	TXD10	–
	3	T14OUT	T14CAP	EC14	MOSI12	TXD12	SDA0
	4	T13OUT	T13CAP	EC13	MISO12	RXD12	SCL0
	5	T12OUT	T12CAP	EC12	SCK12	–	–
	6	T11OUT	T11CAP	EC11	MISO0	RXD0	SS12
	7	T12OUT	T12CAP	EC12	MOSI0	TXD0	–
	8	T11OUT	T11CAP	EC11	SCK0	–	–
	9	–	–	–	–	–	–
	10	–	–	–	–	RXD1	–
	11	–	–	–	–	TXD1	–
PC	0	T21OUT	T21CAP	EC21	–	–	–
	1	T14OUT	T14CAP	EC14	–	–	–
	2	AN10	CLKO	–	T10OUT	T10CAP	–
	3	AN11	–	–	T10OUT	T10CAP	–
	4	–	–	–	–	–	–
	5	–	–	EC10	–	–	–
	6	T10OUT	T10CAP	–	–	–	–
	7	AN12	–	–	MOSI11	TXD11	–
	8	AN13	–	–	MISO11	RXD11	SDA0
	9	–	–	–	SCK11	–	SCL0
	10	–	–	–	SS11	–	–

Table 20. GPIO Alternative Function (continued)

Port	Pin	Alternate Function					
		AF0	AF1	AF2	AF3	AF4	AF5
PD	0	–	–	–	–	–	SCL0
	1	–	–	–	–	–	SDA0
	2	T21OUT	T21CAP	EC21	–	–	–
	3	T20OUT	T20CAP	EC20	–	–	–
	4	T13OUT	T13CAP	EC13	SCK12	–	–
	5	–	–	–	MOSI12	TXD12	SCL2
	6	T13OUT	T13CAP	EC13	MISO12	RXD12	SDA2
	7	–	–	–	SCK1	–	SCL2
	8	–	–	–	MISO1	–	SDA2
PE	0	–	–	–	MOSI1	–	–
	1	–	–	–	MOSI0	–	SS1
	2	–	–	–	MISO0	SDA1	–
	3	–	–	–	SCK0	SCL1	–
	4	T20OUT	T20CAP	EC20	–	–	SS0
	5	T15OUT	T15CAP	EC15	–	–	–
	6	–	–	–	SCK1	–	–
	7	–	–	–	MISO1	–	–
	8	–	–	–	MOSI1	–	–
PF	0	SWDIO	–	–	–	–	–
	1	SWCLK	–	–	–	–	–
	2	–	–	–	–	–	–
	3	SXIN	–	–	–	–	–
	4	SXOUT	–	–	–	–	–
	5	XIN	–	–	–	–	–
	6	XOUT	–	–	–	–	–
	7	–	–	–	–	–	–
	8	–	–	–	–	–	–
	9	–	–	–	–	–	–

NOTES:

1. An unused pin shouldn’t be configured as an input floating.
2. After reset, the PF0 and PF1 pins are configured as SWDIO and SWCLK alternative functions, and the internal pull-down on SWCLK and the internal pull-up on SWDIO are activated.
3. The SWCLK and SWDIO pins shouldn’t be changed as other alternative functions by software during the pins are connected with debugger host.

5.3.6 Interrupt Functionality

The GPIO module has four interrupt sources, and each port can be an interrupt source. To set each GPIO pin as interrupt sources, users must configure the interrupt trigger selection register (INTC_PnTRIG) and the interrupt control register (INTC_PnCR).

A level-triggered or edge-triggered interrupt can be set for each pin. Once an interrupt occurs, the pin's corresponding bit of the INTC_PnFLAG register is flagged. This flag can be cleared by writing a '1' to the bit.

Table 21. Interrupt Sources and Corresponding Pins in GPIO Module

Interrupt Name	Configurable Pins
GPIOB	Port B[11:0]
GPIOC	Port C[7:0]
GPIOE	Port E[8:0]
GPIOF	Port F[4:2]

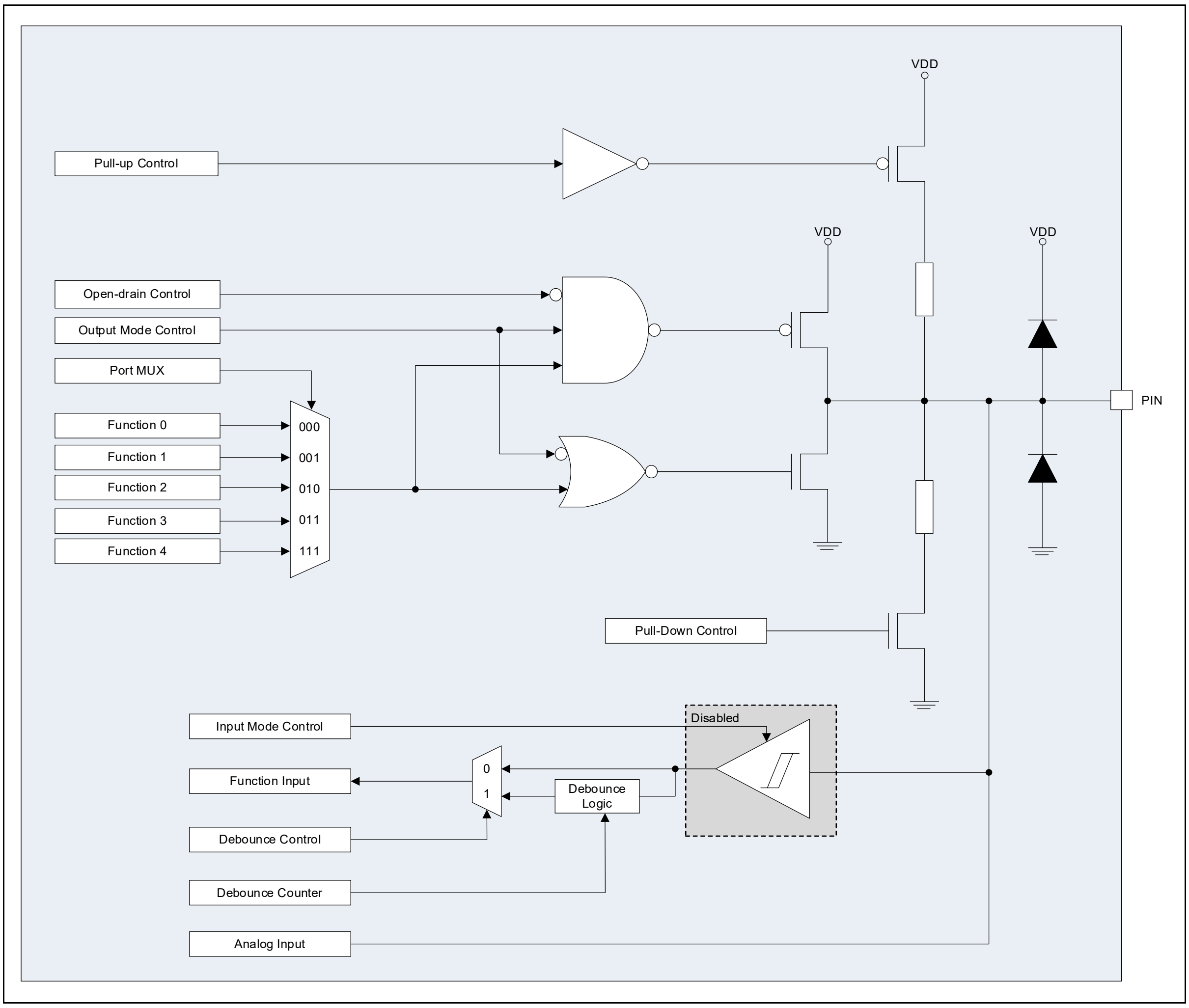
5.3.7 Output Configuration

When the I/O port is programed as output, it features the followings:

- The push-pull driver is turned on or off depending on the Pn_TYP register or function output value.
- Open-drain mode is available.
- The use of the pull-up and pull-down resistors is determined by the Pn_PUPD register.

Figure 24 shows a block diagram of an output port.

Figure 24. Output Port Block Diagram



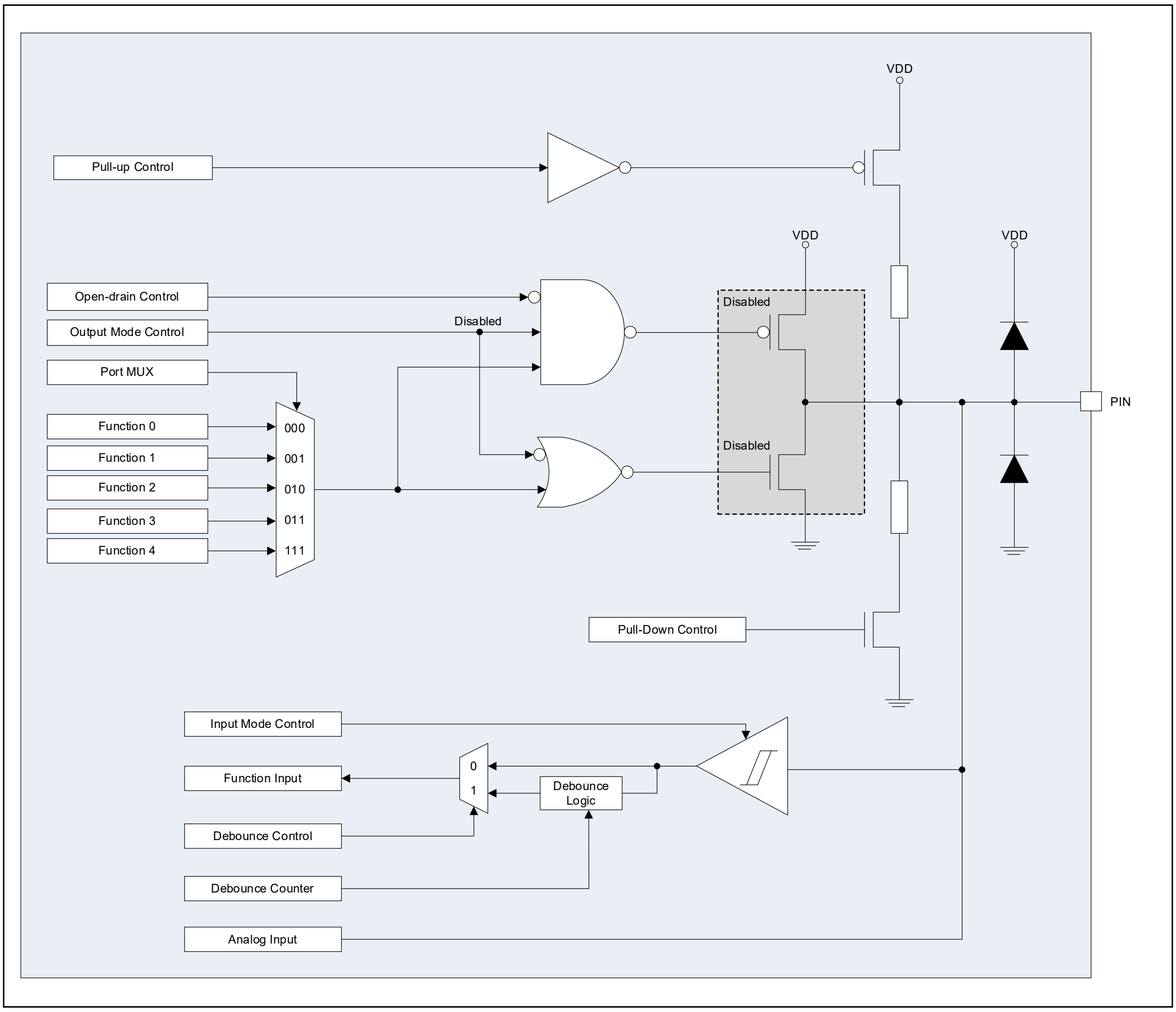
5.3.8 Input Configuration

When the I/O port is programed as input, it features the followings:

- All Push-pull drivers are turned off, and Input Schmitt Trigger opens its input.
- Input data is stored in the Pn_INDR register, or input signal enters toward functions.

Figure 25 shows a block diagram of an input port.

Figure 25. Input Port Block Diagram



5.3.9 Alternative Function Configuration

For the A31S134, all pins except those with specific purposes can be used as GPIO pins, and the GPIO function can be set by the Pn_MOD register on each port.

Users can select different functions for each port' pins based on the settings of the alternative function selection register. The input data registers capture data entering each I/O pin, as either un-debounced or debounced, at every GPIO clock cycle.

5.3.10 Debouncing Functionality

For the A31S134, each port is capable of debouncing input signals. Debouncing is to filter out noise that can interfere with the port's data input.

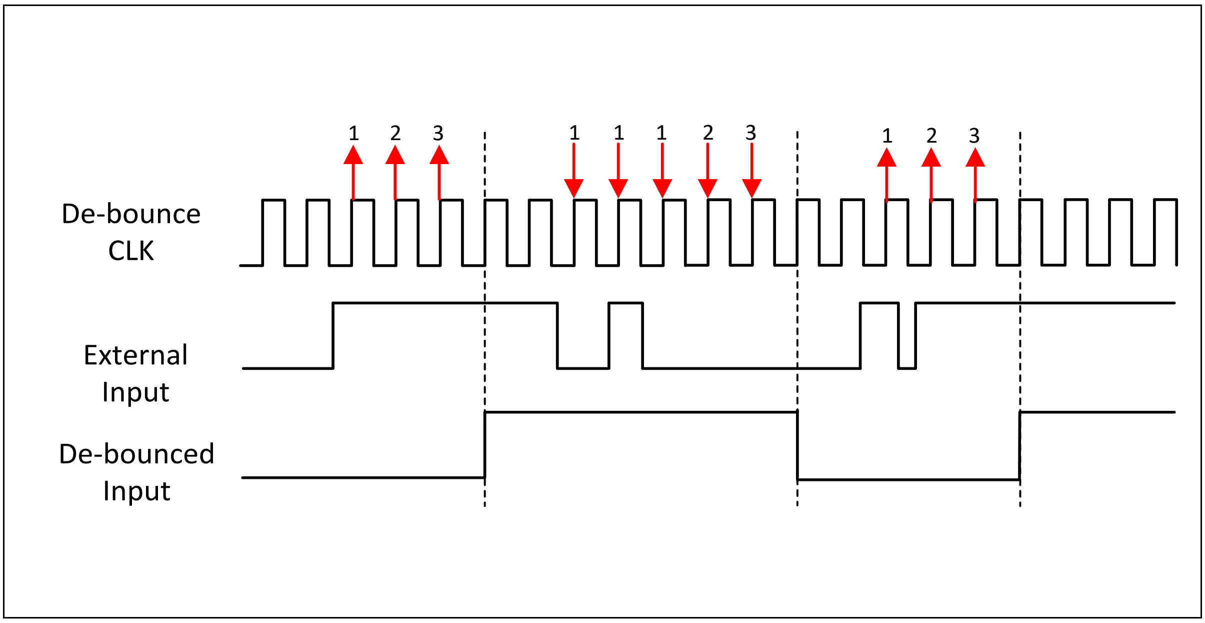
Filtering levels can be adjusted for each 16-pin port, and each individual pin is configured based on whether filtering is enabled or disabled. The debouncing clock used for each port is set in the Pn_DBCR registers.

When enabled, the debouncing logic is clocked at the frequency division ratio set in the Pn_DBCR.DBCLK[2:0] bits, depending on which register the port belongs to.

If a level is not detected on an enabled pin three or more times in a low at the sampling clock, the signal is eliminated as noise.

Figure 26 shows a timing diagram of a port debouncing.

Figure 26. Example Timing Diagram of Port Debouncing



5.4 GPIO Registers

The base address and register map of PCU are described in the followings:

Table 22. Base Address of PCU

Name	Base Address	Description
PA	0x3000_0000	General Port A
PB	0x3000_0100	General Port B
PC	0x3000_0200	General Port C
PD	0x3000_0300	General Port D
PE	0x3000_0400	General Port E
PF	0x3000_0500	General Port F

Table 23. PCU and GPIO Register Map

Name	Offset	Type	Description	Reset Value	Reference
Pn_MOD	0x00	RW	Port n Mode Register	0x00000000	5.4.1
Pn_TYP	0x04	RW	Port n Output Type Selection Register	0x00000000	5.4.2
Pn_AFSR1	0x08	RW	Port n Alternative Function Selection Register 1	0x00000000	5.4.3
Pn_AFSR2	0x0C	RW	Port n Alternative Function Selection Register 2	0x00000000	5.4.3
Pn_PUPD	0x10	RW	Port n Pull-up/down Resistor Selection Register	0x00000000	5.4.4
Pn_INDR	0x14	RO	Port n Input Data Register	0x0000xxxx	5.4.5
Pn_OUTDR	0x18	RW	Port n Output Data Register	0x00000000	5.4.6
Pn_BSR	0x1C	WO	Port n Output Bit Set Register	0x00000000	5.4.7
Pn_BCR	0x20	WO	Port n Output Bit Clear Register	0x00000000	5.4.8
Pn_OUTDMSK	0x24	RW	Port n Output Data Mask Register	0x00000000	5.4.9
Pn_DBCR	0x28	RW	Port n Debounce Control Register	0x00000000	5.4.10

NOTE:

1. n = A, B, C, D, E, and F.

5.4.1 Pn_MOD: Port n Mode Register

The Pn_MOD register selects one from input mode and output mode for each port pin. Each pin can be configured as an input pin, an output pin, or an Alternative Function pin.

The Pn_MOD register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_MOD=0x3000_0000, PB_MOD=0x3000_0100, PC_MOD=0x3000_0200
PD_MOD=0x3000_0300, PE_MOD=0x3000_0400, PF_MOD=0x3000_0500

31 30 29 28 27 26 25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
Reserved	MODE11[1:0]	MODE10[1:0]	MODE9[1:0]	MODE8[1:0]	MODE7[1:0]	MODE6[1:0]	MODE5[1:0]	MODE4[1:0]	MODE3[1:0]	MODE2[1:0]	MODE1[1:0]	MODE0[1:0]
-	00	00	00	00	00	00	00	00	00	00	00	00
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	MODEx[1:0]	Port n Mode Selection bits, x = 0 to 11.	
		00	Input mode
		01	Output mode
		10	Alternative function mode
		11	Reserved

NOTES:

1. The MODEx bits for PF[4:3] are clear to “00” by the reset of POR but retained by the other reset.
2. The mode bits for PF[1:0] are set to “10” for alternative function by reset.
3. PF0: SWDIO, PF1: SWCLK

5.4.2 Pn_TYP: Port n Output Type Selection Register

The Pn_TYP register selects an output type of a port pin from Push-pull output and Open-drain output.

The Pn_TYP register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_TYP=0x3000_0004, PB_TYP=0x3000_0104, PC_TYP=0x3000_0204
PD_TYP=0x3000_0304, PE_TYP=0x3000_0404, PF_TYP=0x3000_0504

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TYP11	TYP10	TYP9	TYP8	TYP7	TYP6	TYP5	TYP4	TYP3	TYP2	TYP1	TYP0
-	0	0	0	0	0	0	0	0	0	0	0	0
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	TYPx	Port n Output Type Selection bit, x = 0 to 11.	
		0	Push-pull output
		1	Open-drain output

5.4.3 Pn_AFSR1/2: Port n Alternative Function Selection Register 1/2

The Pn_AFSR1/2 register must be set properly before using the port. Otherwise, the port may not function properly.

The Pn_AFSR1/2 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

PA_AFSR1=0x3000_0008, PB_AFSR1=0x3000_0108, PC_AFSR1=0x3000_0208
PD_AFSR1=0x3000_0308, PE_AFSR1=0x3000_0408, PF_AFSR1=0x3000_0508

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
AFSR7[3:0]	AFSR6[3:0]	AFSR5[3:0]	AFSR4[3:0]	AFSR3[3:0]	AFSR2[3:0]	AFSR1[3:0]	AFSR0[3:0]
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4x+3 4x	AFSRx[3:0]	Port n Alternative Function Selection bits, x = 0 to 7	
		0000	Alternative Function 0 (AF0)
		0001	Alternative Function 1 (AF1)
		0010	Alternative Function 2 (AF2)
		0011	Alternative Function 3 (AF3)
		0100	Alternative Function 4 (AF4)
		0101	Alternative Function 5 (AF5)
		Others	Reserved

NOTE:

1. If XSOSC clock is used, both PF[3:4] pins should be selected as SXIN and SXOUT functions by the corresponding bits of PF_AFSR1 register.

PA_AFSR2=0x3000_000C, PB_AFSR2=0x3000_010C, PC_AFSR2=0x3000_020C
PD_AFSR2=0x3000_030C, PE_AFSR2=0x3000_040C, PF_AFSR2=0x3000_050C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Reserved	AFSR11[3:0]	AFSR10[3:0]	AFSR9[3:0]	AFSR8[3:0]
-	0000	0000	0000	0000
-	RW	RW	RW	RW

4(x-8)+3 4(x-8)	AFSRx[3:0]	Port n Alternative Function Selection bits, x = 8 to 11	
		0000	Alternative Function 0 (AF0)
		0001	Alternative Function 1 (AF1)
		0010	Alternative Function 2 (AF2)
		0011	Alternative Function 3 (AF3)
		0100	Alternative Function 4 (AF4)
		0101	Alternative Function 5 (AF5)
		Others	Reserved

5.4.4 Pn_PUPD: Port n Pull-Up/Pull-Down Resistor Selection Register

Every pin of the port has an on-chip pull-up/pull-down resistor, which can be configured by Pn_PUPD registers.

The Pn_PUPD register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_PUPD=0x3000_0010, PB_PUPD=0x3000_0110, PC_PUPD=0x3000_0210
PD_PUPD=0x3000_0310, PE_PUPD=0x3000_0410, PF_PUPD=0x3000_0510

31 30 29 28 27 26 25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
Reserved	PUPD11[1:0]	PUPD10[1:0]	PUPD9[1:0]	PUPD8[1:0]	PUPD7[1:0]	PUPD6[1:0]	PUPD5[1:0]	PUPD4[1:0]	PUPD3[1:0]	PUPD2[1:0]	PUPD1[1:0]	PUPD0[1:0]
-	00	00	00	00	00	00	00	00	00	00	00	00
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	PUPDx[1:0]	Port n pull-up/pull-down resistor selection bits, x = 0 to 11.	
		00	Disable pull-up/pull-down resistor
		01	Enable pull-up resistor
		10	Enable pull-down resistor
		11	Reserved

NOTES:

1. The pull-up/pull-down resistor of PF[6:3] is automatically disabled regardless of the corresponding PUPDx value. If the pins are configured as the alternative function for X-tal (XIN, XOUT, SXIN, and SXOUT).
2. The PUPDx bits for PF0 and PF1 are set to “01” and “10” for SWDIO/SWCLK by reset, respectively.
3. PF0: SWDIO, PF1: SWCLK

5.4.5 Pn_INDR: Port n Input Data Register

Each pin level status can be read in the Pn_INDR register. Except for analog input and alternative mode output, the pin level can be detected in the Pn_INDR register.

The Pn_INDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_INDR=0x3000_0014, PB_INDR=0x3000_0114, PC_INDR=0x3000_0214
PD_INDR=0x3000_0314, PE_INDR=0x3000_0414, PF_INDR=0x3000_0514

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				INDR11	INDR10	INDR9	INDR8	INDR7	INDR6	INDR5	INDR4	INDR3	INDR2	INDR1	INDR0
-																				0	0	0	0	0	0	0	0	0	0	0	0
-																				RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

x	INDRx	Port n Input Data bit, x = 0 to 11.

5.4.6 Pn_OUTDR: Port n Output Data Register

When a pin is set as an output in GPIO mode, the output level of the pin is defined by Pn_OUTDR registers.

The Pn_OUTDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_OUTDR=0x3000_0018, PB_OUTDR=0x3000_0118, PC_OUTDR=0x3000_0218
PD_OUTDR=0x3000_0318, PE_OUTDR=0x3000_0418, PF_OUTDR=0x3000_0518

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				OUTDR11	OUTDR10	OUTDR9	OUTDR8	OUTDR7	OUTDR6	OUTDR5	OUTDR4	OUTDR3	OUTDR2	OUTDR1	OUTDR0
-																				0	0	0	0	0	0	0	0	0	0	0	0
-																				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDRx	Port n Output Data bit, x = 0 to 11. The OUTDR bits can be individually set or cleared by writing to the Pn_BSR/Pn_BCR register.

5.4.7 Pn_BSR: Port n Output Bit Set Register

The Pn_BSR register is used for controlling each bit of the Pn_OUTDR register. Writing a ‘1’ into the specific bit position will set a corresponding bit of Pn_OUTDR to ‘1’. Writing ‘0’ in the register has no effect.

The Pn_BSR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_BSR=0x3000_001C, PB_BSR=0x3000_011C, PC_BSR=0x3000_021C
PD_BSR=0x3000_031C, PE_BSR=0x3000_041C, PF_BSR=0x3000_051C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				BSR11	BSR10	BSR9	BSR8	BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0
-																				0	0	0	0	0	0	0	0	0	0	0	0
-																				WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

x	BSRx	Port n Output Set bit, x = 0 to 11. These bits are always read to 0x00.	
		0	No effect
		1	Set the corresponding OUTDRx bit (Automatically cleared to ‘0’)

5.4.8 Pn_BCR: Port n Output Bit Clear Register

The Pn_BCR register is used for controlling each bit of the Pn_OUTDR register. Writing a ‘1’ into the specific bit will set a corresponding bit of Pn_OUTDR to ‘0’. Writing ‘0’ in this register has no effect.

The Pn_BCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_BCR=0x3000_0020, PB_BCR=0x3000_0120, PC_BCR=0x3000_0220
PD_BCR=0x3000_0320, PE_BCR=0x3000_0420, PF_BCR=0x3000_0520

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				BCR11	BCR10	BCR9	BCR8	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
-																				0	0	0	0	0	0	0	0	0	0	0	0
-																				WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

x	BCRx	Port n Output Clear bit, x = 0 to 11. These bits are always read to 0x00.	
		0	No effect
		1	Clear the corresponding OUTDRx bit (Automatically cleared to ‘0’)

5.4.9 Pn_OUTDMSK: Port n Output Data Mask Register

The Pn_OUTDMSK register is used for protecting each bit of Pn_OUTDR register. Writing a ‘1’ into the specific bit will protect a corresponding bit of Pn_OUTDR. Writing ‘0’ in this register is unmask.

The Pn_OUTDMSK register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_OUTDMSK=0x3000_0024, PB_OUTDMSK=0x3000_0124, PC_OUTDMSK=0x3000_0224
PD_OUTDMSK=0x3000_0324, PE_OUTDMSK=0x3000_0424, PF_OUTDMSK=0x3000_0524

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				OUTDMSK11	OUTDMSK10	OUTDMSK9	OUTDMSK8	OUTDMSK7	OUTDMSK6	OUTDMSK5	OUTDMSK4	OUTDMSK3	OUTDMSK2	OUTDMSK1	OUTDMSK0
-																				0	0	0	0	0	0	0	0	0	0	0	0
-																				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDMSKx	Port n Output Data Mask bit, x = 0 to 11.	
		0	Unmask. The corresponding OUTDRx bit can be changed.
		1	Mask. The corresponding OUTDRx bit is protected.

5.4.10 Pn_DBCR: Port n Debounce Control Register

The Pn_DBCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = A to F).

PA_DBCR=0x3000_0028, PB_DBCR=0x3000_0128, PC_DBCR=0x3000_0228
PD_DBCR=0x3000_0328, PE_DBCR=0x3000_0428, PF_DBCR=0x3000_0528

31 30 29 28 27 26 25 24 23 22 21 20 19	18	17	16	15 14 13 12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DBCLK[2:0]			Reserved	DBEN11	DBEN10	DBEN9	DBEN8	DBEN7	DBEN6	DBEN5	DBEN4	DBEN3	DBEN2	DBEN1	DBEN0
-	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0
-	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

18 16	DBCLK[2:0]	Port n Debounce Filter Sampling Clock Selection.	
		000	HCLK/1
		001	HCLK/4
		010	HCLK/16
		011	HCLK/64
		100	HCLK/256
		101	HCLK/1024
		110	Reserved
		111	Reserved
x	DBENx	Port n Debounce Enable bit, x = 0 to 11.	
		0	Disable debounce filter
		1	Enable debounce filter

NOTES:

1. If a level is not detected on an enabled pin three or more times in a row at the sampling clock, the signal is eliminated as noise.
2. The port debounce should be disabled before DEEP SLEEP mode.
3. The A31S134 has the debounce filters for all ports.

5.4.11 PCU and GPIO Register Map Summary

Table 24. PCU and GPIO Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	Pn_MOD	Res	Res	Res	Res	Res	Res	Res	Res	MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	Pn_TYP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TYP11	TYP10	TYP9	TYP8	TYP7	TYP6	TYP5	TYP4	TYP3	TYP2	TYP1	TYP0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x08	Pn_AFSR1	AFSR7[3:0]				AFSR6[3:0]				AFSR5[3:0]				AFSR4[3:0]				AFSR3[3:0]				AFSR2[3:0]				AFSR1[3:0]				AFSR0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	Pn_AFSR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AFSR11[3:0]				AFSR10[3:0]				AFSR9[3:0]				AFSR8[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	Pn_PUPD	Res	Res	Res	Res	Res	Res	Res	Res	PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	Pn_INDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	INDR11	INDR10	INDR9	INDR8	INDR7	INDR6	INDR5	INDR4	INDR3	INDR2	INDR1	INDR0
	Reset value																					x	x	x	x	x	x	x	x	x	x	x	x
0x18	Pn_OUTDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OUTDR11	OUTDR10	OUTDR9	OUTDR8	OUTDR7	OUTDR6	OUTDR5	OUTDR4	OUTDR3	OUTDR2	OUTDR1	OUTDR0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x1C	Pn_BSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BSR11	BSR10	BSR9	BSR8	BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x20	Pn_BCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BCR11	BCR10	BCR9	BCR8	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x24	Pn_OUTDMSK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OUTDMSK11	OUTDMSK10	OUTDMSK9	OUTDMSK8	OUTDMSK7	OUTDMSK6	OUTDMSK5	OUTDMSK4	OUTDMSK3	OUTDMSK2	OUTDMSK1	OUTDMSK0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x28	Pn_DBCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBCLK[2:0]			Res	Res	Res	Res	DBEN11	DBEN10	DBEN9	DBEN8	DBEN7	DBEN6	DBEN5	DBEN4	DBEN3	DBEN2	DBEN1	DBEN0
	Reset value														0	0	0					0	0	0	0	0	0	0	0	0	0	0	0

5.5 Interrupt Registers

The base address and register map of the interrupt registers are shown in the followings:

Table 25. Base Address of Interrupt Register

Name	Base Address	Description
Interrupt Register	0x4000_1000	Interrupt Register

Table 26. Interrupt Register Map

Name	Offset	Type	Description	Reset Value	Reference
INTC_PnTRIG	0x0000	RW	Port n Interrupt Trigger Selection Register	0x00000000	5.5.1
INTC_PnCR	0x0100	RW	Port n Interrupt Control Register	0x00000000	5.5.2
INTC_PnFLAG	0x0200	RW	Port n Interrupt Flag Register	0x00000000	5.5.3
INTC_EINTxCONF1 INTC_EINTxCONF2	0x0300	RW	External Interrupt Configuration Register1, 2	0x00000000	5.5.4 5.5.5
INTC_MSK	0x0400	RW	Interrupt Source Mask Register	0x00000000	5.5.6

NOTES:

1. n = B, C, E, and F.
2. x = 0 to 3.

5.5.1 INTC_PnTRIG: Port n Interrupt Trigger Selection Register

The INTC_PnTRIG register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_PBTRIG =0x4000_1004, INTC_PCTRIG =0x4000_1008,
INTC_PETRIG =0x4000_1010, INTC_PFTRIG =0x4000_1014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				ITRIG11	ITRIG10	ITRIG9	ITRIG8	ITRIG7	ITRIG6	ITRIG5	ITRIG4	ITRIG3	ITRIG2	ITRIG1	ITRIG0
-																				0	0	0	0	0	0	0	0	0	0	0	0
.																				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	ITRIGx	Port n Interrupt Trigger Selection bits, x = 0 to 11.	
		0	Edge trigger interrupt
		1	Level trigger interrupt

NOTE:

1. The A31S134 has only the INTC_PnTRIG registers for PB[11:0], PC[7:0], PE[8:0], PF[4:2].

5.5.2 INTC_PnCR: Port n Interrupt Control Register

The INTC_PnCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_PBCR=0x4000_1104, INTC_PCCR =0x4000_1108,
INTC_PECR =0x4000_1110, INTC_PFCR =0x4000_1114

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								INTCTL11[1:0]		INTCTL10[1:0]		INTCTL9[1:0]		INTCTL8[1:0]		INTCTL7[1:0]		INTCTL6[1:0]		INTCTL5[1:0]		INTCTL4[1:0]		INTCTL3[1:0]		INTCTL2[1:0]		INTCTL1[1:0]		INTCTL0[1:0]	
-								00		00		00		00		00		00		00		00		00		00		00		00	
.								RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW	

2x+1: 2x	INTCTLx[1:0]	Port n Interrupt Control bits, x = 0 to 11.	
		00	Disable external interrupt (the flag bit won't be set)
		01	Interrupt on falling edge or on low level
		10	Interrupt on rising edge or on high level
		11	Interrupt on both of falling and rising edge or no level interrupt

NOTES:

1. The A31S134 has only the INTC_PnCR registers for PB[11:0], PC[7:0], PE[8:0], PF[4:2].
2. Do not write "11" to the corresponding INTCTLx[1:0] bits during ITRIGx bit of INTC_PnTRIG is '1'. If so, it may cause a malfunction.

5.5.3 INTC_PnFLAG: Port n Interrupt Flag Register

The INTC_PnFLAG register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_PBFLAG =0x4000_1204, INTC_PCFLAG =0x4000_1208,
INTC_PEFLAG =0x4000_1210, INTC_PFFLAG =0x4000_1214

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				FLAG11	FLAG10	FLAG9	FLAG8	FLAG7	FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
-																				0	0	0	0	0	0	0	0	0	0	0	0
.																				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	FLAGx	Port n Interrupt Flag bits, x 0 to 11.	
		0	No request occurred
		1	Request occurred, This bit is cleared to '0' when write '1'.

NOTE:

1. The A31S134 has only the INTC_PnFLAG registers for PB[11:0], PC[7:0], PE[8:0], PF[4:2].

5.5.4 INTC_EINTnCONF1: External Interrupt Configuration Register 1

The INTC_EINTnCONF1 register is a 32-bit size and sets the external interrupt pin assignment. This register is available at 32-/ 16-/ 8-bit access.

INTC_EINT0CONF1=0x4000_1300, INTC_EINT1CONF1=0x4000_1304,
INTC_EINT2CONF1=0x4000_1308, INTC_EINT3CONF1=0x4000_130C

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
CONF7[3:0]	CONF6[3:0]	CONF5[3:0]	CONF4[3:0]	CONF3[3:0]	CONF2[3:0]	CONF1[3:0]	CONF0[3:0]
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4x+3: 4x	CONFx[3:0]	Configuration bits for External Interrupt Group n, x = 0 to 7.	
		0001	PBx
		0010	PCx
		0100	PEx
		0101	PFx

NOTE:

1. The A31S134 has only the External Interrupts for PB[11:0], PC[7:0], PE[8:0], PF[4:2].

5.5.5 INTC_EINTnCONF2: External Interrupt Configuration Register 2

The INTC_EINTnCONF2 register is a 32-bit size and sets the external interrupt pin assignment. . This register is available at 32-/ 16-/ 8-bit access.

INTC_EINT0CONF2=0x4000_1310, INTC_EINT1CONF2=0x4000_1314, INTC_EINT2CONF2=0x4000_1318, INTC_EINT3CONF2=0x4000_131C

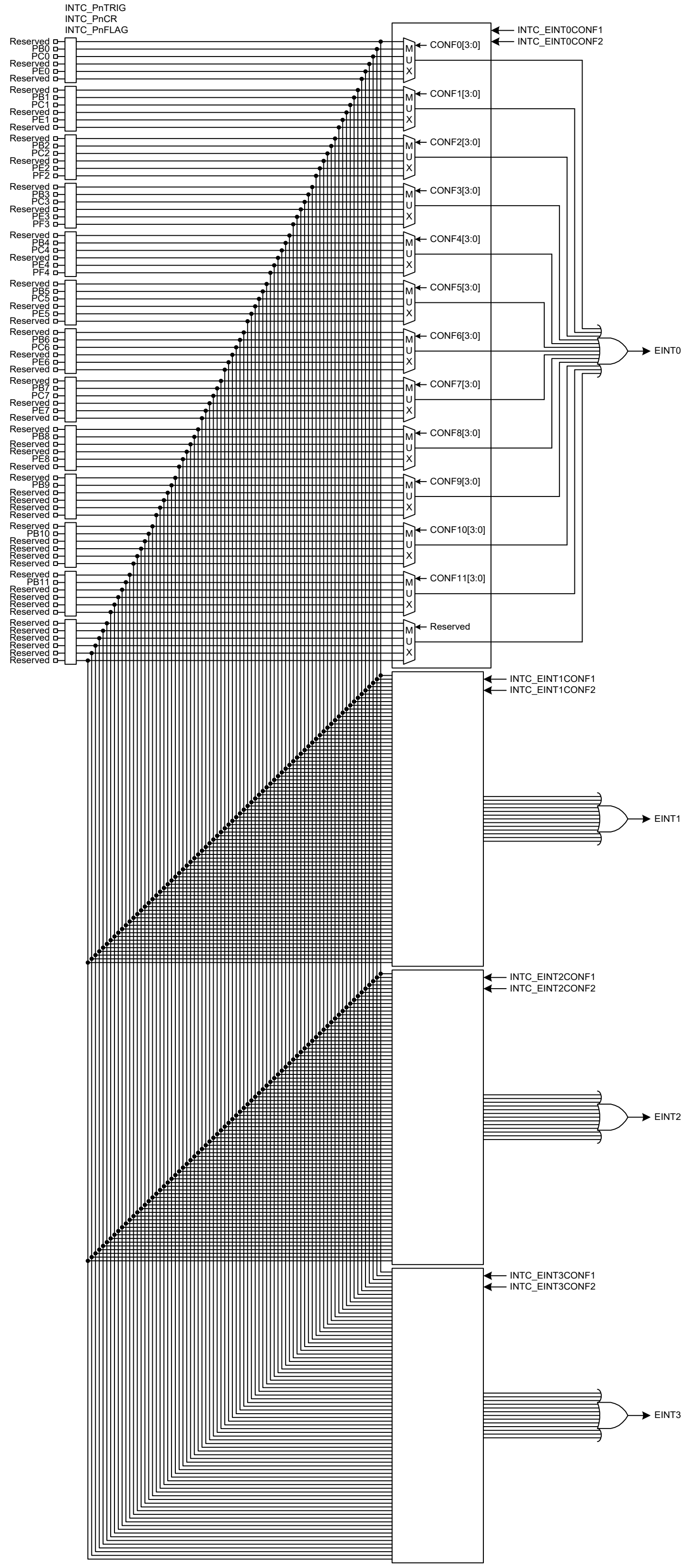
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Reserved	CONF11[3:0]	CONF10[3:0]	CONF9[3:0]	CONF8[3:0]
-	0000	0000	0000	0000
-	RW	RW	RW	RW

4(x-8)+3 :4(x-8)	CONFx[3:0]	Configuration bits for External Interrupt Group n, x = 8 to 11.
		0001 PBx
		0010 PCx
		0100 PEx
		0101 PFx

NOTE:

1. The A31S134 has only the External Interrupts for PB[11:0], PC[7:0], PE[8:0], PF[4:2].

Figure 27. Configuration Map for External Interrupt Group 0/1/2/3



5.5.6 INTC_MSK: Interrupt Source Mask Register

The INTC_MSK register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_MSK =0x4000_1400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMSK31	IMSK30	IMSK29	IMSK28	IMSK27	IMSK26	IMSK25	IMSK24	IMSK23	IMSK22	IMSK21	IMSK20	IMSK19	IMSK18	IMSK17	IMSK16	IMSK15	IMSK14	IMSK13	IMSK12	IMSK11	IMSK10	IMSK9	IMSK8	IMSK7	IMSK6	IMSK5	IMSK4	IMSK3	IMSK2	IMSK1	IMSK0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	IMSKx	Interrupt Source Mask bit, x = 0 to 31.	
		0	Mask. The corresponding interrupt is disabled.
		1	Unmask

NOTE:

1. A mask interrupt source is not used for a wake-up source from “sleep” or ”deep sleep” mode.

Table 27. Corresponding Interrupts of IMSKx

Source Mask	Interrupt Source Name
IMSK0	LVI
IMSK1	WUT
IMSK2	WDT
IMSK3	EINT0
IMSK4	EINT1
IMSK5	EINT2
IMSK6	EINT3
IMSK7	TIMER10
IMSK8	TIMER11
IMSK9	TIMER12
IMSK10	I2C0
IMSK11	USART10
IMSK12	WT
IMSK13	Reserved
IMSK14	I2C1
IMSK15	TIMER20
IMSK16	TIMER21
IMSK17	USART11
IMSK18	ADC
IMSK19	UART0
IMSK20	UART1
IMSK21	TIMER13
IMSK22	TIMER14
IMSK23	TIMER15
IMSK24	DMACH[1:0]
IMSK25	I2C2
IMSK26	USART12
IMSK27	Reserved
IMSK28	Reserved
IMSK29	SPI0
IMSK30	SPI1
IMSK31	Reserved

5.5.7 Interrupt Register Map Summary

Table 28. Interrupt Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000	INTC_PnTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIG11	TRIG10	TRIG9	TRIG8	TRIG7	TRIG6	TRIG5	TRIG4	TRIG3	TRIG2	TRIG1	TRIG0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x1100	INTC_PnCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTCTL11[1:0]		INTCTL10[1:0]		INTCTL9[1:0]		INTCTL8[1:0]		INTCTL7[1:0]		INTCTL6[1:0]		INTCTL5[1:0]		INTCTL4[1:0]		INTCTL3[1:0]		INTCTL2[1:0]		INTCTL1[1:0]		INTCTL0[1:0]	
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1200	INTC_PnFLAG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLAG11	FLAG10	FLAG9	FLAG8	FLAG7	FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x1300	INTC_EINTnCONF1	CONF7[3:0]				CONF6[3:0]				CONF5[3:0]				CONF4[3:0]				CONF3[3:0]				CONF2[3:0]				CONF1[3:0]				CONF0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1310	INTC_EINTnCONF2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONF11[3:0]				CONF10[3:0]				CONF9[3:0]				CONF8[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1400	INTC_MSK	IMSK31	IMSK30	IMSK29	IMSK28	IMSK27	IMSK26	IMSK25	IMSK24	IMSK23	IMSK22	IMSK21	IMSK20	IMSK19	IMSK18	IMSK17	IMSK16	IMSK15	IMSK14	IMSK13	IMSK12	IMSK11	IMSK10	IMSK9	IMSK8	IMSK7	IMSK6	IMSK5	IMSK4	IMSK3	IMSK2	IMSK1	IMSK0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6. Embedded Flash Memory

6.1 Introduction

The flash memory controller is an interface controller for embedded flash memory. It manages data stored on the flash memory.

The flash memory of A31S134 has several key features as listed below:

- Code flash memory for 128 KB with write protection configure option
- 128 KB bulk erase
- 256 bytes page erase
- Flash Read Protection

6.2 Flash Memory Main Features

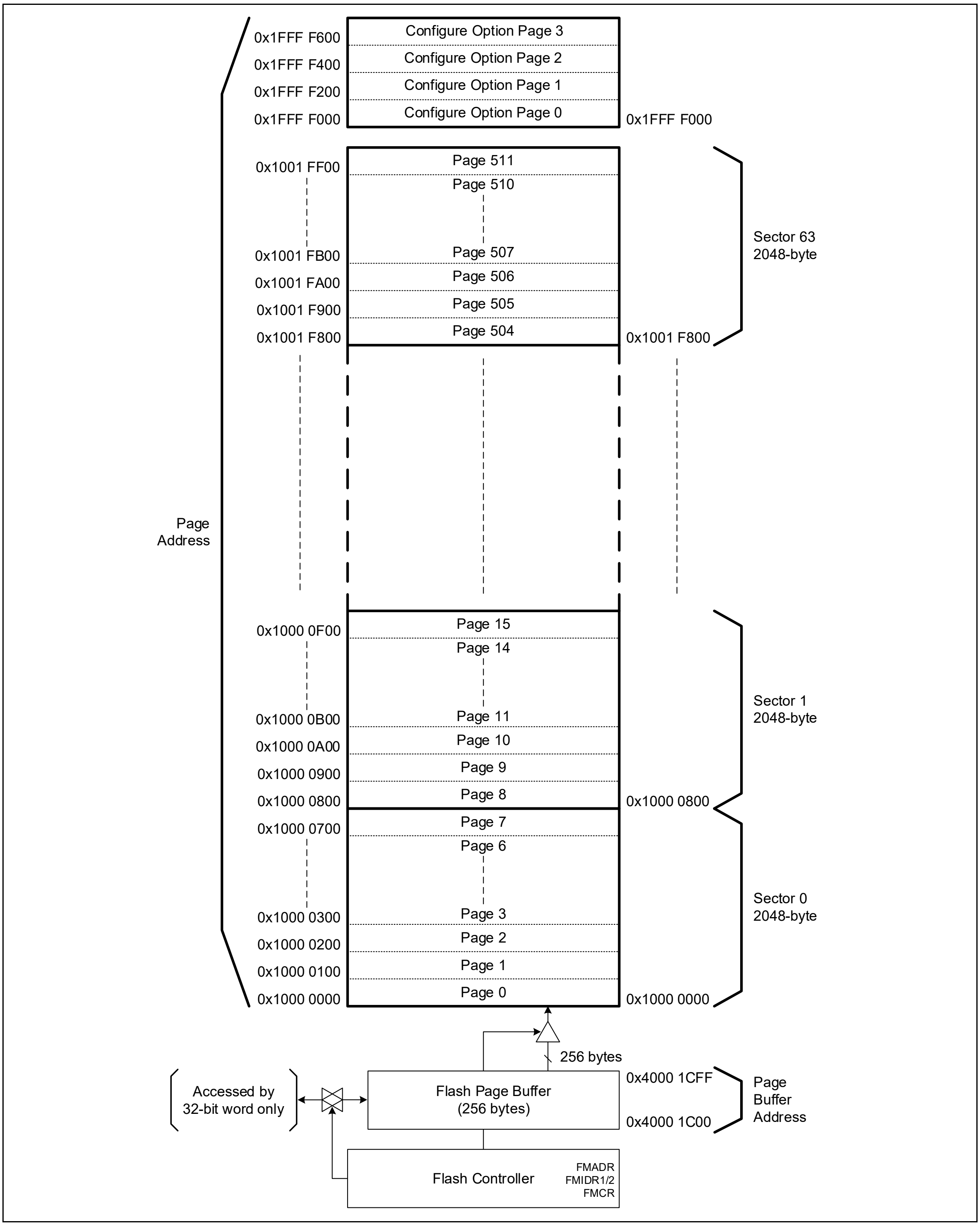
The A31S134 has built-in Flash memory with the following features:

- 128 KB flash memory
- 32-bit wide read data bus
- 256-byte sized page
- Page erase and bulk erase
- Programming in 256-byte units

Table 29. Flash Memory Features

Item	Description
Size	128 KB
Start Address	0x1000_0000
End Address	0x1001_FFFF
Page Size	256 bytes
Total Page Count	512 pages
PGM Unit	256 bytes
Erase Unit	256 bytes or bulk

Figure 28. Flash Memory Map (128 KB Code Flash)



6.2.1 Procedure For Flash Memory Operation

- The high frequency internal RC oscillator (HIRC) should be enabled by software for flash memory operation.
- The procedure will be cleared, the related registers will be reset, and a FMOPFLAG will be set if there is a wrong sequence.
- The address range is 0x10000000 to 0x17FFFFFF when "flash memory area" is selected.
- The address range is 0x1FFFF000 to 0x1FFFFFFF when "configure option area" is selected.
- If the CPU is in the flash memory, the CPU will be held during the flash memory operation is activated.
- The "configure option page 0" won't be erased at flash bulk erase mode.
- The "configure option page 1/2/3" can be erased at flash bulk erase mode if the CNFxBEN has correct values.
- The CPU should not be in the flash memory area on flash bulk erase mode.
- A write to the flash related register is ignored during flash operation.
- An NMI source should not be selected during the flash memory operation is activated.
- The LVR should be enabled while the flash memory operation is activated (recommendation: 2.28 V over).
- The global interrupt should be disabled.
- The CPU should not enter sleep and deep sleep mode during flash erase/write mode.
- It should not be entered into the flash erase/write modes during data flash program. So, before the flash erase/write operation, the data flash busy bit (DFMOPFLAG) should be checked whether the bit is '1' . If the bit is '1' , do not start flash program.

6.2.1.1 Page Erase Procedure

1. Write 0x5FFFFFFF to the FMC_ADR register during the register is equal to 0x5FFFFF80.
2. Write 0x08192A3B to the FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.
3. Write 0x4C5D6E7F to the FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.
4. Write 0x6C930001 to the FMC_CR register for page buffer reset during the FMC_ADR register is equal to 0x5FFFFFFF.
5. Clear page buffer (256 bytes) by writing 0xFFFFFFFF repeatedly during the FMC_ADR register is 0x5FFFFFFF.
6. Write a page address to the FMC_ADR register.
7. Read and check the FMC_IDR1 and FMC_IDR2 registers in turn.
8. Write 0x6C93A402 (flash memory area) or 0x6C933802 (configure option area) to the

FMC_CR register.

9. Check whether the FMBUSY bit is ‘0’.

10. Verify the erased page of flash memory.

6.2.1.2 Page Write Procedure

1. Write 0x5FFFFFFF to the FMC_ADR register during the register is equal to 0x5FFFFF80.

2. Write 0x08192A3B to the FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.

3. Write 0x4C5D6E7F to the FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.

4. Write 0x6C930001 to the FMC_CR register for page buffer reset during the FMC_ADR register is equal to 0x5FFFFFFF.

5. Write data to page buffer (any byte) during the FMC_ADR register is equal to 0x5FFFFFFF.

6. Write a page address to FMC_ADR register.

7. Read and check the FMC_IDR1 and FMC_IDR2 registers in turn.

8. Write 0x6C93A404 (flash memory area) or 0x6C933804 (configure option area) to the FMC_CR register.

9. Check whether the FMBUSY bit is ‘0’.

10. Verify the written page of flash memory.

6.2.1.3 Flash Bulk Erase Procedure

1. Write 0x5FFFFFFF to the FMC_ADR register during the register is equal to 0x5FFFFF80.

2. Write 0x08192A3B to the FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.

3. Write 0x4C5D6E7F to the FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.

4. Write 0x6C930001 to the FMC_CR register for page buffer reset during the FMC_ADR register is equal to 0x5FFFFFFF.

5. Write the value 0x5F9A30D7 to the FMC_ADR register.

6. Read and check the FMC_IDR1 and FMC_IDR2 register in turn.

7. Write 0x6C93A408 to the FMC_CR register.

8. Check whether the FMBUSY bit is ‘0’.

9. Verify all pages of the flash memory.

6.2.1.4 Flash Bulk Erase Procedure Including Configure Option Page

1. Write 0x5FFFFFFF to the FMC_ADR register during the register is equal to 0x5FFFFF80.
2. Write 0x08192A3B to the FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.
3. Write 0x4C5D6E7F to the FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.
4. Write 0x6C930001 to the FMC_CR register for page buffer reset during the FMC_ADR register is equal to 0x5FFFFFFF.
5. Write the value 0xC1BE0VVV to FMC_BCR register. If V==5, the corresponding option page will be erased.
6. Write the value 0x5F9A30D7 to the FMC_ADR register.
7. Read and check the FMC_IDR1 and FMC_IDR2 register in turn.
8. Write 0x6C93A408 to FMC_CR register.
9. Check whether the FMBUSY bit is ‘0’.
10. Verify all pages of the flash memory.

6.3 Flash Memory Controller Registers

The base address and register map of the Flash Memory Controller (FMC) are described in the following tables.

Table 30. Base Address of Flash Memory Controller

Name	Base Address
FMC	0x4000_1B00

Table 31. FMC Register Map

Name	Offset	Type	Description	Reset Value	Reference
FMC_ADR	0x0000	RW	Flash Memory Address Register	0x5FFFFF80	6.3.1
FMC_IDR1	0x0004	RW	Flash Memory Identification Register 1	0x00000000	6.3.2
FMC_IDR2	0x0008	RW	Flash Memory Identification Register 2	0x00000000	6.3.3
FMC_CR	0x000C	RW	Flash Memory Control Register	0x00000000	6.3.4
FMC_BCR	0x0010	RW	Flash Memory Configure Area Bulk Erase Control Register	0x00000000	6.3.5
FMC_ERFLAG	0x0014	RW	Flash Memory Error Flag	0x00000000	6.3.6
FMC_PAGEBUF	0x0100- 0x01FF	WO	Flash Memory Page Buffer Area	0x00000000	-

6.3.1 FMC_ADR: Flash Memory Address Register

The FMC_ADR register is used to remember the internal Flash memory address. This register is 32 bits wide.

FMC_ADR=0x4000_1B00

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ADDR[31:0]			
0x5FFF_FF80			
RW			

31 0	ADDR[31:0]	Flash Memory Address Pointer. This register is reset to 0x5FFFFF80 immediately after a single operation.

NOTE:

1. The LSB-byte of the target Flash address is always considered to “0x00”.

6.3.2 FMC_IDR1: Flash Memory Identification Register 1

The FMC_IDR1 register is an internal Flash memory identification register for Flash mode. This register is 32 bits wide.

FMC_IDR1=0x4000_1B04

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ID1[31:0]			
0x0000_0000			
RW			

31 0	ID1[31:0]	Flash Memory Identification 1	
		0x08192A3B	Identification value for a Flash mode
		Others	No identification value

6.3.3 FMC_IDR2: Flash Memory Identification Register 2

The FMC_IDR2 register is an internal Flash memory identification register for Flash mode. This register is 32 bits wide.

FMC_IDR2=0x4000_1B08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID2[31:0]																															
0x0000_0000																															
RW																															

31 0	ID2[31:0]	Flash Memory Identification 2	
		0x4C5D6E7F	Identification value for a Flash mode
		Others	No identification value

NOTES:

1. The FMC_IDR1/2 registers are automatically cleared to logic 0x00000000 immediately after one time operation except “Flash page buffer reset mode”.
2. The FMC_IDR1/2 registers should be written with correct values in turn.
3. If incorrect values are written to the FMC_IDR1/2 registers, the registers are cleared to logic 0x00000000.

6.3.4 FMC_CR: Flash Memory Control Register

The FMC_CR register is an internal Flash memory control register. This register is a 32-bit size.

FMC_CR=0x4000_1B0C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4	3	2	1	0
WTIDKY[15:0]	FMKEY[7:0]	FMBUSY	Reserved	FMOD[3:0]			
0x0000	0x00	0	-	0	0	0	0
WO	RW	RO	-	RW	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key.	
		When writing, write 0x6C93 to these bits, or else writing is ignored.	
15 8	FMKEY[7:0]	Flash Memory Operation Area Selection.	
		0x00	Selects no area but for Flash page buffer reset mode.
		0x38	Selects “configure option area” for Flash memory erase/write.
		0xA4	Selects “Flash memory area” for Flash memory erase/write.
		Others	Not allowed. FMOPFLAG will be set.
7	FMBUSY	Flash Memory Operation Mode Busy.	
		0	No effect.
		1	Busy.
3 0	FMOD[3:0]	Flash Memory Operation Mode Selection.	
		0001	“Flash page buffer reset mode” and start regardless of the Flash operation rule. (Clear all 256bytes page buffer to 0xFFFFFFFF)
		0010	“Flash page erase mode” and start when the Flash operation rule is satisfied.
		0100	“Flash page write mode” and start when the Flash operation rule is satisfied.
		1000	“Flash bulk erase mode” and start when the Flash operation rule is satisfied.
		Others	Not allowed. FMOPFLAG will be set.

NOTES:

1. During a Flash memory operation mode, all interrupts are on disable regardless of enable bits.
2. The FMKEY[7:0] and FMOD[3:0] bits are automatically cleared to logic “0x00” immediately after a single operation.
3. In Read Protection level 1/2, the Data Flash memory must be erased before performing Flash bulk erase.

6.3.5 FMC_BCR: Flash Memory Configure Area Bulk Erase Control Register

The FMC_BCR register is used to permit bulk erase. This register is a 32-bit size.

FMC_BCR=0x4000_1B10

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
WTIDKY[15:0]	Reserved	CNF3BEN[3:0]	CNF2BEN[3:0]	CNF1BEN[3:0]
0x0000	-	0000	0000	0000
WO	-	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key.	
		When writing, write 0xC1BE to these bits, or else writing is ignored.	
11 8	CNF3BEN[3:0]	Configure Option Page 3 Bulk Erase Enable.	
		0x5	Permit “Configure Option Page 3” erase at bulk erase
		Others	Protect “Configure Option Page 3” erase at bulk erase
7 4	CNF2BEN[3:0]	Configure Option Page 2 Bulk Erase Enable.	
		0x5	Permit “Configure Option Page 2” erase at bulk erase
		Others	Protect “Configure Option Page 2” erase at bulk erase
3 0	CNF1BEN[3:0]	Configure Option Page 1 Bulk Erase Enable.	
		0x5	Permit “Configure Option Page 1” erase at bulk erase
		Others	Protect “Configure Option Page 1” erase at bulk erase

NOTE:

1. This register is automatically cleared to logic “0x00” immediately after one time operation.

6.3.6 FMC_ERFLAG: Flash Memory Error Flag Register

The FMC_ERFLAG register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

FMC_ERFLAG=0x4000_1B14

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="30">Reserved</td><td>INSTFLAG</td><td>FMOPFLAG</td></tr>
<tr><td colspan="30">-</td><td>0</td><td>0</td></tr>
<tr><td colspan="30">.</td><td>RW</td><td>RW</td></tr>
</table>

1	INSTFLAG	Don’t care	
0	FMOPFLAG	Error bit of Flash Memory Operation Procedure. This bit is set to logic 1 if there is a wrong procedure for Flash memory operation.	
		0	No wrong procedure.
		1	A wrong procedure occurred. The bit is cleared to ‘0’ when ‘1’ is written.

6.3.7 FMC Register Map Summary

Table 32. FMC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FMC_ADR	ADDR[31:0]																															
	Reset value	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0x04	FMC_IDR1	ID1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	FMC_IDR2	ID2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	FMC_CR	WTIDKY[15:0]																FMKEY[7:0]								FMBUSY	Res	Res	Res	FMOD[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0
0x10	FMC_BCR	WTIDKY[15:0]																Res	Res	Res	Res	CNF3BEN				CNF2BEN				CNF1BEN			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x14	FMC_ERFLAG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	INSTFLAG	FMOPFLAG
	Reset value																															0	0
0x100	FMC_PAGEBUF	PGBUF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x104	FMC_PAGEBUF	PGBUF[63:32]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1FC	FMC_PAGEBUF	PGBUF[2047:2016]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4 Data Flash Memory Main Features

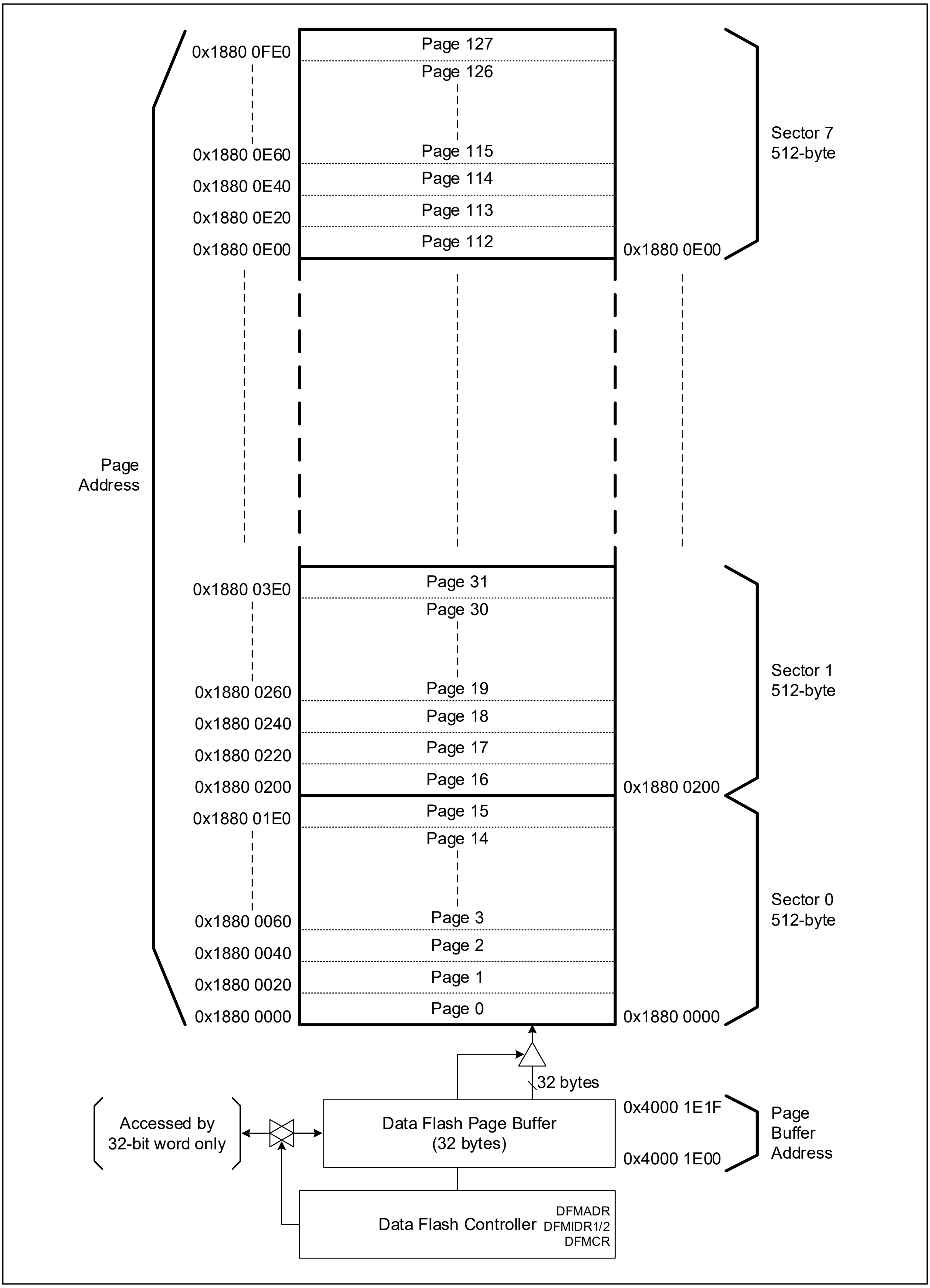
The A31S134 has built-in data flash memory with the following features:

- 4 KB data flash memory
- 32-bit wide read data bus
- 32-byte sized page
- Page erase and bulk erase
- Programming in 32-byte units

Table 33. Data Flash Memory Features

Item	Description
Size	4 KB
Start Address	0x1880_0000
End Address	0x1880_0FFF
Page Size	32 bytes
Total Page Count	128 pages
Program Unit	32 bytes
Erase Unit	32 bytes or bulk

Figure 29. Data Flash Memory Map (4 KB Data Flash)



6.4.1 Procedure For Data Flash Memory Operation

- The high frequency internal RC oscillator (HIRC) should be enabled by software for data flash memory operation.
- The procedure will be cleared, the related registers will be reset, and a DFMOPFLAG will be set if there is a wrong sequence.
- The address range is 0x18800000 to 0x188FFFFF when “data flash memory area” is selected.
- A write to the data flash related register is ignored during data flash operation.
- The LVR should be enabled while the data flash memory operation is activated (Recommend: 2.28 V over).
- The CPU should not enter sleep and deep sleep mode during data flash erase/write mode.
- It should not be entered into the data flash erase/write modes during flash program. So, before the data flash erase/write operation, the flash busy bit (FMOPFLAG) should be checked whether the bit is ‘1’ or not. If the bit is ‘1’, do not start data flash program.

6.4.1.1 Page Erase Procedure

1. Write 0x59999990 to the DFMC_DADR register while the register is equal to 0x5FFF9990.
2. Write 0xA9C46E91 to the DFMC_DIDR1 register while the DFMC_DADR register is equal to 0x59999990.
3. Write 0x78F5B3D2 to the DFMC_DIDR2 register during the DFMC_DADR register is equal to 0x59999990.
4. Write 0xB69C0001 to the DFMC_DCR register for page buffer reset during the DFMC_DADR register is equal to 0x59999990.
5. Clear page buffer (32 bytes) by writing 0xFFFFFFFF repeatedly during the DFMC_DADR register is 0x59999990.
6. Write a page address to the DFMC_DADR register.
7. Read and check the DFMC_DIDR1 and DFMC_DIDR2 registers in turn.
8. Write 0xB69CA402 (data flash memory area) to the DFMC_DCR register.
9. Check whether the DFMBUSY bit is ‘0’.
10. Verify the erased page of data flash memory.

6.4.1.2 Page Write Procedure

1. Write 0x59999990 to the DFMC_DADR register while the register is equal to 0x5FFF9990.
2. Write 0xA9C46E91 to the DFMC_DIDR1 register while the DFMC_DADR register is equal to 0x59999990.
3. Write 0x78F5B3D2 to the DFMC_DIDR2 register during the DFMC_DADR register is equal to 0x59999990.
4. Write 0xB69C0001 to the DFMC_DCR register for page buffer reset during the DFMC_DADR

register is equal to 0x59999990.

5. Write data to data flash page buffer (any byte) during the DFMC_DADR register is equal to 0x59999990.
6. Write a page address to the DFMC_DADR register.
7. Read and check the DFMC_DIDR1 and DFMC_DIDR2 registers in turn.
8. Write 0xB69CA404 (data flash memory area) to the DFMC_DCR register.
9. Check whether the DFMBUSY bit is ‘0’.
10. Verify the written page of data flash memory.

6.4.1.3 Data Flash Bulk Erase Procedure

1. Write 0x59999990 to the DFMC_DADR register while the register is equal to 0x5FFF9990.
2. Write 0xA9C46E91 to the DFMC_DIDR1 register while the DFMC_DADR register is equal to 0x59999990.
3. Write 0x78F5B3D2 to the DFMC_DIDR2 register during the DFMC_DADR register is equal to 0x59999990.
4. Write 0xB69C0001 to the DFMC_DCR register for page buffer reset during the DFMC_DADR register is equal to 0x59999990.
5. Write the value 0x4BC27F54 to DFMC_DADR register.
6. Read and check the DFMC_DIDR1 and DFMC_DIDR2 register in turn.
7. Write 0xB69CA408 to the DFMC_DCR register.
8. Check whether the DFMBUSY bit is ‘0’.
9. Verify all the pages of flash memory.

6.5 Data Flash Memory Controller Registers

The base address and register map of the Data Flash Memory Controller (DFMC) are described in the following tables.

Table 34. Base Address of Data Flash Memory Controller

Name	Base Address
DFMC	0x4000_1D00

Table 35. DFMC Register Map

Name	Offset	Type	Description	Reset Value	Reference
DFMC_DADR	0x0000	RW	Data Flash Memory Address Register	0x5FFF9990	6.5.1
DFMC_DIDR1	0x0004	RW	Data Flash Memory Identification Register 1	0x00000000	6.5.2
DFMC_DIDR2	0x0008	RW	Data Flash Memory Identification Register 2	0x00000000	6.5.3
DFMC_DCR	0x000C	RW	Data Flash Memory Control Register	0x00000000	6.5.4
DFMC_DERFLAG	0x0014	RW	Data Flash Memory Error Flag	0x00000000	6.5.5
DFMC_DPAGEBUF	0x0100- 0x011F	WO	Data Flash Memory Page Buffer Area	0x00000000	-

6.5.1 DFMC_DADR: Data Flash Memory Address Register

The DFMC_DADR register is used to remember the internal data flash memory address. This register is 32 bits wide.

DFMC_DADR=0x4000_1D00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[31:0]																															
0x5FFF_9990																															
RW																															

31 0	ADDR[31:0]	Data Flash Memory Address Pointer. This register is reset to 0x5FFF9990 immediately after a single operation.

NOTE:

1. The LSB-5bits of the target flash memory address is always considered to “00000”.

6.5.2 DFMC_DIDR1: Data Flash Memory Identification Register 1

The DFMC_DIDR1 register is an internal data flash memory identification register for data flash mode. This register is 32 bits wide.

DFMC_DIDR1=0x4000_1D04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID1[31:0]																															
0x0000_0000																															
RW																															

31 0	ID1[31:0]	Data Flash Memory Identification 1	
		0xA9C46E91	Identification value for a Data Flash mode
		Others	No identification value

6.5.3 DFMC_DIDR2: Data Flash Memory Identification Register 2

The DFMC_DIDR2 register is an internal data flash memory identification register for data flash mode. This register is 32 bits wide.

DFMC_DIDR2=0x4000_1D08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID2[31:0]																															
0x0000_0000																															
RW																															

31 0	ID2[31:0]	Flash Memory Identification 2	
		0x78F5B3D2	Identification value for a Data Flash mode
		Others	No identification value

NOTES:

1. The DFMC_DIDR1/2 registers are automatically cleared to logic 0x00000000 immediately after one time operation except “Data Flash page buffer reset mode”.
2. The DFMC_DIDR1/2 registers should be written with correct values in turn.
3. If incorrect values are written to the DFMC_DIDR1/2 registers, the registers are cleared to logic 0x00000000.

6.5.4 DFMC_DCR: Data Flash Memory Control Register

The DFMC_DCR register is an internal data flash memory control register. This register is a 32-bit size.

DFMC_DCR=0x4000_1D0C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4	3	2	1	0
WTIDKY[15:0]	DFMKEY[7:0]	DFMBUSY	Reserved	DFMOD[3:0]			
0x0000	0x00	0	-	0	0	0	0
WO	RW	RO	-	RW	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key.	
		When writing, write 0xB69C to these bits, or else writing is ignored.	
15 8	DFMKEY[7:0]	Data Flash Memory Operation Area Selection.	
		0x00	Selects no area but for Data Flash page buffer reset mode.
		0xA4	Selects “Data Flash memory area” for Data Flash memory erase/write.
		Others	Not allowed. DFMOPFLAG will be set.
7	DFMBUSY	Data Flash Memory Operation Mode Busy.	
		0	No effect.
		1	Busy.
3 0	DFMOD[3:0]	Data Flash Memory Operation Mode Selection.	
		0001	“Data Flash page buffer reset mode” and start regardless of the operation rule. (Clear all 32bytes page buffer to 0xFF)
		0010	“Data Flash page erase mode” and start when the operation rule is satisfied.
		0100	“Data Flash page write mode” and start when the operation rule is satisfied.
		1000	“Data Flash bulk erase mode” and start when the operation rule is satisfied.
		Others	Not allowed. DFMOPFLAG will be set.

NOTE:

1. The DFMKEY[7:0] and DFMOD[3:0] bits are automatically cleared to logic “0x00” immediately after a single operation.

6.5.5 DFMC_DERFLAG: Data Flash Memory Error Flag Register

The DFMC_DERFLAG register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

DFMC_DERFLAG=0x4000_1D14

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	0
Reserved	DFMOPFLAG
-	0
.	RW

0	DFMOPFLAG	Error bit of data flash memory operation procedure. This bit is set to logic 1 if there is a wrong procedure for data flash memory operation.	
		0	No wrong procedure.
		1	A wrong procedure occurred. The bit is cleared to '0' when '1' is written.

6.5.6 DFMC Register Map Summary

Table 36. DFMC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DFMC_DADR	ADDR[31:0]																															
	Reset value	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0
0x04	DFMC_DIDR1	ID1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	DFMC_DIDR2	ID2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DFMC_DCR	WTIDKY[15:0]																DFMKEY[7:0]								DFMBUSY	Res.	Res.	Res.	DFMOD[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0
0x14	DFMC_DERFLAG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFMOPFLAG
	Reset value																																0
0x100	DFMC_DPAGEBUF	PGBUF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x104	DFMC_DPAGEBUF	PGBUF[63:32]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x11C	DFMC_DPAGEBUF	PGBUF[255:224]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

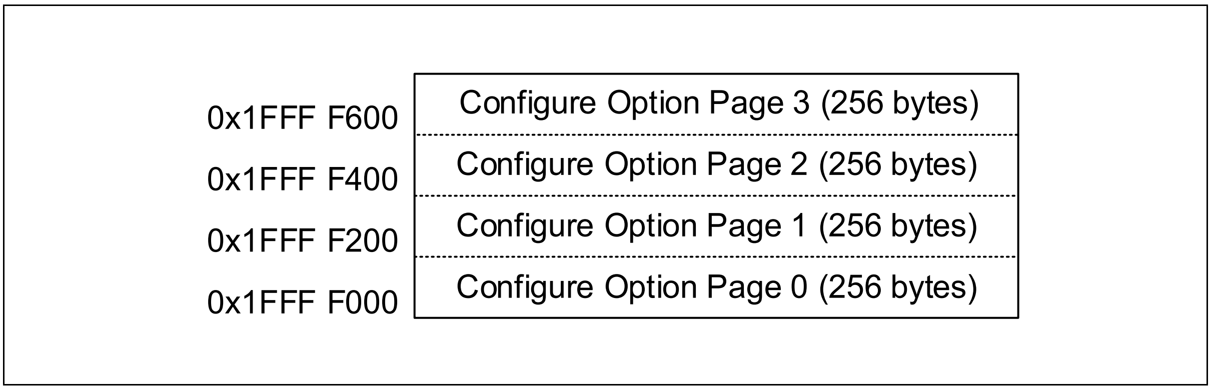
6.6 Configure Option Area

The configuration option area of the A31S134 is used for system related trimming values, user option, and user data. The configure option area consists of four pages in the flash memory, which can be erased and written by the flash memory controller. This area can be read by any instruction.

Four pages of the configure option area are listed below:

- Page 0: System related trimming values and 128-bit unique device ID registers
- Page 1: User option for Read Protection, watchdog timer, and LVR voltage level configurations.
- Page 2: User data 0 area
- Page 3: User data 1 area

Figure 30. Configure Option Area Structure


0x1FFF F600
Configure Option Page 3 (256 bytes)
0x1FFF F400
Configure Option Page 2 (256 bytes)
0x1FFF F200
Configure Option Page 1 (256 bytes)
0x1FFF F000
Configure Option Page 0 (256 bytes)

6.7 Configure Option Pages

The base address of the configure option area ranges from 0x1FFF_F000 to 0x1FFF_F600.

The area map is shown in Table 37.

Table 37. Configure Option Area Map

Page	Name	Address	Description
0	-	0x1FFF_F000 to 0x1FFF_F047 0x1FFF_F060 to 0x1FFF_F07F	System Trimming Values
	CONF_MF1CNFIG	0x1FFF_F050	Manufacture Information 1 for 128-bit unique ID
	CONF_MF2CNFIG	0x1FFF_F054	Manufacture Information 2 for 128-bit unique ID
	CONF_MF3CNFIG	0x1FFF_F058	Manufacture Information 3 for 128-bit unique ID
	CONF_MF4CNFIG	0x1FFF_F05C	Manufacture Information 4 for 128-bit unique ID
1	CONF_RPCNFIG	0x1FFF_F200	Configuration for Read Protection
	CONF_WDTCNFIG	0x1FFF_F20C	Configuration for Watch-Dog Timer
	CONF_LVRCNFIG	0x1FFF_F210	Configuration for Low Voltage Reset
	CONF_CNFIGWTP1	0x1FFF_F214	Erase/Write Protection for configure option page 1/2/3
	CONF_FMWTP1	0x1FFF_F240	Erase/Write Protection 1 for Flash Memory
	CONF_FMWTP2	0x1FFF_F244	Erase/Write Protection 2 for Flash Memory
	CONF_DFMWTP1	0x1FFF_F270	Erase/Write Protection 1 for Data Flash Memory
2	-	0x1FFF_F400 to 0x1FFF_F4FF	User Data Area 0
3	-	0x1FFF_F600 to 0x1FFF_F6FF	User Data Area 1

6.7.1 CONF_MF1CNFIG: Configuration for Manufacture Information 1

The Configuration for Manufacture Information 1 is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF1CNFIG=0x1FFF_F050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XYCDN[31:0]																															
Reset Value: Unknown																															
RW																															

31 0	XYCDN[31:0]	X and Y Coordinates.

6.7.2 CONF_MF2CNFIG: Configuration for Manufacture Information 2

The Configuration for Manufacture Information 2 is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF2CNFIG=0x1FFF_F054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOTNO[23:0]																								WAFNO[7:0]							
Reset Value: Unknown																															
RW																								RW							

31 8	LOTNO[23:0]	Lot Number
7 0	WAFNO[7:0]	Wafer Number

6.7.3 CONF_MF3CNFIG: Configuration for Manufacture Information 3

The Configuration for Manufacture Information 3 is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF3CNFIG=0x1FFF_F058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOTNO[55:24]																															
Reset Value: Unknown																															
RW																															

31 0	LOTNO[55:24]	Lot Number

6.7.4 CONF_MF4CNFIG: Configuration for Manufacture Information 4

The Configuration for Manufacture Information 4 is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF4CNFIG=0x1FFF_F05C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOTNO[87:56]																															
Reset Value: Unknown																															
RW																															

31 0	LOTNO[87:56]	Lot Number

6.7.5 CONF_RPCNFIG: Configuration for Read Protection

The configuration for the Flash Memory Read Protection is 32-bit flash memory. This register is a 32-bit size.

CONF_RPCNFIG=0x1FFF_F200

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2	1	0
WTIDKY[27:0]	Reserved	READP[1:0]	
0xFFFFFFF	-	1	1
RW	-	RW	RW

31 4	WTIDKY[27:0]	Write Identification Key These bits are the write key for "Read Protection". So, The WTIDKY[27:0] should be kept with the 0x69C8A27. Otherwise, the Read Protection will be on level 2.	
1 0	READP[1:0]	Read Protection for Flash Memory and Data Flash Memory Area.	
		11	Read Protection level 0, No restriction for read/erase/write.
		10	Read Protection level 1, Not readable/erasable/writable by debugger Bulk erasable only by debugger Readable/erasable/writable by "Instruction from Flash Memory and RAM"
		0x	Read Protection level 2, Where x is don't care Not readable/erasable/writable by "Debug"/"Instruction from RAM" Bulk erasable only by "Instruction from RAM"/"Debug" Readable/erasable/writable by "Instruction from Flash Memory"

NOTES:

1. The Read Protection level can be changed from a lower level to higher level only.
2. The "Configure Option Page 1" cannot be erased by "Debug" unit on "Read Protection level 1/2" and by "Instruction from RAM" on "Read Protection level 2.
3. The configure option area may be read even if the "Read Protection" is on level 1 and 2.
4. A page unit erase/write except a bulk erase isn't executable by "Instruction from RAM" regardless of the CONF_FMWTP1/2 and CONF_DFMWTP1 registers on Read Protection level 2.
5. A page unit erase/write except a bulk erase isn't executable by "Debug" regardless of the CONF_FMWTP1/2 and CONF_DFMWTP1 registers on Read Protection level 1/2.
6. The Read Protection level will be '0' on operation after bulk erase.

6.7.6 CONF_WDTCNFIG: Configuration for Watchdog Timer

The configuration for the Watchdog Timer is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_WDTCNFIG=0x1FFF_F20C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4	3	2	1	0
Reserved	WRCMF[11:0]	Reserved	WCLKMF	WRSTMF	WCNTMF
0xFFFF	0xFFF	-	1	1	1
RW	RW	-	RW	RW	RW

15 4	WRCMF[11:0]	Watchdog Timer RC Oscillator Master Configuration	
		0x96D	The WDTRC oscillation is decided by the WDTRCEN of SCU_CLKSRCR register.
		0x2A7	Master enable WDTRC but disabled at DEEP SLEEP mode.
		Others	Master enable WDTRC.
		NOTE:	
		1.	If the WDTRC is selected for MCLK by SCU_SCCR register when the bits are not 0x96D, the CPU cannot wake up at DEEP SLEEP mode. So, only SLEEP mode on the above case should be used for power down.
2	WCLKMF	Watchdog Timer Clock Selection Master Configuration	
		0	Watchdog Timer clock is selected by the WDTCLK of SCU_PPCLKSR register.
		1	Master selection WDTRC for Watchdog Timer clock
1	WRSTMF	Watchdog Timer Reset Enable Master Configuration	
		0	Master enable WDT reset
		1	Disable/Enable of WDT reset is decided by the RSTEN[5:0] of WDT_CR register.
0	WCNTMF	Watchdog Timer Counter Enable Master Configuration	
		0	Master enable WDT counter
		1	Disable/Enable WDT counter is decided by the CNTEN[5:0] of WDT_CR register.

6.7.7 CONF_LVRCNFIG: Configuration for Low Voltage Reset

The configuration for the Low Voltage Reset is 32-bit flash memory. This register is a 32-bit size.

CONF_LVRCNFIG=0x1FFF_F210

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1	0
WTIDKY[15:0]	LVRENM[7:0]	Reserved	LVRVS[3:0]			
0xFFFF	0xFF	-	1	1	1	1
RW	RW	-	RW	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key: 0x9D58	
		These bits are the write key for "LVR controller".	
15 8	LVRENM[7:0]	LVR Reset Operation Control Master Configuration	
		0xAA	LVR operation is decided by the LVREN of SCU_LVRCR register
		Others	Master enable LVR operation
3 0	LVRVS[3:0]	LVR Voltage Selection.	
		1111	1.62 V
		1011	2.00 V
		1010	2.13 V
		1001	2.28 V
		1000	2.46 V
		0111	2.67 V
		0110	3.04 V
		0101	3.20 V
		0100	3.55 V
		0011	3.75 V
		0010	3.99 V
		0001	4.25 V
		0000	4.55 V

6.7.8 CONF_CNFIGWTP1: Erase/Write Protection for Configure Option Page 1/2/3

The Erase/Write Protection for Configure Option Page is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_CNFIGWTP1=0x1FFF_F214

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																													CP3WP	CP2WP	CP1WP
-																													1	1	1
.																													RW	RW	RW

2	CP3WP	Configure Option Page 3 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction)
		1	Disable protection (Erasable/writable by instruction)
1	CP2WP	Configure Option Page 2 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction)
		1	Disable protection (Erasable/writable by instruction)
0	CP1WP	Configure Option Page 1 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction)
		1	Disable protection (Erasable/writable by instruction)

NOTE:

1. The Configure Option Page which is protected cannot be erased by page unit.

6.7.9 CONF_FMWTP1: Erase/Write Protection 1 for Flash Memory

The Erase/Write Protection 1 for flash memory is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_FMWTP1=0x1FFF_F240

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWTP31	SWTP30	SWTP29	SWTP28	SWTP27	SWTP26	SWTP25	SWTP24	SWTP23	SWTP22	SWTP21	SWTP20	SWTP19	SWTP18	SWTP17	SWTP16	SWTP15	SWTP14	SWTP13	SWTP12	SWTP11	SWTP10	SWTP9	SWTP8	SWTP7	SWTP6	SWTP5	SWTP4	SWTP3	SWTP2	SWTP1	SWTP0
0xFFFFFFFF																															
RW																															

n	SWTPn	Flash Memory Erase/Write Protection bits, n: 0 to 31 (sector 0 to sector 31)	
		0	Protect “Flash memory sector n erase/write”
		1	Permit “Flash memory sector n erase/write”

6.7.10 CONF_FMWTP2: Erase/Write Protection 2 for Flash Memory

The Erase/Write Protection 2 for flash memory is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_FMWTP2=0x1FFF_F244

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWTP63	SWTP62	SWTP61	SWTP60	SWTP59	SWTP58	SWTP57	SWTP56	SWTP55	SWTP54	SWTP53	SWTP52	SWTP51	SWTP50	SWTP49	SWTP48	SWTP47	SWTP46	SWTP45	SWTP44	SWTP43	SWTP42	SWTP41	SWTP40	SWTP39	SWTP38	SWTP37	SWTP36	SWTP35	SWTP34	SWTP33	SWTP32
0xFFFFFFFF																															
RW																															

n-32	SWTPn	Flash Memory Erase/Write Protection bits, n: 32 to 63 (Sector 32 to Sector 63)	
		0	Protect “Flash memory sector n erase/write”
		1	Permit “Flash memory sector n erase/write”

6.7.11 CONF_DFMWTP1: Erase/Write Protection 1 for Data Flash Memory

The Erase/Write Protection 1 for data flash memory is 32-bit flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_DFMWTP1=0x1FFF_F270

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								DSWTP7	DSWTP6	DSWTP5	DSWTP4	DSWTP3	DSWTP2	DSWTP1	DSWTP0
-																								0xFF							
.																								RW							

n	DSWTPn	Data Flash Memory Erase/Write Protection bits, n: 0 to 7 (Sector 0 to Sector 7)
		0 Protect "Data Flash memory sector n erase/write"
		1 Permit "Data Flash memory sector n erase/write"

Table 38. Configure Option Page 0 Configuration Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50	CONF_MF1CNFIG	XYCDN[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x54	CONF_MF2CNFIG	LOTNO[23:0]																								WAFNO[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x58	CONF_MF3CNFIG	LOTNO[55:24]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x5C	CONF_MF4CNFIG	LOTNO[87:56]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Table 39. Configure Option Page 1 Configuration Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CONF_RPCNFIG	WTIDKY[27:0]																												Res	Res	READP[1:0]	
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			1	1
0x0C	CONF_WDTCNFIG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WRCMF[11:0]												Res	WCLKMF	WRSTMF	WCNTMF
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1
0x10	CONF_LVRCNFIG	WTIDKY[15:0]																LVRENM[7:0]								Res	Res	Res	Res	LVRVS[3:0]			
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					1	1	1	1
0x14	CONF_CNFIGWTP1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CP3WP	CP2WP	CP1WP
	Reset value																														1	1	1
0x40	CONF_FMWTP1	SWTP31	SWTP30	SWTP29	SWTP28	SWTP27	SWTP26	SWTP25	SWTP24	SWTP23	SWTP22	SWTP21	SWTP20	SWTP19	SWTP18	SWTP17	SWTP16	SWTP15	SWTP14	SWTP13	SWTP12	SWTP11	SWTP10	SWTP9	SWTP8	SWTP7	SWTP6	SWTP5	SWTP4	SWTP3	SWTP2	SWTP1	SWTP0
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x44	CONF_FMWTP2	SWTP63	SWTP62	SWTP61	SWTP60	SWTP59	SWTP58	SWTP57	SWTP56	SWTP55	SWTP54	SWTP53	SWTP52	SWTP51	SWTP50	SWTP49	SWTP48	SWTP47	SWTP46	SWTP45	SWTP44	SWTP43	SWTP42	SWTP41	SWTP40	SWTP39	SWTP38	SWTP37	SWTP36	SWTP35	SWTP34	SWTP33	SWTP32
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x70	CONF_DFMWTP1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DSWTP7	DSWTP6	DSWTP5	DSWTP4	DSWTP3	DSWTP2	DSWTP1	DSWTP0
	Reset value																									1	1	1	1	1	1	1	1

7. Direct Memory Access Controller (DMA)

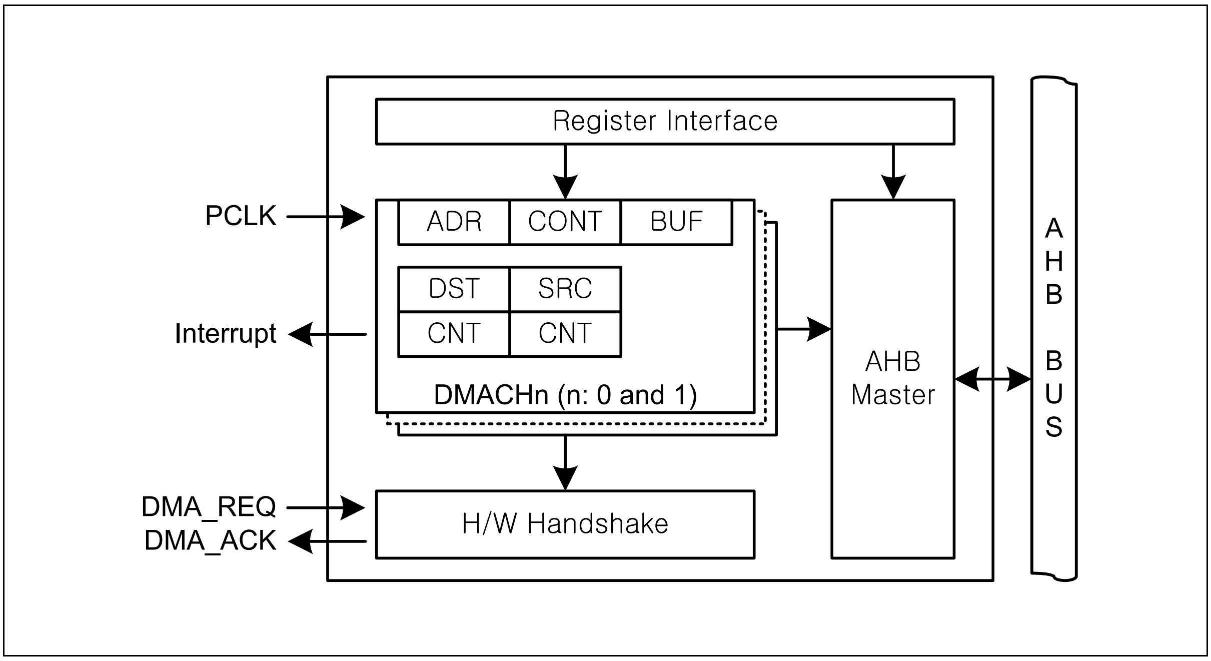
7.1 DMA Introduction

The Direct Memory Access (DMA) controller is used for high-speed data transfers between peripherals and memories. DMA enables quick data transfers having memory to peripheral copying or moving of data peripheral to memory.

- Two channels
- 8-/ 16-/ 32-bit data transfers supported
- Memory-to-peripheral transmission
- Peripheral-to-memory transmission
- Various buffers with the same size supported
- DMA transfers are triggered through peripheral interrupts

Figure 31 shows a DMA block diagram.

Figure 31. DMA Controller Block Diagram



7.2 Registers

The base address of DMA channel 0/1 is described in the following tables:

Table 40. Base Address of DMA channel 0/1

Name	Base Address
DMACH0	0x4000_5D00
DMACH1	0x4000_5D20

Table 41. DMA channel n Register Map (n = 0 and 1)

Name	Offset	Type	Description	Reset Value	Reference
DMACHn_CR	0x0000	RW	DMA Channel n Control Register	0x0000_0000	7.2.1
DMACHn_IESR	0x0004	RW	DMA Channel n Interrupt Enable and Status Register	0x0000_0000	7.2.2
DMACHn_PAR	0x0008	RW	DMA Channel n Peripheral Address Register	0x4000_0000	7.2.3
DMACHn_MAR	0x000C	RW	DMA Channel n Memory Address Register	0x2000_0000	7.2.4

7.2.1 DMACHn_CR: DMA Channel n Controller Register

The DMACHn_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

DMACH0_CR=0x4000_5D00, DMACH1_CR=0x4000_5D20

31 30 29 28	27 26 25 24 23 22 21 20 19 18 17 16	15	14 13	12 11 10 9 8	7 6 5 4	3 2	1	0
Reserved	TRANSCNT[11:0]	ERFGSTP	Reserved	PERISEL[4:0]	Reserved	SIZE[1:0]	DIR	CHnEN
-	0x000	0	-	00000	-	00	0	0
-	RW	RW	-	RW	-	RW	RW	RW

27 16	TRANSCNT[11:0]	The number of times to transfer. These bits should be written out except '0' before the DMA channel is activated.	
		0x000	All transmissions have been completed.
		Others	The number of times left to transfer. The value is decremented by '1' every transfer and the transfer ends when it reaches zero.
15	ERFGSTP	Error Flag Stop bit. This bit is used to stop the DMA transfer when an error flag of a selected peripheral is set to '1'.	
		0	Disable DMA stop function by an error of peripheral
		1	Enable DMA stop function by an error of peripheral
12 8	PERSEL[4:0]	Peripheral Selection.	
		00000	Channel idle
		00001	SPI0 Rx
		00010	SPI0 Tx
		00011	SPI1 Rx
		00100	SPI1 Tx
		00101	USART10 Rx
		00110	USART10 Tx
		00111	I2C0 Rx
		01000	I2C0 Tx
		01001	I2C1 Rx
		01010	I2C1 Tx
		01011	UART0 Rx
		01100	UART0 Tx
		01101	UART1 Rx
		01110	UART1 Tx
		01111	USART11 Rx
		10000	USART11 Tx
		10001	I2C2 Rx
		10010	I2C2 Tx
		10011	USART12 Rx
		10100	USART12 Tx
		Others	Reserved
3 2	SIZE[1:0]	Transfer Size Selection.	
		00	8-bits.
		01	16-bits.
		10	32-bits.

		11	Not used.
1	DIR	Transfer Direction.	
		0	Transfer is from memory to peripheral.
		1	Transfer is from peripheral to memory.
0	CHnEN	DMA Channel Enable. This bit is automatically cleared to ‘0’ immediately after transfer completion or error.	
		0	Disable channel n.
		1	Enable channel n.
		NOTE: 1. All DMA channels must be disabled by software before entering SLEEP and DEEP SLEEP mode.	

NOTES:

1. When SPIn is DMA transfer where n = 0 or 1.
   - The SPInIFLAG bit of SPIn_SR register is the request signal of DMA transfer.
2. When USARTn is DMA transfer where n = 10, 11, or 12.
   - The DREn and RXCn bits of USARTn_ST register are the request signal for Tx and Rx of DMA transfer.
   - If the DORn, FEn, and PEn bits of USARTn_ST register are set during transfer on the ERFGSTP = 1, the corresponding transfer error interrupt flag bit is set, and the transfer will be stopped.
3. When I2Cn is DMA transfer where n = 0, 1, or 2.
   - The values of I2Cn_ST register are the request signal for Tx and Rx of DMA transfer.
   - Abbreviations
     > “SnDA”: Start and Device address.
     > “rSnDA”: Restart and Device address.
     > “CSnSP”: Clear status and stop.
     > “CS”: Clear status.
     > N is the number of bytes to be received or transmitted.
   - On the master Tx: “SnDA” by software + Transmit(N) by DMA + “CSnSP” by software
   - On the master Rx: Up to “rSnDA” by software + Receive (N-1) by DMA + Receive(1+NACK) and “CSnSP” by software
   - On the slave Tx: Up to “rSnDA” by software + Transmit (N) by DMA + “CS” by software
   - On the slave Rx: “SnDA” by software + Receive(N) by DMA + “CS” by software
   - The corresponding DMA channel should be enabled immediately before “SnDA” on a master Tx or slave Rx.
   - The corresponding DMA channel should be enabled immediately before “rSnDA” on a master Rx or slave Tx.
4. When UARTn is DMA transfer where n = 0 or 1.
   - The flags of UARTn_LSR register are the request signal for Tx and Rx of DMA transfer.
   - If an error occurs during transfer on the ERFGSTP = 1, the corresponding transfer error interrupt flag bit is set, and the transfer will be stopped.

7.2.2 DMACHn_IESR: DMA Channel n Interrupt Enable and Status Register

The DMACHn_IESR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1).

DMACH0_IESR=0x4000_5D04, DMACH1_IESR=0x4000_5D24

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6	5	4	3 2	1	0
Reserved	TRERIENn	TRCIENn	Reserved	TRERIFGn	TRCIFGn
-	0	0	-	0	0
.	RW	RW	-	RW	RW

5	TRERIENn	Transfer Error Interrupt Enable.	
		0	Disable transfer error interrupt.
		1	Enable transfer error interrupt.
4	TRCIENn	Transfer Complete Interrupt Enable.	
		0	Disable transfer complete interrupt.
		1	Enable transfer complete interrupt.
1	TRERIFGn	Transfer Error Interrupt Flag bit. This bit is set to ‘1’ when an error occurs on the transfer.	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when write ‘1’.
		NOTE: 1. This bit will be set by an error occur during the ERFGSTP bit of DMACHn_CR register is set to ‘1’.	
0	TRCIFGn	Transfer Complete Interrupt Flag. This bit is set to ‘1’ when the transfer is finished.	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when write ‘1’.
		NOTE: 1. On the DIR bit of DMACHn_CR is ‘0’ (Tx of an interface), this bit is set when there is no data to transmit in DMA memory. That is, since the last data is being transmitted, the next Tx should be started after the corresponding Tx is completed. Check the corresponding flag of Tx interrupt to see if the last Tx is complete.	

7.2.3 DMACHn_PAR: DMA Channel n Peripheral Address Register

The DMACHn_PAR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

DMACH0_PAR=0x4000_5D08, DMACH1_PAR=0x4000_5D28

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PBADR[15:0]	POADR[15:0]
0x4000	0x0000
RO	RW

31 16	PBADR[15:0]	Peripheral Base Address. This is fixed at 0x4000 for APB peripherals.
15 0	POADR[15:0]	Peripheral Offset Address. If the DIR bit is ‘0’, this is the destination offset address of data transfer. If the DIR bit is ‘1’, this is the source offset address of data transfer.

7.2.4 DMACHn_MAR: DMA Channel n Memory Address Register

The DMACHn_MAR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 and 1)

DMACH0_MAR=0x4000_5D0C, DMACH1_MAR=0x4000_5D2C

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14	13 12 11 10 9 8 7 6 5 4 3 2 1 0
MAR[31:16]	MAR[15:14]	MAR[13:0]
0x2000	00	0x0000
RO	RO	RW

31 16	MAR[31:16]	Memory Base Address. These bits are fixed with 0x2000.
15 14	MAR[15:14]	This bit is fixed with “00”.
13 0	MAR[13:0]	Memory Offset Address. This is the address of the memory area from/to which the data will be read/written. This register will be incremented by $2^{SIZE[1:0]}$ for every transfer. When SIZE[1:0] is “01”, the MADR[0] bit is ignored. When SIZE[1:0] is “10”, the MAR[1:0] bits are ignored. If the DIR bit is ‘0’, this is the source memory address of data transfer. If the DIR bit is ‘1’, this is the destination memory address of data transfer.

7.2.5 DMA Register Map Summary

Table 42. DMA Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DMACHn_CR	Res.	Res.	Res.	Res.	TRANSCNT[11:0]												ERFGSTP	Res.	Res.	PERISEL[4:0]					Res.	Res.	Res.	Res.	SIZE[1:0]		DIR	CHnEN
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0					0	0	0	0
0x04	DMACHn_IESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRERIENn	TRCIENn	Res.	Res.	TRERIFGn	TRCIFGn
	Reset value																											0	0			0	0
0x08	DMACHn_PAR	PBADR[15:0]																POADR[15:0]															
	Reset value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DMACHn_MAR	MAR[31:16]																MAR[15:14]		MAR[13:0]													
	Reset value	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8. Watchdog Timer (WDT)

8.1 WDT Introduction

Watchdog Timer (WDT) rapidly detects CPU malfunctions such as endless loops caused by noise and returns the CPU to the normal state. WDT signal for malfunction detection can be used as either a CPU reset or an interrupt request.

When the WDT is not being used for malfunction detection, it can be used as a timer to generate interrupts at fixed intervals. When the WDT_CNT value reaches WDT_WINDR value, a watchdog interrupt can be generated. The underflow time of the WDT can be set by the WDT_DR register. If an underflow occurs, an internal reset may be generated. The WDT operates at 40kHz which is the embedded RC oscillator clock.

8.2 WDT Main Features

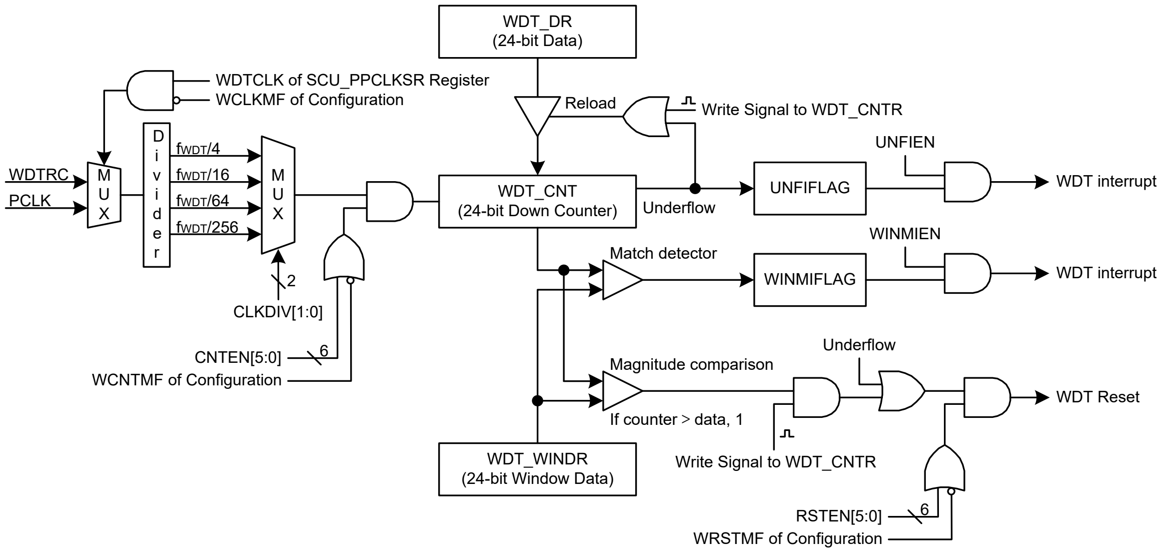
The WDT module has main features as listed below:

- 24-bit down counter (WDT_CNT)
- Reset or periodic interrupt selection
- Count clock selection
- Watchdog overflow output signal
- Counter window function

8.3 WDT Block Diagram

Figure 32 shows a block diagram for the WDT.

Figure 32. WDT Block Diagram



NOTES:

1. When a reload signal is generated, the WDT_DR value is reloaded to WDT_CNT register, after 1-clock delay, with the clock selected by the CLKDIV bits.
2. The WDT_WINDR register is a register that determines the range in which the value of WDT_DR register can be reloaded into the WDT_CNT register without reset occurring in the WDT. In the picture above, it must be reloaded when the value of WDT_WINDR register is greater than the value of WDT_CNT register.

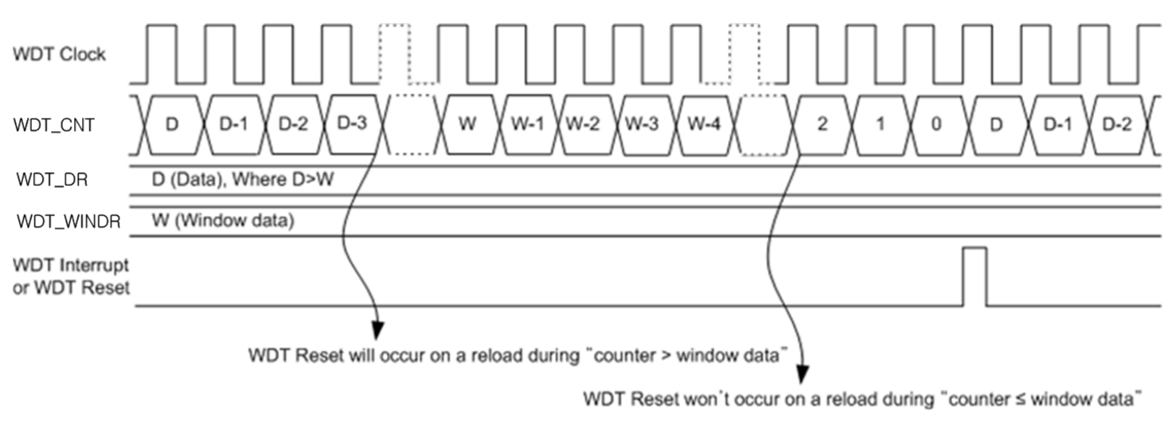
8.4 WDT Functional Description

Watchdog timer count can be enabled by CNTEN (WDT_CR[9:4]) set as any value other than 0x1A. As the WDT activates, the down counter will start counting from the load value. If the RSTEN (WDT_CR[15:10]) is set as any value other than 0x25, WDT reset would be asserted when the WDT counter value reaches zero (underflow event) from WDT_DR value.

Before the WDT counter reaches zero, software can write 0x6A to WDT_CNTR register to reload WDT counter when the counter value is less than or equal to the value of window data register. WDT reset may be asserted if the reload occurs when counter > window data.

8.4.1 Timing Diagram

Figure 33. WDT Interrupt and WDT Reset Timing Diagram


WDT Clock
WDT_CNT
D
D-1
D-2
D-3
W
W-1
W-2
W-3
W-4
2
1
0
D
D-1
D-2
WDT_DR
D (Data), Where D>W
WDT_WINDR
W (Window data)
WDT Interrupt or WDT Reset
WDT Reset will occur on a reload during "counter > window data"
WDT Reset won't occur on a reload during "counter ≤ window data"

8.4.2 Pre-scale Table

The WDT includes a 24-bit down counter with programmable pre-scaler to define different time-out intervals.

Clock sources of the WDT can be WDTRC or PCLK. The PCLK can be selected by setting WDTCLK (SCU_PPCLKSR[0]) to ‘1’ when the CONF_WDTCNFIG[2] bit of configure option page 1 is '0'.

A WDT counter can be set as a base clock by controlling a 2-bit pre-scaler CLKDIV [1:0] in the WDT_CR register. The maximum pre-scaled value is “clock source frequency/256”. The pre-scaled WDT counter clock frequency values are listed in Table 43.

Selectable clock source (40 kHz ~ 32 MHz) and time-out interval at a single count

Time-out period = (Load Value + 1) × (1/pre-scaled WDT counter clock frequency)

*Time out period (when the Load Value reaches zero, underflow flag is set to ‘1’)

Table 43. Prescaled WDT Counter Clock Frequency

Clock Source	WDT CLKIN	WDTCLKIN / 4	WDTCLKIN / 16	WDTCLKIN / 64	WDTCLKIN / 256
WDTRC	40 kHz	10 kHz	2.5 kHz	0.625 kHz	0.156 kHz
PCLK	PCLK	PCLK/4	PCLK/16	PCLK/64	PCLK/256

8.5 WDT Registers

The base address and register map of the WDT are described in the followings:

Table 44. Base Address of WDT

Name	Base Address
WDT	0x4000_1A00

Table 45. WDT Register Map

Name	Offset	Type	Description	Reset Value	Reference
WDT_CR	0x0000	RW	Watchdog Timer Control Register	0x00000000	8.5.1
WDT_SR	0x0004	RW	Watchdog Timer Status Register	0x00000080	8.5.2
WDT_DR	0x0008	RW	Watchdog Timer Data Register	0x00000FFF	8.5.3
WDT_CNT	0x000C	RO	Watchdog Timer Counter Register	0x00000FFF	8.5.4
WDT_WINDR	0x0010	RW	Watchdog Timer Window Data Register	0x00001FFF	8.5.5
WDT_CNTR	0x0014	WO	Watchdog Timer Counter Reload Register	0x00000000	8.5.6

8.5.1 WDT_CR: Watchdog Timer Control Register

The WDT module should be configured properly before running. The WDT module can reset the system or assert an interrupt signal to the system. This register is a 32-bit size.

WDT_CR=0x4000_1A00

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">WTIDKY[15:0]</td><td colspan="6">RSTEN[5:0]</td><td colspan="6">CNTEN[5:0]</td><td>WINMIEN</td><td>UNFIEN</td><td colspan="2">CLKDIV[1:0]</td></tr>
<tr><td colspan="16">0x0000</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="16">WO</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td></tr>
</table>

31 16	WTIDKY[15:0]	Write Identification Key.	
		When writing, write 0x5A69 to these bits, or else writing is ignored.	
15 10	RSTEN[5:0]	Watchdog Timer Reset Enable.	
		0x25	Disable watchdog timer reset.
		Others	Enable watchdog timer reset.
9 4	CNTEN[5:0]	Watchdog Timer Counter Enable.	
		0x1A	Disable watchdog timer counter.
		Others	Enable watchdog timer counter.
3	WINMIEN	Watchdog Timer Window Match Interrupt Enable.	
		0	Disable window data match interrupt.
		1	Enable window data match interrupt.
2	UNFIEN	Watchdog Timer Underflow Interrupt Enable.	
		0	Disable watchdog timer underflow interrupt.
		1	Enable watchdog timer underflow interrupt.
1 0	CLKDIV[1:0]	Watchdog Timer Clock Divider. The watchdog timer clock is selected by SCU_PPCLKSR[0] bit of clock generation and CONF_WDTCNFIG[2] bit of Configure Option Page 1.	
		00	fWDT/4
		01	fWDT/16
		10	fWDT/64
		11	fWDT/256

8.5.2 WDT_SR: Watchdog Timer Status Register

The WDT_SR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

WDT_SR=0x4000_1A04

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved	DBGCNTEN	Reserved					WINMIFLAG	UNFIFLAG
-	1	-	-	-	-	-	0	0
-	RW	-	-	-	-	-	RW	RW

7	DBGCNTEN	Watchdog Timer Counter Enable bit when the core is halted in debug mode.	
		0	The watchdog timer counter continues operation even if the core is halted.
		1	The watchdog timer counter stops when the core is halted.
		NOTE: 1. This bit is set to '1' by POR reset.	
1	WINMIFLAG	Watchdog Timer Window Match Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to '0' when '1' is written.
0	UNFIFLAG	Watchdog Timer Underflow Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to '0' when '1' is written.

8.5.3 WDT_DR: Watchdog Timer Data Register

The WDT_DR register is used to update WDT_CNT register. This register is a 32-bit size.

WDT_DR=0x4000_1A08

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	DATA[23:0]		
-	0x000FFF		
.	RW		

23 0	DATA[23:0]	Watchdog Timer Data. The range is 0x000000 to 0xFFFFFF

NOTE:

1. Once any value is written to this data register, the register cannot be changed until system reset.

8.5.4 WDT_CNT: Watchdog Timer Counter Register

The WDT_CNT register represents the current count value of the 32-bit down counter. When the counter value reaches zero, an interrupt or a reset will be asserted. This register is a 32-bit size.

WDT_CNT=0x4000_1A0C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	CNT[23:0]		
-	0x000FFF		
.	RO		

23 0	CNT[23:0]	Watchdog Timer Counter

8.5.5 WDT_WINDR: Watchdog Timer Window Data Register

The WDT_WINDR register is used to compare to WDT_CNT for WINDOW function. This register is a 32-bit size.

WDT_WINDR=0x4000_1A10

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	WDATA[23:0]		
-	0x001FFF		
.	RO		

23 0	WDATA[23:0]	Watchdog Timer Window Data. The range is 0x000000 to 0xFFFFFF.

NOTE:

1. Once any value is written to this window data register, the register cannot be changed until system reset.

8.5.6 WDT_CNTR: Watchdog Timer Counter Reload Register

The WDT_CNTR register is used to generate a reload signal. When a reload signal is generated, the WDT_DR value is reloaded to WDT_CNT. This register is a 32-bit size.

WDT_CNTR=0x4000_1A14

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			CNTR[7:0]
-			0x00
.			WO

7 0	CNTR[7:0]	Watchdog Timer Counter Reload bits.	
		0x6A	Reload the WDT_DR value to watchdog timer counter and re-start. (Automatically cleared to “0x00” after operation)
		Others	No effect

8.5.7 WDT Register Map Summary

Table 46. WDT Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WDT_CR	WTIDKY[15:0]																RSTEN[5:0]						CNTEN[5:0]						WINMIEN	UNFIEN	CLKDIV[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	WDT_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBGCNTEN	Res	Res	Res	Res	Res	WINMIFLAG	UNFIFLAG
	Reset value																									1						0	0
0x08	WDT_DR	Res	Res	Res	Res	Res	Res	Res	Res	DATA[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	WDT_CNT	Res	Res	Res	Res	Res	Res	Res	Res	CNT[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0x10	WDT_WINDR	Res	Res	Res	Res	Res	Res	Res	Res	WDATA[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	WDT_CNTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNTR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

9. Watch Timer (WT)

9.1 WT Introduction

Watch Timer (WT) has a function for RTC (Real-Time Clock) operation. It is generally used for RTC design. The internal structure of the watch timer consists of the clock source select circuit, timer counter circuit, output select circuit and watch timer control register. To operate the watch timer, determine the input clock source, output interval, and set the WTEN bit as ‘1’ in watch timer control register (WT_CR). It can operate simultaneously or individually. To stop the WT, clear the WTEN bit in the WT_CR register to ‘0’. Even when the CPU is in STOP mode, the sub clock is working, and the WT can continue operation. Through the WT registers, it is possible to clear the WT counter, set the interval value, and read the 12-bit WT counter value.

9.2 WT Main Features

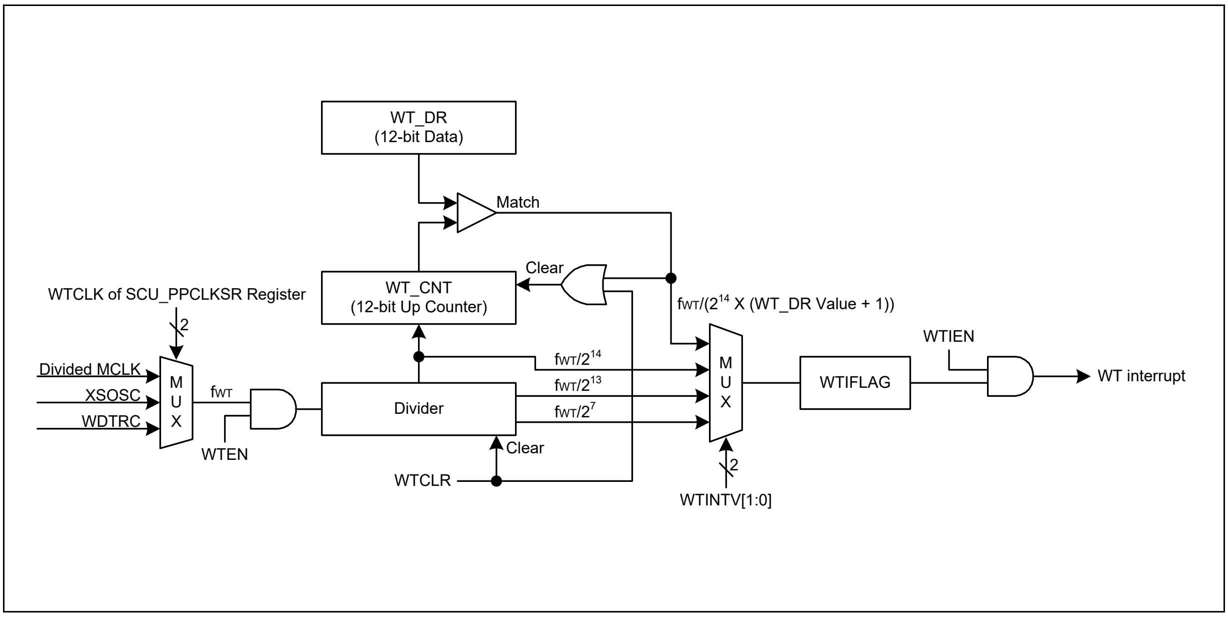
The WT module has main features as listed below:

- 14-bit divider
- 12-bit up-counter
- RTC function

9.3 WT Block Diagram

Figure 34 shows a block diagram for the WT.

Figure 34. WT Block Diagram



9.4 WT Registers

The base address and register map of the WT are described in the followings:

Table 47. Base Address of WT

Name	Base Address
WT	0x4000_2000

Table 48. WT Register Map

Name	Offset	Type	Description	Reset Value	Reference
WT_CR	0x0000	RW	Watch Timer Control Register	0x00000000	9.4.1
WT_DR	0x0004	RW	Watch Timer Data Register	0x00000FFF	9.4.2
WT_CNT	0x0008	RO	Watch Timer Counter Register	0x00000000	9.4.3

9.4.1 WT_CR: Watch Timer Control Register

The WT_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

WT_CR=0x4000_2000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			WTEN	Reserved	WTINTV[1:0]		WTIEN	Reserved	WINMIFLAG	UNFIFLAG
-			0	-	0	0	0	-	0	0
-			RW	-	RW	RW	RW	-	RW	RW

7	WTEN	Watch Timer Operation Enable.	
		0	Disable watch timer operation.
		1	Enable watch timer operation.
5 4	WTINTV[1:0]	Watch Timer Interval Selection.	
		00	fWT/2^7
		01	fWT/2^{13}
		10	fWT/2^{14}
		11	fWT/(2^{14}x(WT_DR value + 1))
		NOTE: 1. These bits should be changed while the WTEN bit is '0'.	
3	WTIEN	Watch Timer Interrupt Enable.	
		0	Disable watch timer interrupt.
		1	Enable watch timer interrupt.
1	WTIFLAG	Watch Timer Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to '0' when '1' is written.
0	WTCLR	Watch Timer Counter and Divider Clear.	
		0	No effect.
		1	Clear the counter and divider (automatically cleared to '0' after operation)

9.4.2 WT_DR: Watch Timer Data Register

The WT_DR register is a 32-bit size. This register is available at 32-/ 16-bit access.

WT_DR=0x4000_2004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				WTDATA[11:0]											
-																				0xFFF											
-																				RW											

11 0	WTDATA[11:0]	Watch Timer Data. The range is 0x001 to 0xFFF.

9.4.3 WT_CNT: Watch Timer Counter Register

The WT_CNT register is a 32-bit size. This register is available at 32-/ 16-bit access.

WT_CNT=0x4000_2008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				CNT[11:0]											
-																				0x000											
-																				RO											

11 0	CNT[11:0]	Watch Timer Counter.

9.4.4 WT Register Map Summary

Table 49. WT Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WT_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WTEN	Res.	WTINTV[1:0]		WTIEN	Res.	WTIFLAG	WTCLR
	Reset value																									0		0	0	0		0	0
0x04	WT_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WTDATA[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x08	WT_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0

10. Timer Counter 10/11/12/13/14/15

10.1 Timer Counter 10/11/12/13/14/15 Introduction

The timer block comprises six channels of 16-bit general purpose timers. Each has an independent 16-bit counter and a dedicated prescaler that feeds counting clock. They support periodic timer, PWM pulse, one-shot and capture mode.

One more optional free-run timer is provided. The main purpose of this timer is a periodical tick timer or a wake-up source.

10.2 Timer Counter 10/11/12/13/14/15 Main Features

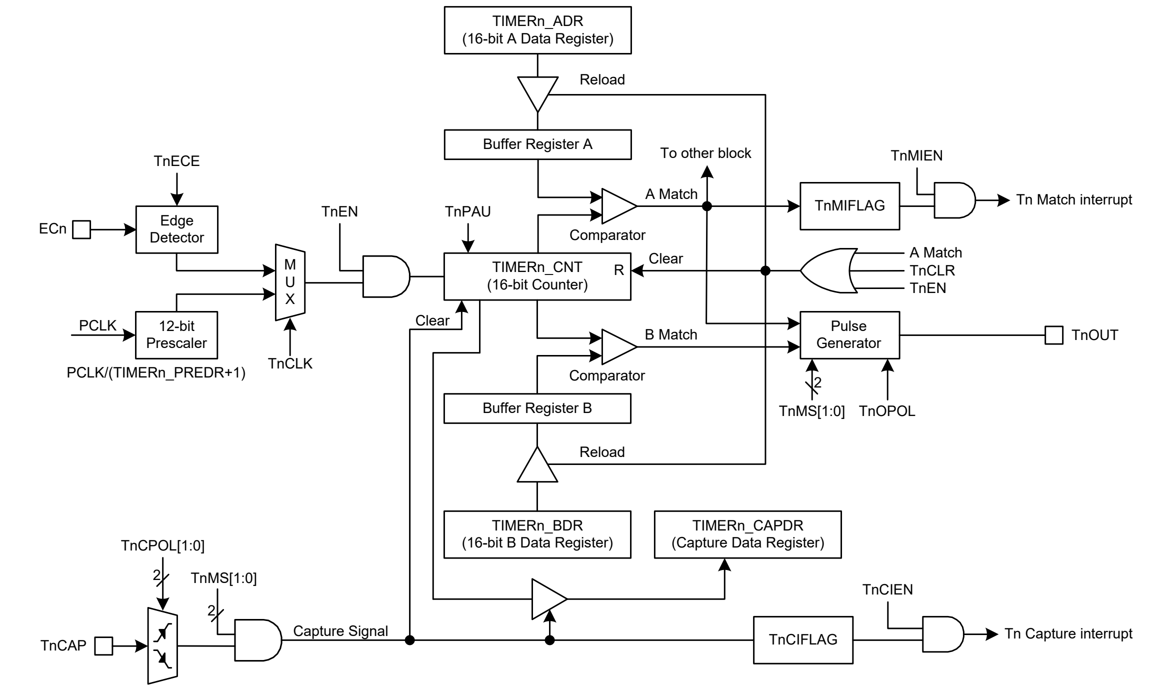
Timer counter 10/11/12/13/14/15 includes the features below:

- 16-bit up-counter and 12-bit prescaler
- Four operating modes:
  - Periodic Mode
  - One-shot Mode
  - PWM Mode
  - Capture Mode
- Various interrupts:
  - Match interrupt
  - Capture interrupt
- Synchronous start and clear function

10.3 Timer Counter 10/11/12/13/14/15 Functional Description

10.3.1 Block Diagram

Figure 35. Timer Counter n Timer Block Diagram (n = 10, 11, 12, 13, 14, and 15)



NOTES:

1. TnEN bit is automatically cleared to logic ‘0’ after one pulse is generated at a PPG one-shot mode.
2. After the TnEN bit is set, counting of the TIMERn_CNT may be delayed two clocks or more.

10.4 Pin Description for Timer Counter 10/11/12/13/14/15

Table 50. Pins and External Signals for Timer Counter n (n = 10, 11, 12, 13, 14, and 15)

Pin Name	Type	Description
ECn	I	External clock input
TnCAP	I	Capture input
TnOUT	O	PWM/one-shot output

10.5 Functional Description

10.5.1 Timer Counter 10/11/12/13/14/15

Timer/counter n can use an internal or an external clock source (ECn). A clock selection logic can select a clock source and it is controlled by clock selection bits (TnCLK).

- Timer n clock source: {PCLK / (TIMERn_PREDR+1)}, ECn

In Capture mode, by TnCAP, data is captured into a corresponding capture data register (TIMERn_CAPDR). Timer n outputs the comparison result between counter and data register through TnOUT port in Timer/counter mode. Also, timer n output PWM waveform through TnOUT port under PPG mode. (n = 10, 11, 12, 13, 14 and 15)

Table 51. Timer n Operating Modes (n = 10, 11, 12, 13, 14, and 15)

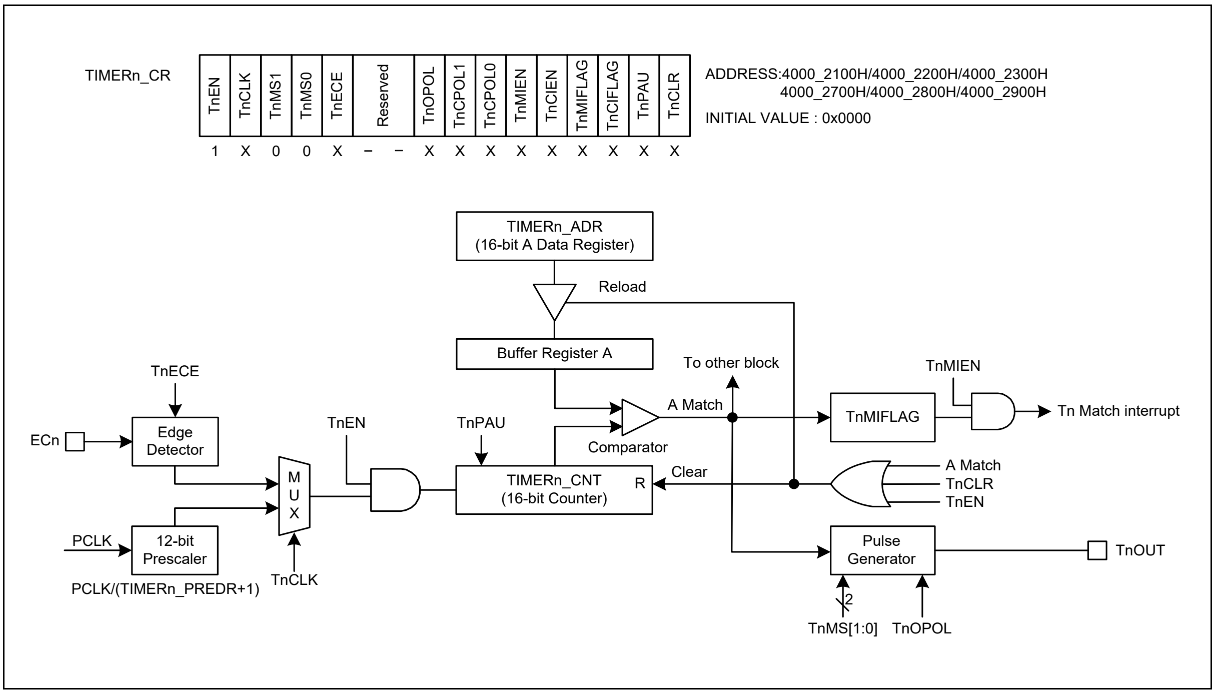
TnEN	Alternative Mode	TnMS	TIMERn_PREDR	Timer n
1	TIMER10: T10OUT TIMER11: T11OUT TIMER12: T12OUT TIMER13: T13OUT TIMER14: T14OUT TIMER15: T15OUT	00	0xXXX	Timer/Counter Mode
1	TIMER10: T10CAP TIMER11: T11CAP TIMER12: T12CAP TIMER13: T13CAP TIMER14: T14CAP TIMER15: T15CAP	01	0xXXX	Capture Mode
1	TIMER10: T10OUT TIMER11: T11OUT TIMER12: T12OUT TIMER13: T13OUT TIMER14: T14OUT TIMER15: T15OUT	10	0xXXX	PPG Mode (one-shot mode)
1		11	0xXXX	PPG Mode (repeat mode)

10.5.2 16-bit Timer/Counter Mode

16-bit Timer/counter mode is selected by control register as shown in Figure 36. The 16-bit timer has a counter register and a data register. The counter register is increased by internal or external clock input. Timer n can use an input clock with 12-bit prescaler division rates (TIMERn_PREDR) and an external clock (ECn). When the values of TIMERn_CNT and TIMERn_ADR are the same in the timer n, a match signal is generated, and the interrupt of Timer n takes place.

The TIMERn_CNT values are automatically cleared by the match signal. It can also be cleared by software (TnCLR).

Figure 36. 16-bit Timer/Counter Mode for Timer n (n = 10, 11, 12, 13, 14, and 15)



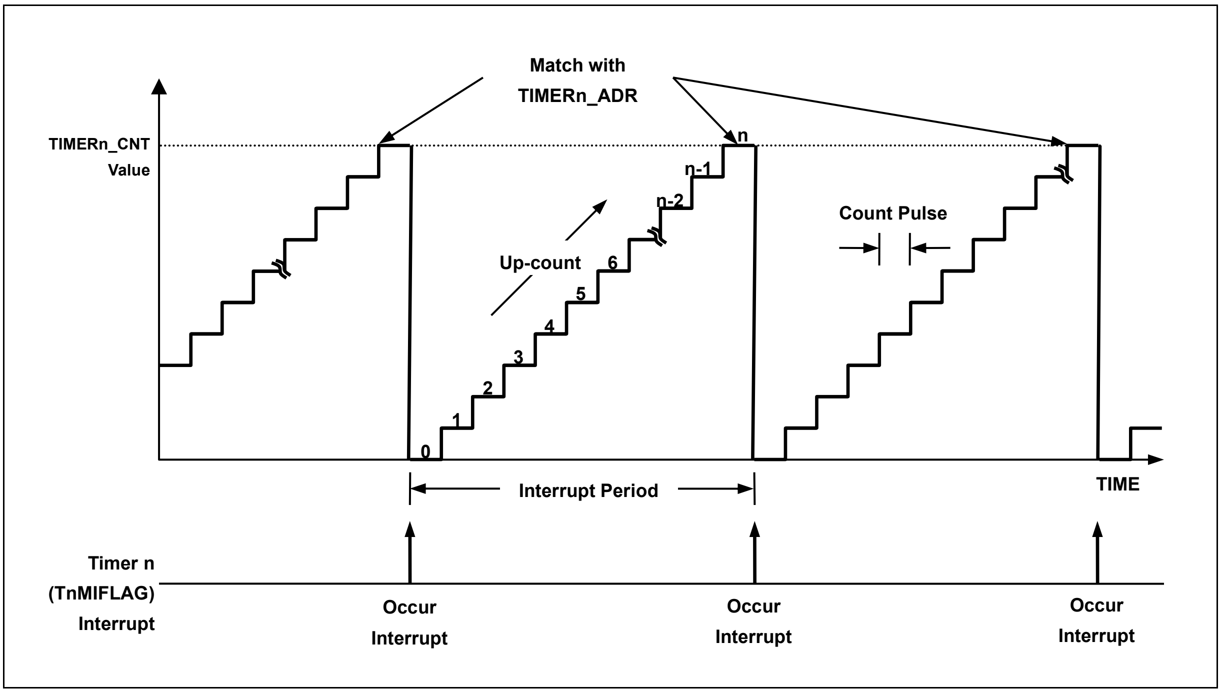
TIMERn_CR

TnEN	TnCLK	TnMS1	TnMS0	TnECE	Reserved		TnOPOL	TnCPOL1	TnCPOL0	TnMIEN	TnCIEN	TnMIFLAG	TnCIFLAG	TnPAU	TnCLR
1	X	0	0	X	–	–	X	X	X	X	X	X	X	X	X

ADDRESS:4000_2100H/4000_2200H/4000_2300H
4000_2700H/4000_2800H/4000_2900H

INITIAL VALUE : 0x0000

Figure 37. 16-bit Timer/Counter n Example (n = 10, 11, 12, 13, 14, and 15)



10.5.3 16-bit Capture Mode

Timer n Capture mode is evoked by configuring the TnMS[1:0] bits as “01”. The internal clock can be used as a clock source. It basically has the same function as the 16-bit timer/counter mode and an interrupt takes place when the TIMERn_CNT value becomes equal to the value of TIMERn_ADR register.

This timer interrupt in Capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. The capture result is loaded into the TIMERn_CAPDR register. In the timer n capture mode, timer n output (TnOUT) waveform is not available.

Figure 38. 16-bit Capture Mode for Timer n (n = 10, 11, 12, 13, 14, and 15)

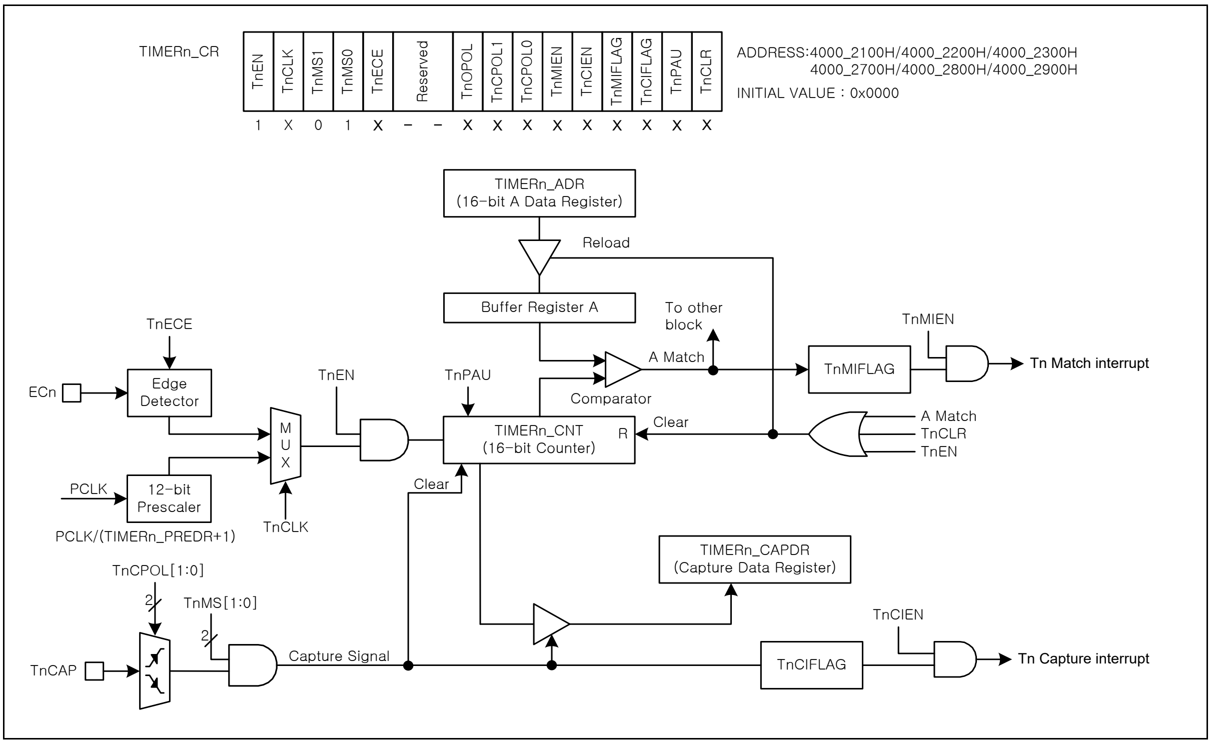


Figure 39. 16-bit Capture Mode for Timer n (n = 10, 11, 12, 13, 14, and 15)

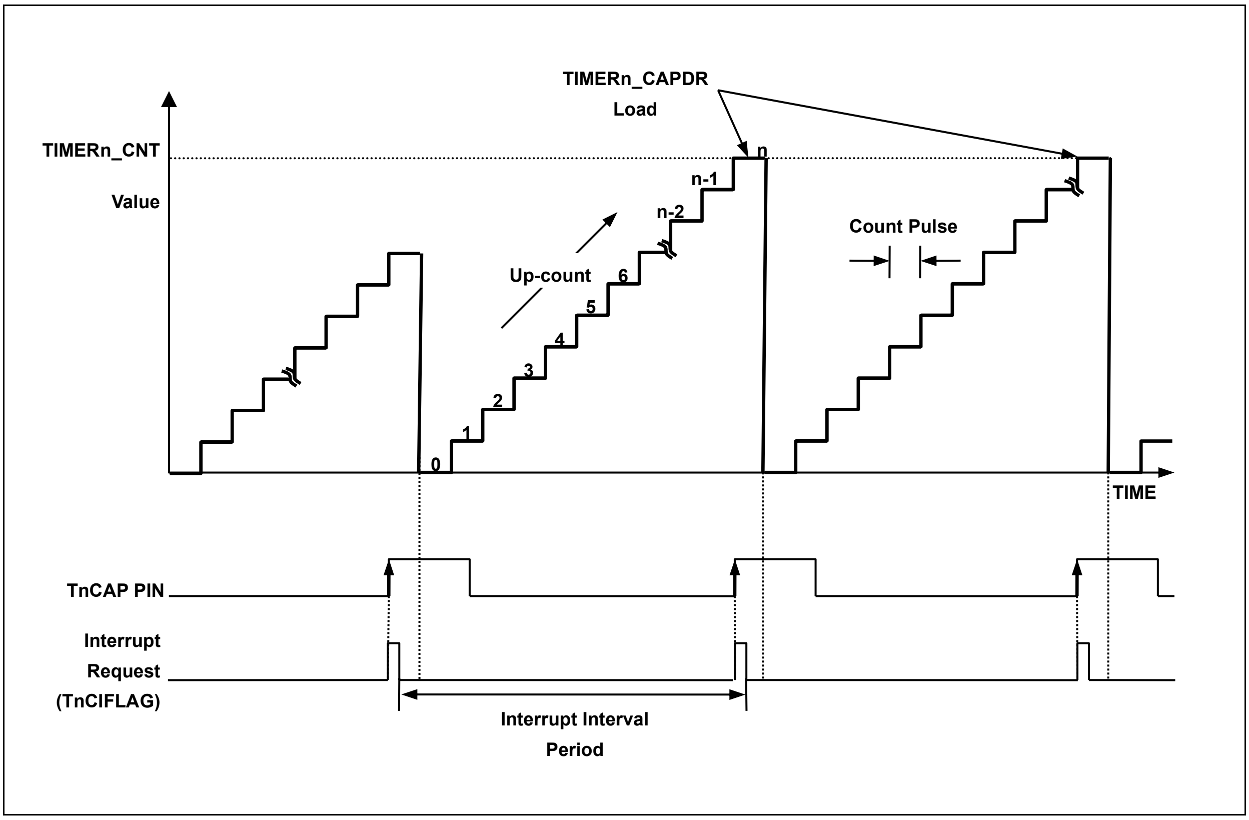
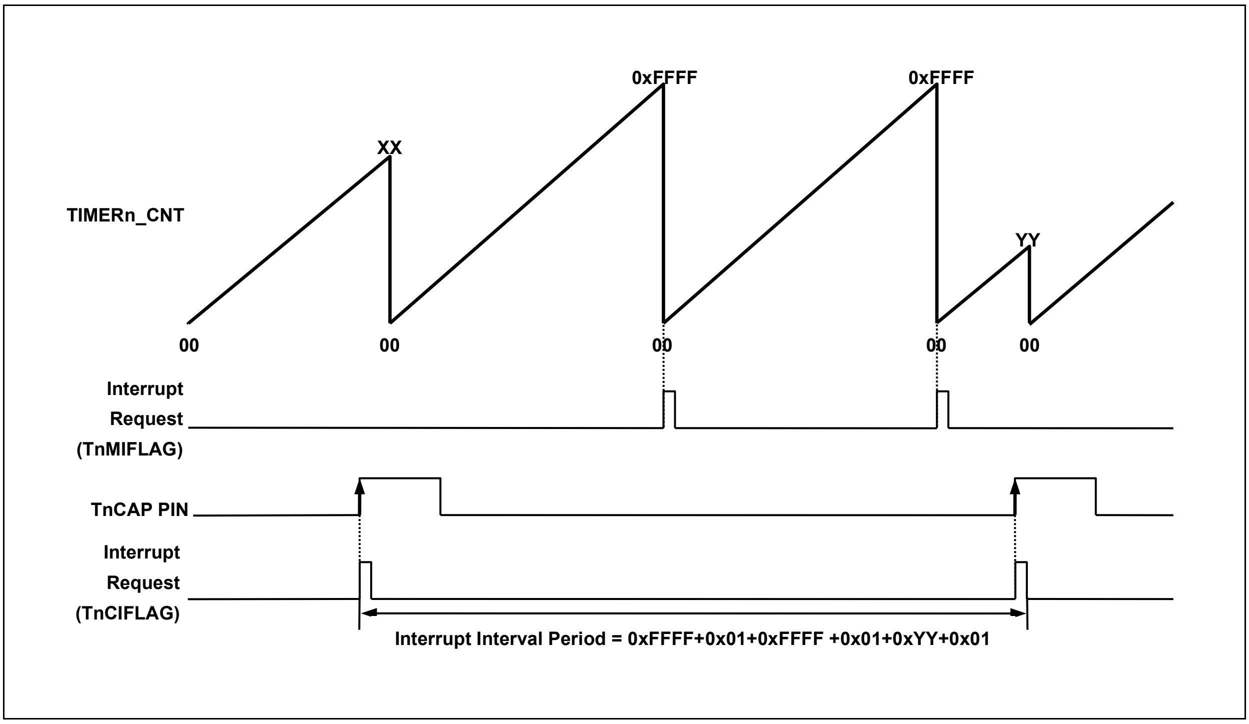


Figure 40. Express Timer Overflow in Capture Mode (n = 10, 11, 12, 13, 14, and 15)



10.5.4 16-bit PPG Mode

Timer n has a PPG (Programmable Pulse Generation) function. In PPG mode, the TnOUT pin generates PWM output of up to 16-bit resolution. This pin should be configured as a PWM output by setting Px_AFSR1, Px_AFSR2 to 'AF0' or 'AF3'. The period of PWM output is determined by the TIMERn_ADR. The duty of PWM output is determined by the TIMERn_BDR register. (x = A to F)

Figure 41. 16-bit PPG Repeat and One-shot Mode for Timer n (n = 10, 11, 12, 13, 14, and 15)

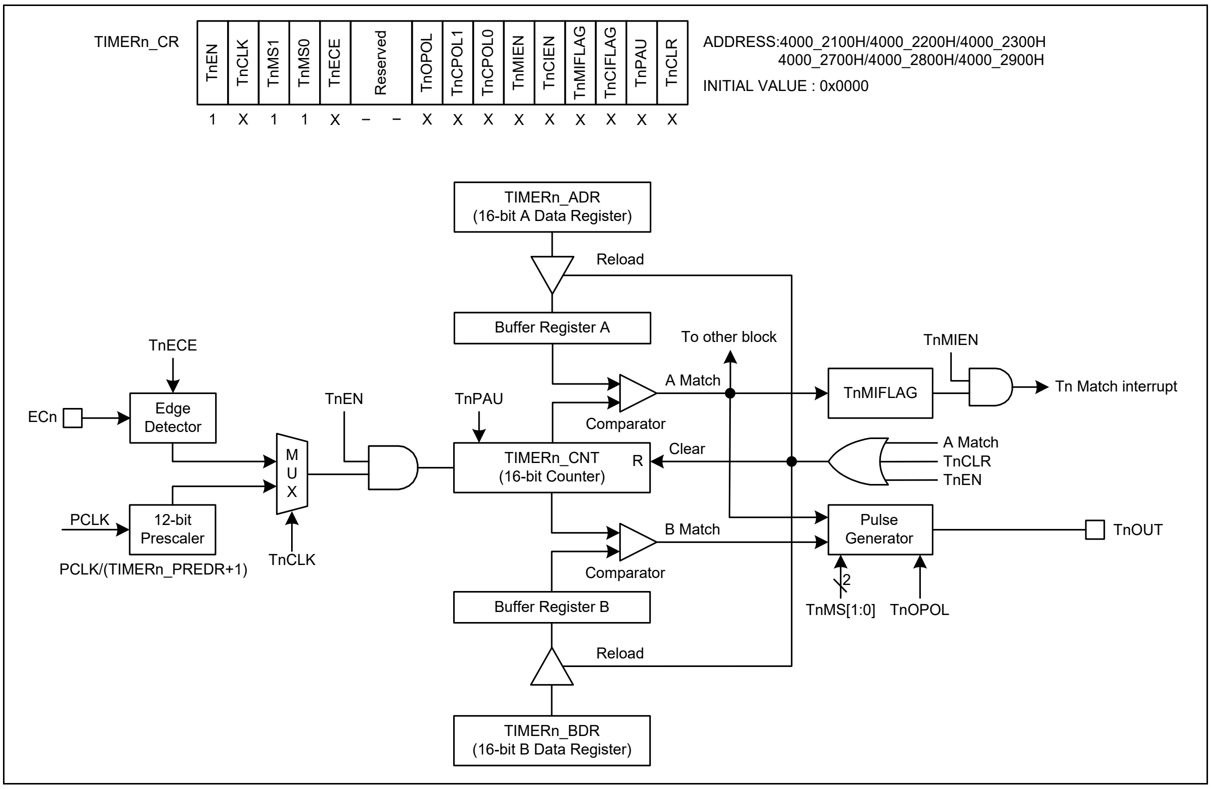
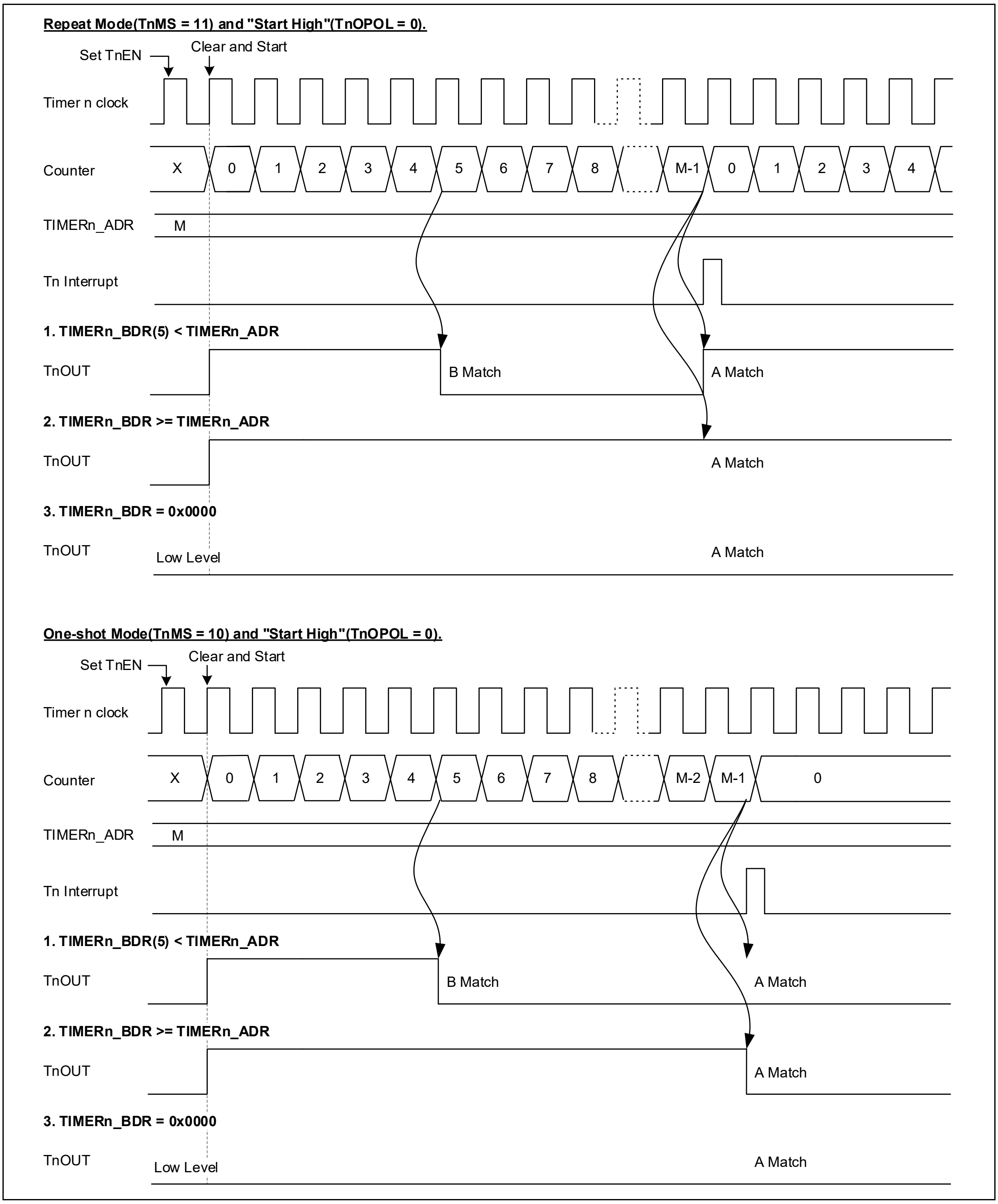


Figure 42. 16-bit PPG Mode Timing chart for Timer n (n = 10, 11, 12, 13, 14, and 15)



10.6 TIMER10/11/12/13/14/15 Registers

The base address of Timer 10/11/12/13/14/15 is described in the followings:

Table 52. Base Address of Timer 10/11/12/13/14/15

Name	Base Address
TIMER10	0x4000_2100
TIMER11	0x4000_2200
TIMER12	0x4000_2300
TIMER13	0x4000_2700
TIMER14	0x4000_2800
TIMER15	0x4000_2900

Table 53. TIMER Register Map

Name	Offset	Type	Description	Reset Value	Reference
TIMERn_CR	0x00	RW	Timer/Counter n Control Register	0x0000_0000	10.6.1
TIMERn_ADR	0x04	RW	Timer/Counter n A Data Register	0x0000_FFFF	10.6.2
TIMERn_BDR	0x08	RW	Timer/Counter n B Data Register	0x0000_FFFF	10.6.3
TIMERn_CAPDR	0x0C	RO	Timer/Counter n Capture Data Register	0x0000_0000	10.6.4
TIMERn_PREDR	0x10	RW	Timer/Counter n Prescaler Data Register	0x0000_0FFF	10.6.5
TIMERn_CNT	0x14	RO	Timer/Counter n Counter Register	0x0000_0000	10.6.6

NOTE:

1. n = 10, 11, 12, 13, 14, and 15.

10.6.1 TIMERn_CR: Timer/Counter n Control Register

The timer module should be configured properly before running. The timer should be configured with the appropriate value in TIMERn_CR register for designated operating mode. After configuring this register, a user can start or stop the timer function by using this register.

TIMERn_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, 12, 13, 14, and 15).

TIMER10_CR=0x4000_2100, TIMER11_CR=0x4000_2200, TIMER12_CR=0x4000_2300
TIMER13_CR=0x4000_2700, TIMER14_CR=0x4000_2800, TIMER15_CR=0x4000_2900

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15	14	13 12	11	10 9	8	7 6	5	4	3	2	1	0
Reserved	TnEN	TnCLK	TnMS[1:0]	TnECE	Reserved	TnOPOL	TnCPOL[1:0]	TnMIEN	TnCIEN	TnMIFLAG	TnCIFLAG	TnPAU	TnCLR
-	0	0	00	0	-	0	00	0	0	0	0	0	0
-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW

15	TnEN	Timer n Operation Enable.	
		0	Disable timer n operation.
		1	Enable timer n operation. (Counter clear and start)
14	TnCLK	Timer n Clock Selection.	
		0	Select an internal prescaler clock.
		1	Select an external clock.
		NOTE: 1. This bit should be changed while TnEN bit is ‘0’.	
13 12	TnMS[1:0]	Timer n Operation Mode Selection.	
		00	Timer/Counter mode. (TnOUT: toggle at A-match)
		01	Capture mode. (The A-match interrupt can occur)
		10	PPG one-shot mode. (TnOUT: Programmable pulse output)
		11	PPG repeat mode. (TnOUT: Programmable pulse output)
		NOTE: 1. This bit should be changed while TnEN bit is ‘0’.	
11	TnECE	Timer n External Clock Edge Selection.	
		0	Select falling edge of external clock.
		1	Select rising edge of external clock.
8	TnOPOL	TnOUT Polarity Selection.	
		0	Start high. (TnOUT is low level at disable)
		1	Start low. (TnOUT is high level at disable)
7 6	TnCPOL[1:0]	Timer n Capture Polarity Selection.	
		00	Capture on falling edge.
		01	Capture on rising edge.
		10	Capture on both falling and rising edge.
		11	Reserved.
5	TnMIEN	Timer n Match Interrupt Enable.	
		0	Disable timer n match interrupt.
		1	Enable timer n match interrupt.
4	TnCIEN	Timer n Capture Interrupt Enable.	
		0	Disable timer n capture interrupt.

		1	Enable timer n capture interrupt.
3	TnMIFLAG	Timer n Match Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
2	TnCIFLAG	Timer n Capture Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
1	TnPAU	Timer n Counter Temporary Pause Control.	
		0	Continue counting.
		1	Temporary pause.
0	TnCLR	Timer n Counter and Prescaler Clear.	
		0	No effect.
		1	Clear timer n counter and prescaler. (Automatically cleared to ‘0’ after operation)

10.6.2 TIMERn_ADR: Timer/Counter n A Data Register

The TIMERn_ADR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, 13, 14, and 15).

TIMER10_ADR=0x4000_2104, TIMER11_ADR=0x4000_2204, TIMER12_ADR=0x4000_2304
TIMER13_ADR=0x4000_2704, TIMER14_ADR=0x4000_2804, TIMER15_ADR=0x4000_2904

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		ADATA[15:0]	
-		0xFFFF	
-		RW	

15 0	ADATA[15:0]	Timer/Counter n A Data. The range is 0x0002 to 0xFFFF.

NOTE:

1. Do not write “0x0000” in the TIMERn_ADR register under PPG mode.

10.6.3 TIMERn_BDR: Timer/Counter n B Data Register

The TIMERn_BDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, 12, 13, 14, and 15).

TIMER10_BDR=0x4000_2108, TIMER11_BDR=0x4000_2208, TIMER12_BDR=0x4000_2308
TIMER13_BDR=0x4000_2708, TIMER14_BDR=0x4000_2808, TIMER15_BDR=0x4000_2908

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		BDATA[15:0]	
-		0xFFFF	
-		RW	

15 0	BDATA[15:0]	Timer/Counter n B Data. The range is 0x0000 to 0xFFFF.

10.6.4 TIMERn_CAPDR: Timer/Counter n Capture Data Register

The TIMERn_CAPDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, 12, 13, 14, and 15).

TIMER10_CAPDR=0x4000_210C, TIMER11_CAPDR=0x4000_220C, TIMER12_CAPDR=0x4000_230C
TIMER13_CAPDR=0x4000_270C, TIMER14_CAPDR=0x4000_280C, TIMER15_CAPDR=0x4000_290C

<table>
<tr><td colspan="8">31 30 29 28 27 26 25 24</td><td colspan="8">23 22 21 20 19 18 17 16</td><td colspan="8">15 14 13 12 11 10 9 8</td><td colspan="8">7 6 5 4 3 2 1 0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">CAPD[15:0]</td></tr>
<tr><td colspan="16">-</td><td colspan="16">0x0000</td></tr>
<tr><td colspan="16">-</td><td colspan="16">RO</td></tr>
</table>

15 0	CAPD[15:0]	Timer/Counter n Capture Data.

10.6.5 TIMERn_PREDR: Timer/Counter n Prescaler Data Register

The TIMERn_PREDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, 12, 13, 14, and 15).

TIMER10_PREDR=0x4000_2110, TIMER11_PREDR=0x4000_2210, TIMER12_PREDR=0x4000_2310
TIMER13_PREDR=0x4000_2710, TIMER14_PREDR=0x4000_2810, TIMER15_PREDR=0x4000_2910

<table>
<tr><td colspan="8">31 30 29 28 27 26 25 24</td><td colspan="8">23 22 21 20 19 18 17 16</td><td colspan="8">15 14 13 12 11 10 9 8</td><td colspan="8">7 6 5 4 3 2 1 0</td></tr>
<tr><td colspan="20">Reserved</td><td colspan="12">PRED[11:0]</td></tr>
<tr><td colspan="20">-</td><td colspan="12">0x0FFF</td></tr>
<tr><td colspan="20">-</td><td colspan="12">RW</td></tr>
</table>

11 0	PRED[11:0]	Timer/Counter n Prescaler Data.

10.6.6 TIMERn_CNT: Timer/Counter n Counter Register

The TIMERn_CNT register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, 12, 13, 14, and 15).

TIMER10_CNT=0x4000_2114, TIMER11_CNT=0x4000_2214, TIMER12_CNT=0x4000_2314
TIMER13_CNT=0x4000_2714, TIMER14_CNT=0x4000_2814, TIMER15_CNT=0x4000_2914

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">CNT[15:0]</td></tr>
<tr><td colspan="16">-</td><td colspan="16">0x0000</td></tr>
<tr><td colspan="16">-</td><td colspan="16">RO</td></tr>
</table>

15 0	CNT[15:0]	Timer/Counter n Counter.

10.6.7 TIMERn Register Map Summary

Table 54. TIMERn Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMERn_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TnEN	TnCLK	TnMS[1:0]		TnECE	Res.	Res.	TnOPOL	TnCPOL		TnMIEN	TnCIEN	TnMIFLAG	TnCIFLAG	TnPAU	TnCLR
	Reset value																	0	0	0	0	0			0	0	0	0	0	0	0	0	0
0x04	TIMERn_ADR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADATA[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	TIMERn_BDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDATA[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	TIMERn_CAPDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAPD[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	TIMERn_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	TIMERn_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOTE:

1. n = 10, 11, 12, 13, 14, and 15

11. Timer Counter 20/21

11.1 Timer Counter 20/21 Introduction

The timer block comprises two channels 32-bit general-purpose timers. Each timer has an independent 32-bit counter and a dedicated prescaler that feeds counting clock. It supports periodic timer, PWM pulse, one-shot timer, and capture mode.

One more optional free-run timer is provided. Main purpose of this timer is a periodical tick timer or a wake-up source.

11.2 Timer Counter 20/21 Main Features

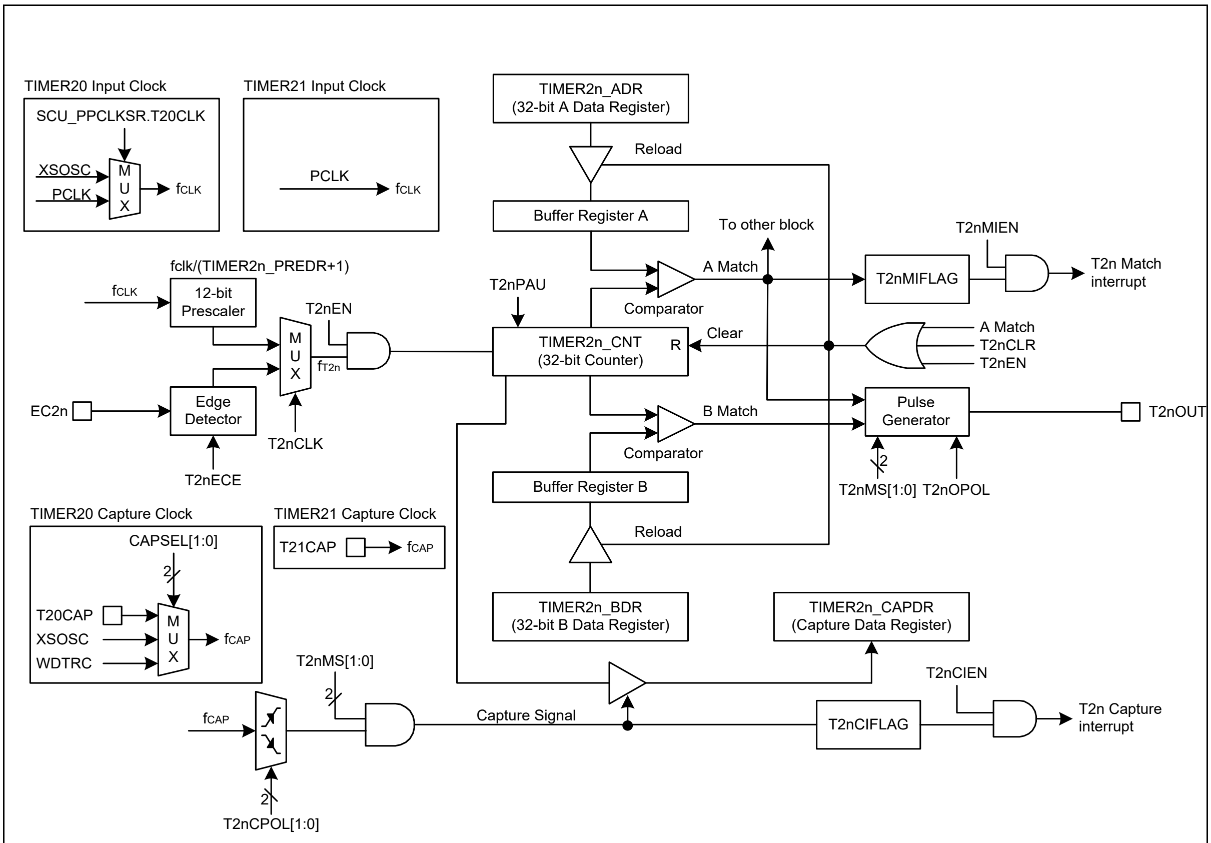
Timer counter 20/21 includes the features below:

- 32-bit up-counter and 12-bit prescaler
- Four operating modes:
  - Periodic Mode
  - One-shot Mode
  - PWM Mode
  - Capture Mode
- Various interrupts:
  - Match interrupt
  - Capture interrupt
- Synchronous start and clear function

11.3 Timer Counter 20/21 Functional Description

11.3.1 Block Diagram

Figure 43. Timer Counter 2n Timer Block Diagram (n = 0, 1)



NOTES:

1. T2nEN is automatically cleared to logic ‘0’ after one pulse is generated in PPG one-shot mode.
2. After T2nEN bit is set, counting of TIMER2n_CNT may be delayed by two clocks or more.

11.4 Pin Description for Timer Counter 20/21

Table 55. Pins and External Signals for Timer Counter 2n (n = 0, 1)

Pin Name	Type	Description
EC2n	I	External clock input
T2nCAP	I	Capture input
T2nOUT	O	PWM/one-shot output

11.5 Functional Description

11.5.1 Timer Counter 20/21

The Timer/Counter 2n (n = 0, 1) can use an internal or an external clock as a clock source (EC2n). A clock selection logic selects the clock source, and the clock selection logic is controlled by clock selection bits (T2nCLK).

- TIMER 2n clock sources are listed as followings (n = 0, 1):
  - PCLK / (TIMER2n_PREDR +1)
  - XSOSC / (TIMER2n_PREDR +1) (TIMER20 only)
  - EC2n

In capture mode, by T2nCAP, XSOSC or WDTRC (XSOSC and WDTRC: TIMER20 only) data is captured into input capture data register (TIMER2n_CAPDR). Timer 2n outputs the comparison results between counter and data register through T2nOUT port in Timer/counter mode. In addition, Timer 2n outputs PWM waveform through T2nOUT port in PPG mode.

Table 56. Timer 2n Operating Modes (n = 0, 1)

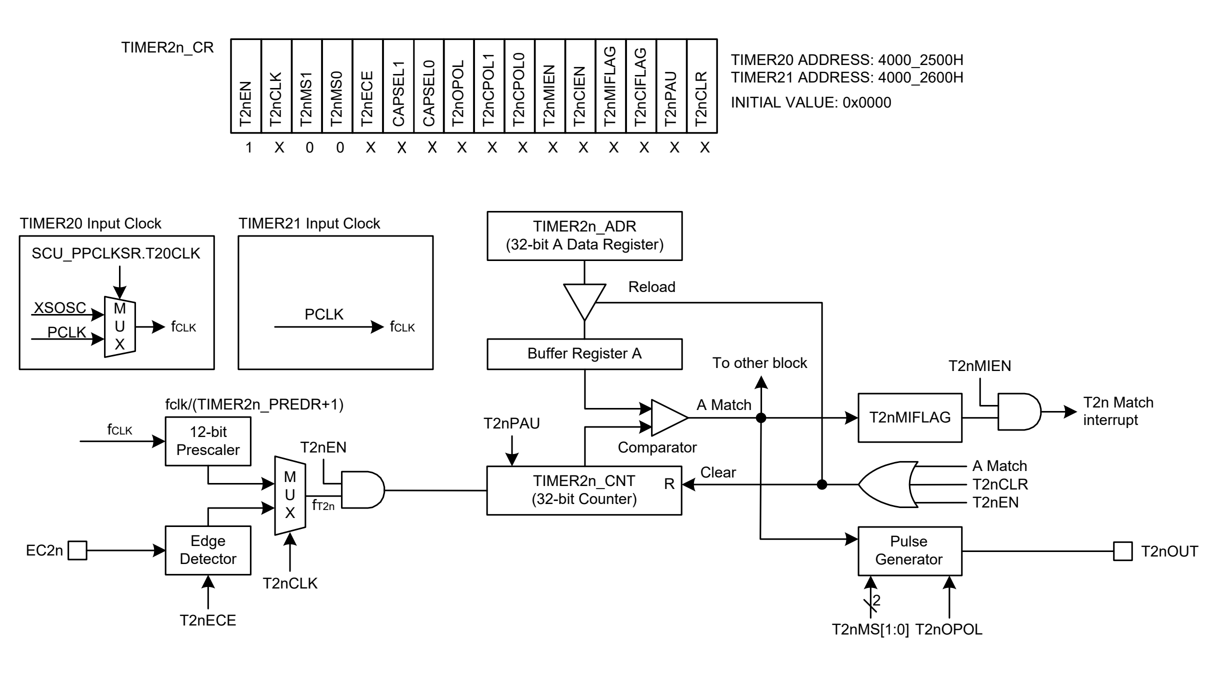
T2nEN	Alternative Mode	T2nMS	TIMER2n_PREDR	Timer 2n
1	T2nOUT	00	0xXXX	32-bit Timer/Counter Mode
1	T2nCAP	01	0xXXX	32-bit Capture Mode
1	T2nOUT	10	0xXXX	32-bit PPG Mode (one-shot mode)
1	T2nOUT	11	0xXXX	32-bit PPG Mode (repeat mode)

11.5.2 32-bit Timer/Counter Mode

32-bit Timer/counter mode is selected by control register as shown in Figure 44. The 32-bit timer has a counter register and a data register. The counter register is increased by internal or external clock input. Timer 2n can use an input clock with 12-bit prescaler division rates (TIMER2n_PREDR) and an external Clock (EC2n). When the values of TIMER2n_CNT and TIMER2n_ADR are the same in the timer 2n, a match signal is generated and the interrupt of Timer 2n takes place. The TIMER2n_CNT values are automatically cleared by match signal. It can also be cleared by software (T2nCLR) (n = 0, 1).

The TIMER2n_CNT values are automatically cleared by match signal. It can also be cleared by software (T2nCLR).

Figure 44. 32-bit Timer/Counter Mode for Timer 2n (n = 0, 1)

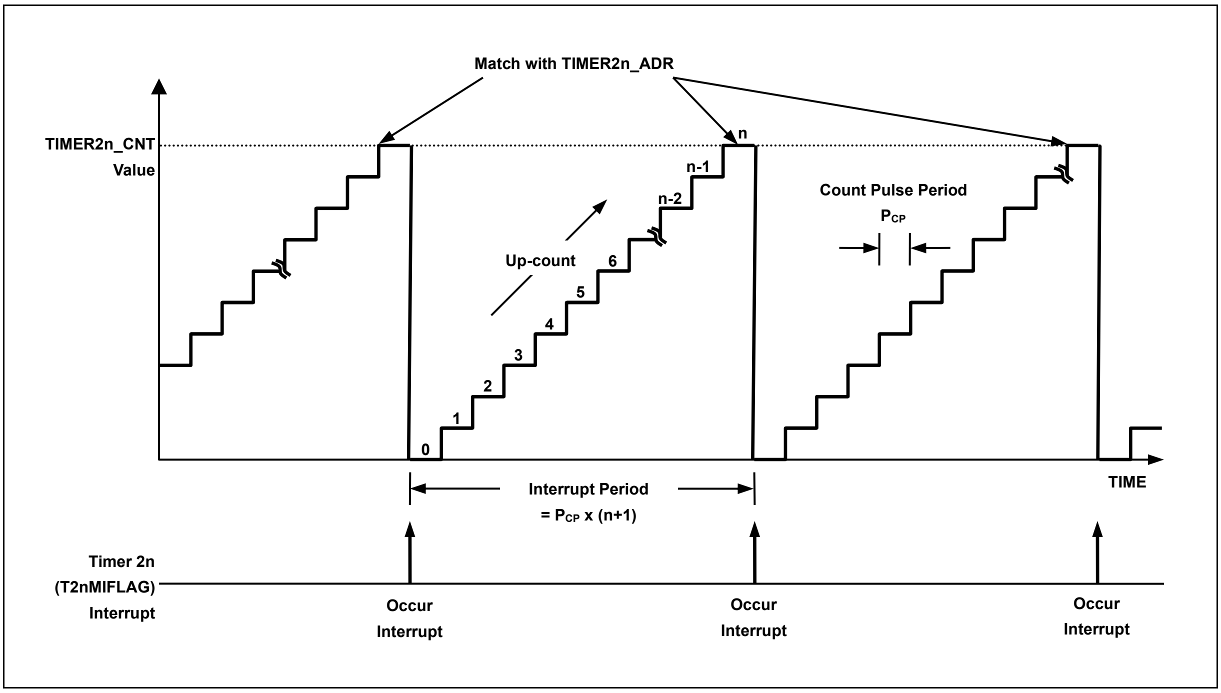


TIMER2n_CR

T2nEN	T2nCLK	T2nMS1	T2nMS0	T2nECE	CAPSEL1	CAPSEL0	T2nOPOL	T2nCPOL1	T2nCPOL0	T2nMIEN	T2nCIEN	T2nMIFLAG	T2nCIFLAG	T2nPAU	T2nCLR
1	X	0	0	X	X	X	X	X	X	X	X	X	X	X	X

TIMER20 ADDRESS: 4000_2500H
TIMER21 ADDRESS: 4000_2600H
INITIAL VALUE: 0x0000

Figure 45. 32-bit Timer/Counter 2n Example (n = 0, 1)



11.5.3 32-bit Capture Mode

Timer 2n Capture mode is evoked by configuring T2nMS[1:0] as “01”. The internal clock can be used as a clock source. It basically has the same function as the 32-bit timer/counter mode and an interrupt takes place when TIMER2n_CNT becomes equal to TIMER2n_ADR. TIMER2n_CNT values are cleared by software (T2nCLR) (n = 0, 1).

This timer interrupt in Capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. The capture result is loaded into TIMER2n_CAPDR. In the Timer 2n capture mode, Timer 2n output (T2nOUT) waveform is not available.

Figure 46. 32-bit Capture Mode for Timer 2n (n = 0, 1)

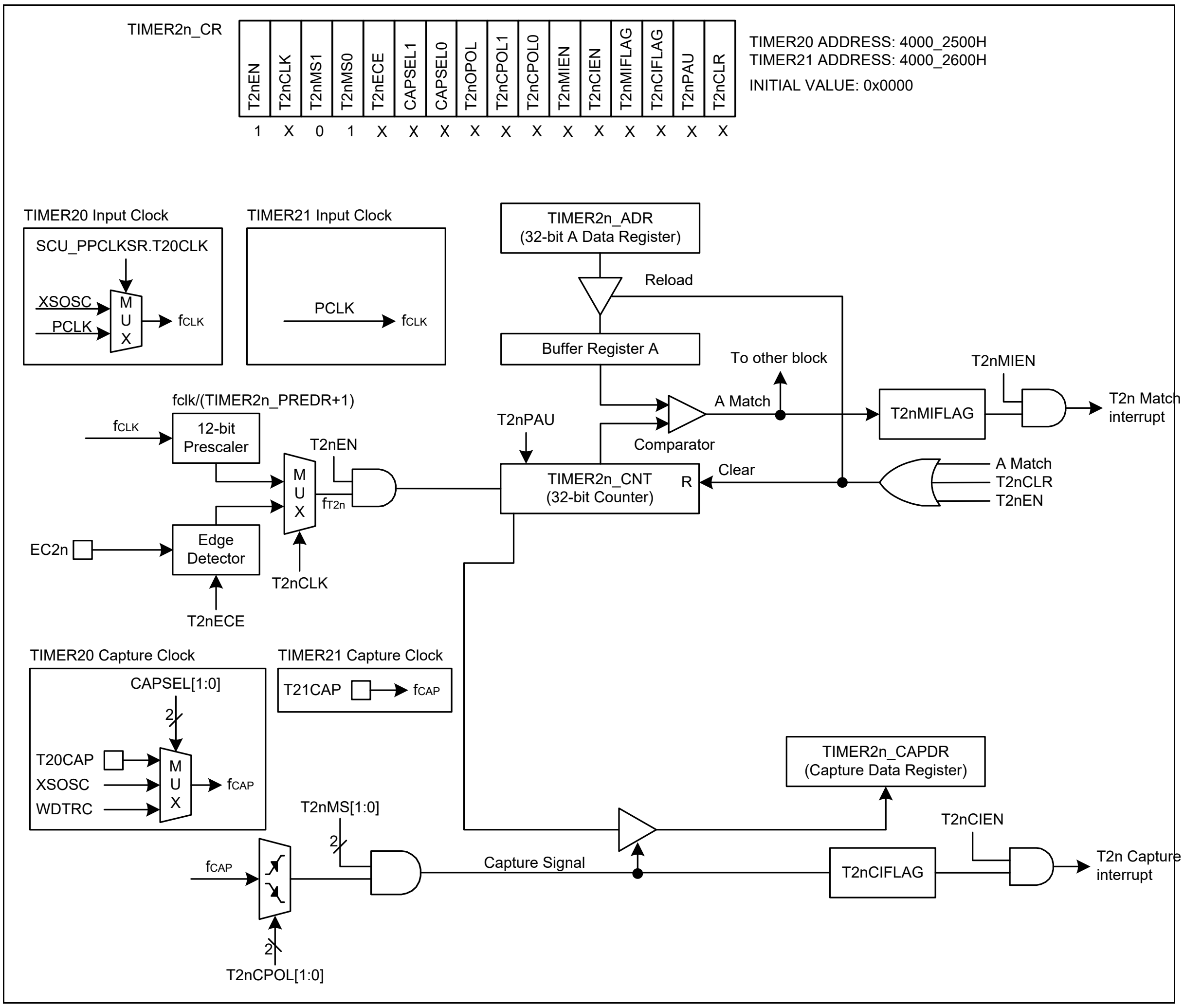


Figure 47. 32-bit Capture Mode for Timer 2n (n = 0, 1)

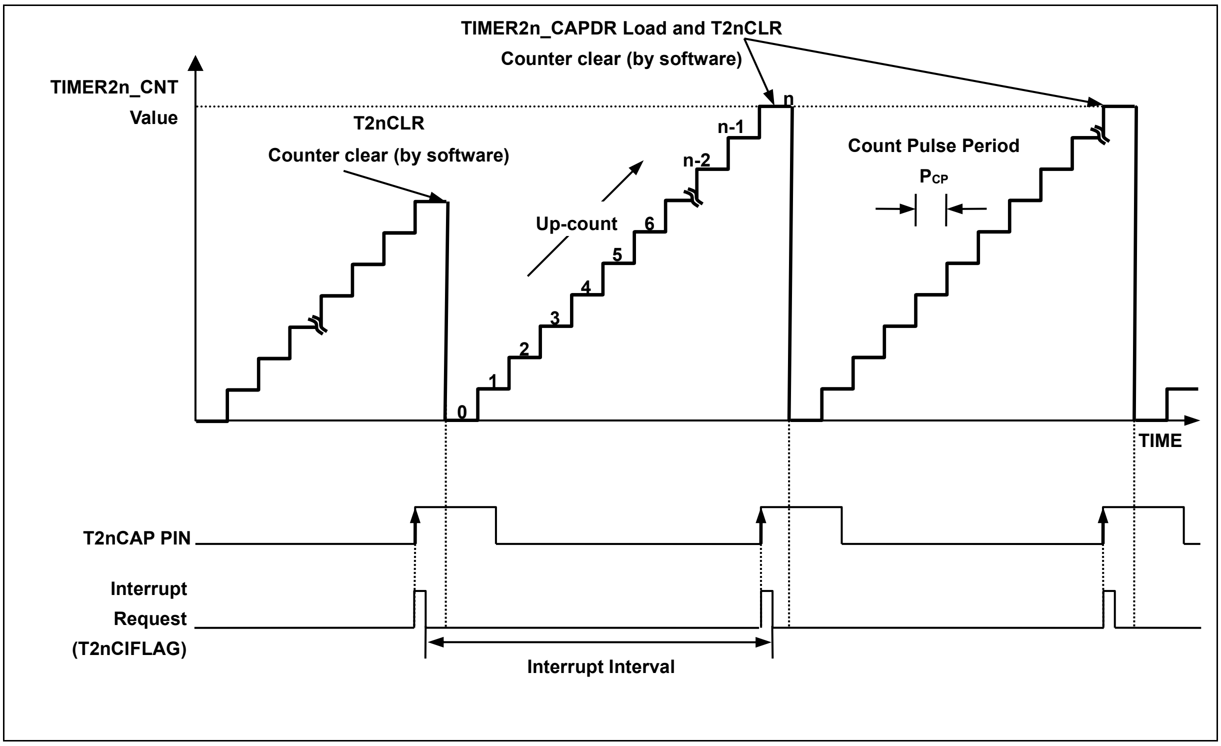
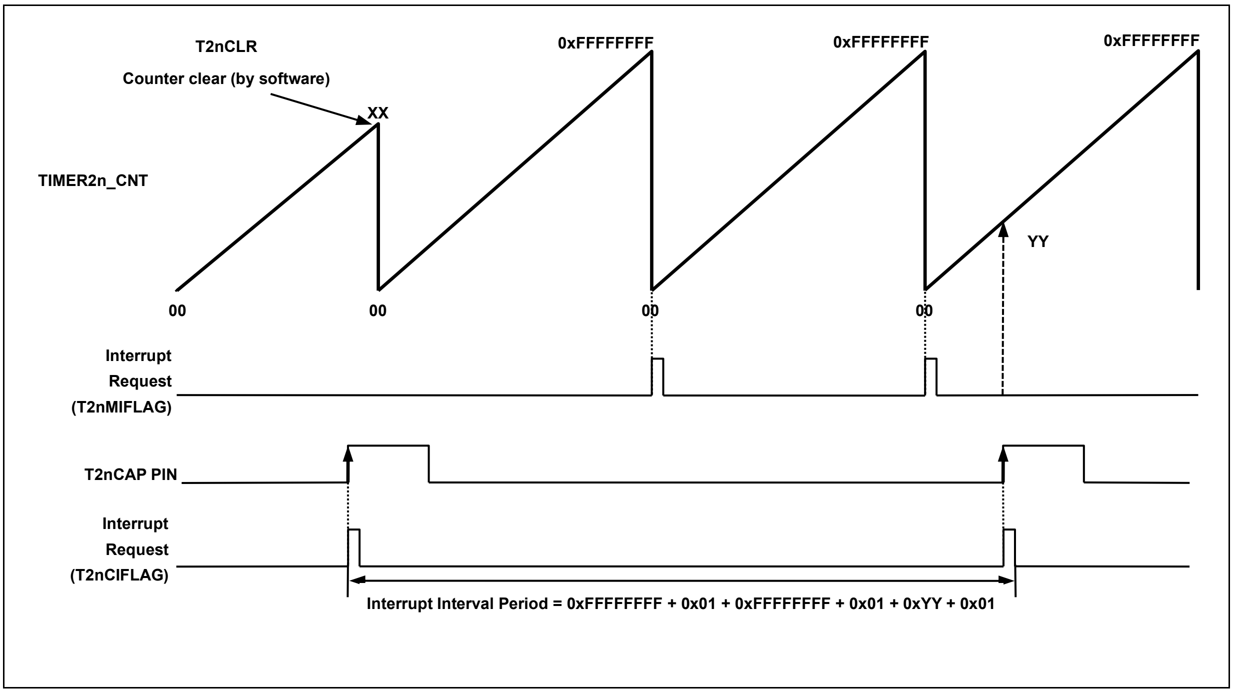


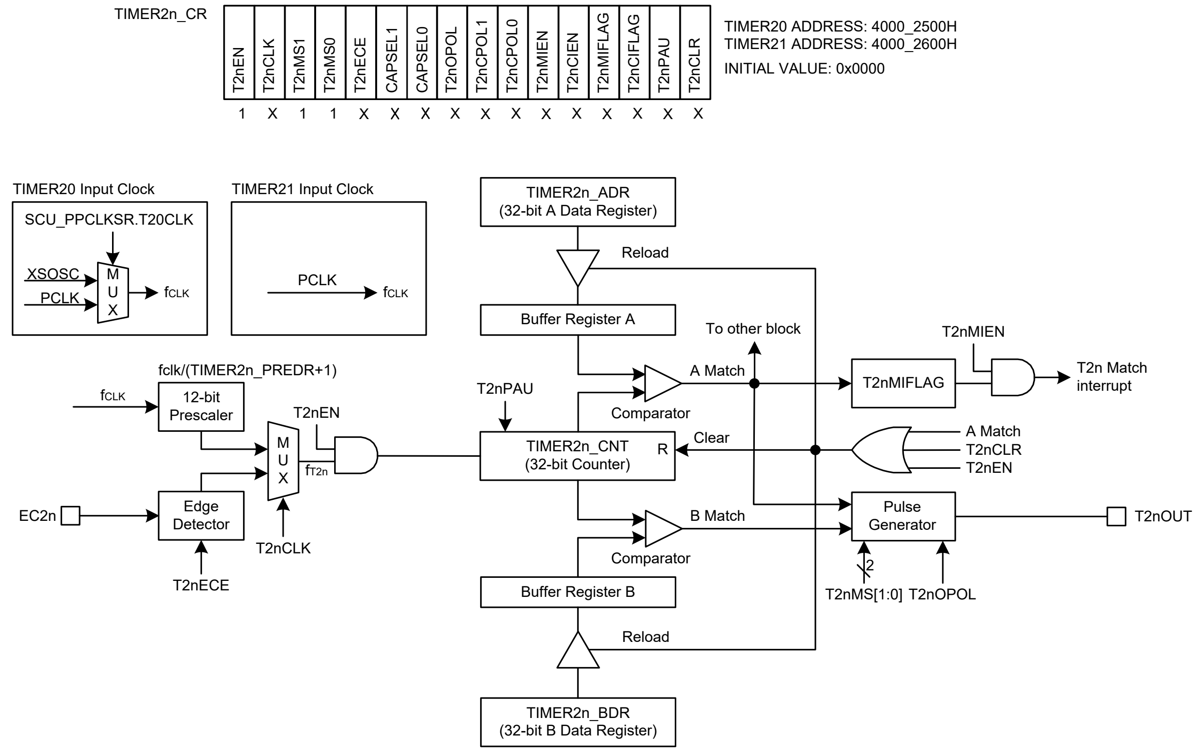
Figure 48. Express Timer Overflow in Capture Mode



11.5.4 32-bit PPG Mode

The Timer 2n has a PPG (Programmable Pulse Generation) function. In PPG mode, the T2nOUT pin generates PWM output of up to 32-bit resolution. This pin should be configured as a PWM output by setting ports alternative function selection register. The period of PWM output is determined by the TIMER2n_ADR register value. The duty of PWM output is determined by the TIMER2n_BDR register value.

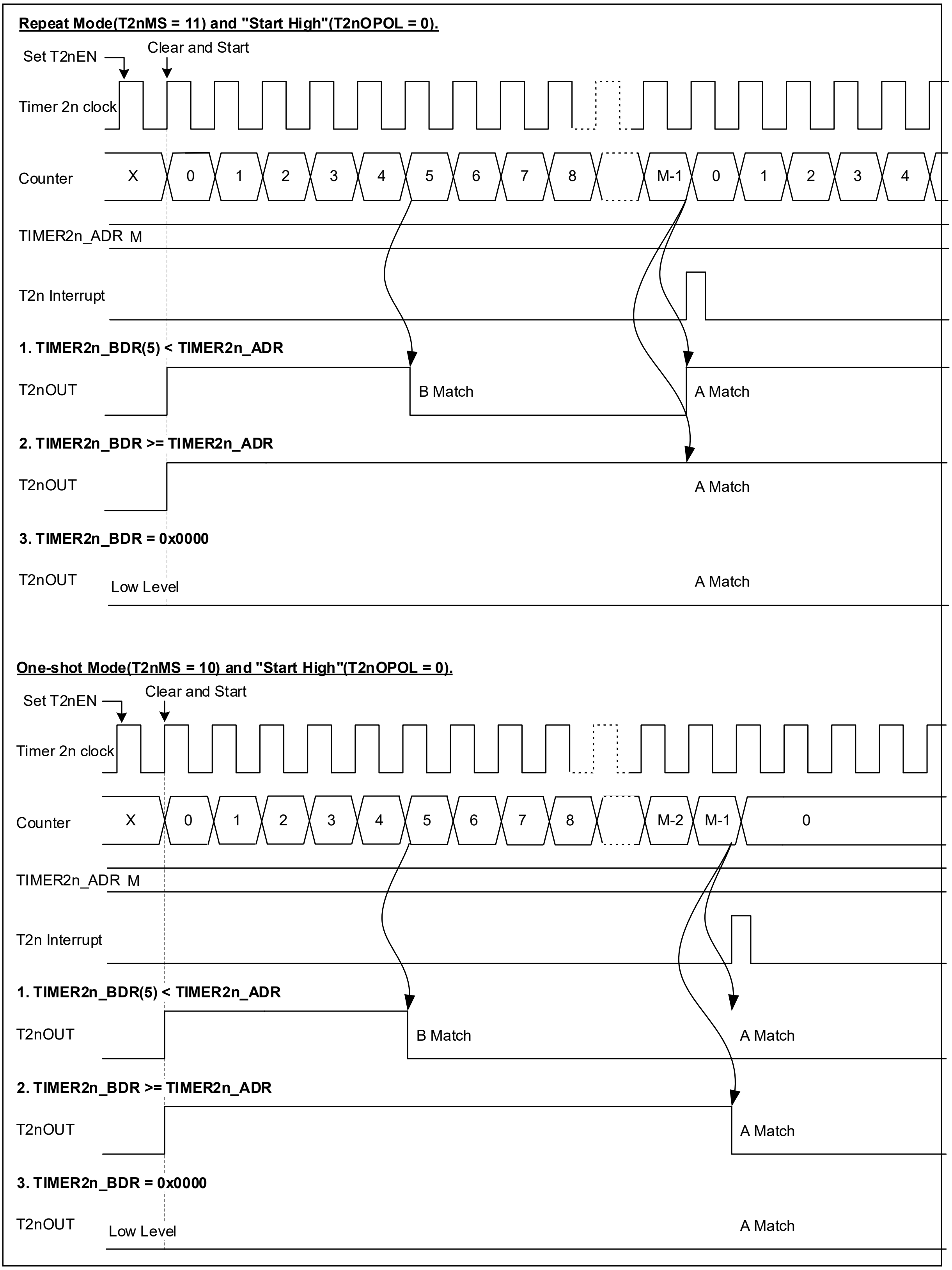
Figure 49. 32-bit PPG Repeat and One-shot Mode for Timer 2n (n = 0, 1)



NOTE:

1. The T2nEN bit is automatically cleared to logic ‘0’ after one pulse is generated in PPG one-shot mode (n = 0, 1).

Figure 50. 32-bit PPG Mode Timing chart for Timer 2n (n = 0, 1)



11.6 TIMER 20/21 Registers

The base address of the Timer 20/21 is described in the followings:

Table 57. Base Address of Timer 20/21

Name	Base Address
TIMER20	0x4000_2500
TIMER21	0x4000_2600

Table 58. TIMER20/21 Register Map

Name	Offset	Type	Description	Reset Value	Reference
TIMER2n_CR	0x00	RW	Timer/Counter 2n Control Register	0x0000_0000	11.6.1
TIMER2n_ADR	0x04	RW	Timer/Counter 2n A Data Register	0xFFFF_FFFF	11.6.2
TIMER2n_BDR	0x08	RW	Timer/Counter 2n B Data Register	0xFFFF_FFFF	11.6.3
TIMER2n_CAPDR	0x0C	RO	Timer/Counter 2n Capture Data Register	0x0000_0000	11.6.4
TIMER2n_PREDR	0x10	RW	Timer/Counter 2n Prescaler Data Register	0x0000_0FFF	11.6.5
TIMER2n_CNT	0x14	RO	Timer/Counter 2n Counter Register	0x0000_0000	11.6.6

NOTE:

1. n = 0 and 1.

11.6.1 TIMER2n_CR: Timer/Counter 2n Control Register

The Timer module should be configured properly before running. Once the target purpose is defined, the timer can be configured in the TIMER2n_CR register. After configuring this register, a user can start or stop the timer function by using this register.

The TIMER2n_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_CR=0x4000_2500, TIMER21_CR=0x4000_2600

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																T2nEN	T2nCLK	T2nMS[1:0]		T2nECE	CAPSEL[1:0]		T2nOPOL	T2nCPOL[1:0]		T2nMIEN	T2nCIEN	T2nMIFLAG	T2nCIFLAG	T2nPAU	T2nCLR
-																0	0	00		0	00		0	00		0	0	0	0	0	0
-																RW	RW	RW		RW	RW		RW	RW		RW	RW	RW	RW	RW	RW

15	T2nEN	Timer 2n Operation Enable.	
		0	Disable Timer 2n operation.
		1	Enable Timer 2n operation. (Counter clear and start)
14	T2nCLK	Timer 2n Clock Selection.	
		0	Select an internal prescaler clock.
		1	Select an external clock.
		NOTE: 1. This bit should be changed while T2nEN bit is ‘0’.	
13 12	T2nMS[1:0]	Timer 2n Operation Mode Selection.	
		00	Timer/Counter mode. (T2nOUT: toggle at A-match)
		01	Capture mode. (The A-match interrupt can occur)
		10	PPG one-shot mode. (T2nOUT: Programmable pulse output)
		11	PPG repeat mode. (T2nOUT: Programmable pulse output)
		NOTE: 1. This bit should be changed while T2nEN bit is ‘0’.	
11	T2nECE	Timer 2n External Clock Edge Selection.	
		0	Select falling edge of external clock.
		1	Select rising edge of external clock.
10 9	CAPSEL[1:0]	Timer 2n Capture Signal Selection (TIMER20 only)	
		00	Select an external capture signal.
		01	Select the XSOSC (External sub oscillator) signal.
		10	Select the WDTRC (Watchdog timer RC oscillator) signal.
		11	Not used
		NOTE: 1. This bit should be changed while T20EN bit is ‘0’.	
8	T2nOPOL	T2nOUT Polarity Selection.	
		0	Start high. (T2nOUT is low level at disable)
		1	Start low. (T2nOUT is high level at disable)
7 6	T2nCPOL[1:0]	Timer 2n Capture Polarity Selection.	
		00	Capture on falling edge.
		01	Capture on rising edge.
		10	Capture on both falling and rising edge.

		11	Reserved.
5	T2nMIEN	Timer 2n Match Interrupt Enable.	
		0	Disable Timer 2n match interrupt.
		1	Enable Timer 2n match interrupt.
4	T2nCIEN	Timer 2n Capture Interrupt Enable.	
		0	Disable Timer 2n capture interrupt.
		1	Enable Timer 2n capture interrupt.
3	T2nMIFLAG	Timer 2n Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
2	T2nCIFLAG	Timer 2n Capture Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
1	T2nPAU	Timer 2n Counter Temporary Pause Control bit.	
		0	Continue counting.
		1	Temporary pause.
0	T2nCLR	Timer 2n Counter and Prescaler Clear bit.	
		0	No effect.
		1	Clear Timer 2n counter and prescaler (automatically cleared to ‘0’ after operation).

11.6.2 TIMER2n_ADR: Timer/Counter 2n A Data Register

The TIMER2n_ADR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_ADR=0x4000_2504, TIMER21_ADR=0x4000_2604

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADATA[31:0]																															
0xFFFFFFFF																															
RW																															

31 0	ADATA[31:0]	Timer/Counter 2n A Data. The range is 0x00000002 to 0xFFFFFFFF. (n = 0, 1)

NOTE:

1. Do not write “0x00000000” in the TIMER2n_ADR register under PPG mode.

11.6.3 TIMER2n_BDR: Timer/Counter 2n B Data Register

The TIMER2n_BDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_BDR=0x4000_2508, TIMER21_BDR=0x4000_2608

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDATA[31:0]																															
0xFFFFFFFF																															
RW																															

31 0	BDATA[31:0]	Timer/Counter 2n B Data. The range is 0x00000000 to 0xFFFFFFFF.

11.6.4 TIMER2n_CAPDR: Timer/Counter 2n Capture Data Register

The TIMER2n_CAPDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_CAPDR=0x4000_250C, TIMER21_CAPDR=0x4000_260C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPD[31:0]																															
0x00000000																															
RO																															

31 0	CAPD[31:0]	Timer/Counter 2n Capture Data (n = 0, 1).

11.6.5 TIMER2n_PREDR: Timer/Counter 2n Prescaler Data Register

The TIMER2n_PREDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_PREDR=0x4000_2510, TIMER21_PREDR=0x4000_2610

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				PRED[11:0]											
-																				0xFFF											
-																				RW											

11 0	PRED[11:0]	Timer/Counter 2n Prescaler Data (n = 0, 1).

11.6.6 TIMER2n_CNT: Timer/Counter 2n Counter Register

The TIMER2n_CNT register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_CNT=0x4000_2514, TIMER21_CNT=0x4000_2614

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[31:0]																															
0x00000000																															
RO																															

31 0	CNT[31:0]	Timer/Counter 2n Counter (n = 0, 1).

11.6.7 TIMER20 Register Map Summary

Table 59. TIMER20 Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMER20_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T20EN	T20CLK	T20MS[1:0]		T20ECE	CAPSEL[1:0]		T20OPOL	T20CPOL[1:0]		T20MIEN	T20CIEN	T20MIFLAG	T20CIFLAG	T20PAU	T20CLR
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	TIMER20_ADR	ADATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	TIMER20_BDR	BDATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	TIMER20_CAPDR	CAPD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	TIMER20_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	TIMER20_CNT	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.8 TIMER21 Register Map Summary

Table 60. TIMER21 Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMER21_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	T21EN	T21CLK	T21MS[1:0]		T21ECE	Res	Res	T21OPOL	T21CPOL[1:0]		T21MIEN	T21CIEN	T21MIFLAG	T21CIFLAG	T21PAU	T21CLR
	Reset value																	0	0	0	0	0			0	0	0	0	0	0	0	0	0
0x04	TIMER21_ADR	ADATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	TIMER21_BDR	BDATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	TIMER21_CAPDR	CAPD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	TIMER21_PREDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRED[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	TIMER21_CNT	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12. USART (UART and SPI)

12.1 Introduction

USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter) is a flexible serial communication device. USART supports UART and SPI modes. The A31S134 has three channels of USART module.

12.2 Main Features

The USART of A31S134 has the following features:

- Full-Duplex operation (independent serial receive and transmit registers)
- Asynchronous or synchronous operation
- Baud-rate generator
- Supports serial frames with 5, 6, 7, 8, or 9 data bits and 1 or 2 stop bits
- Odd or even parity generation, and parity check supported by hardware.
- Supports receive character detection and receive time out function
- Supports Local-Interconnection Network (LIN)
- Data OverRun Detection
- Framing Error Detection
- Three separate interrupts on TX completion, TX data register empty and RX completion
- Double Speed Asynchronous communication mode
- Up to 8 MHz data transfer for SPI

12.3 USART 10/11/12 Block Diagram

Figure 51 shows a block diagram of the UART and LIN block.

Figure 51. UART and LIN Block Diagram of USART (n = 10, 11, and 12)

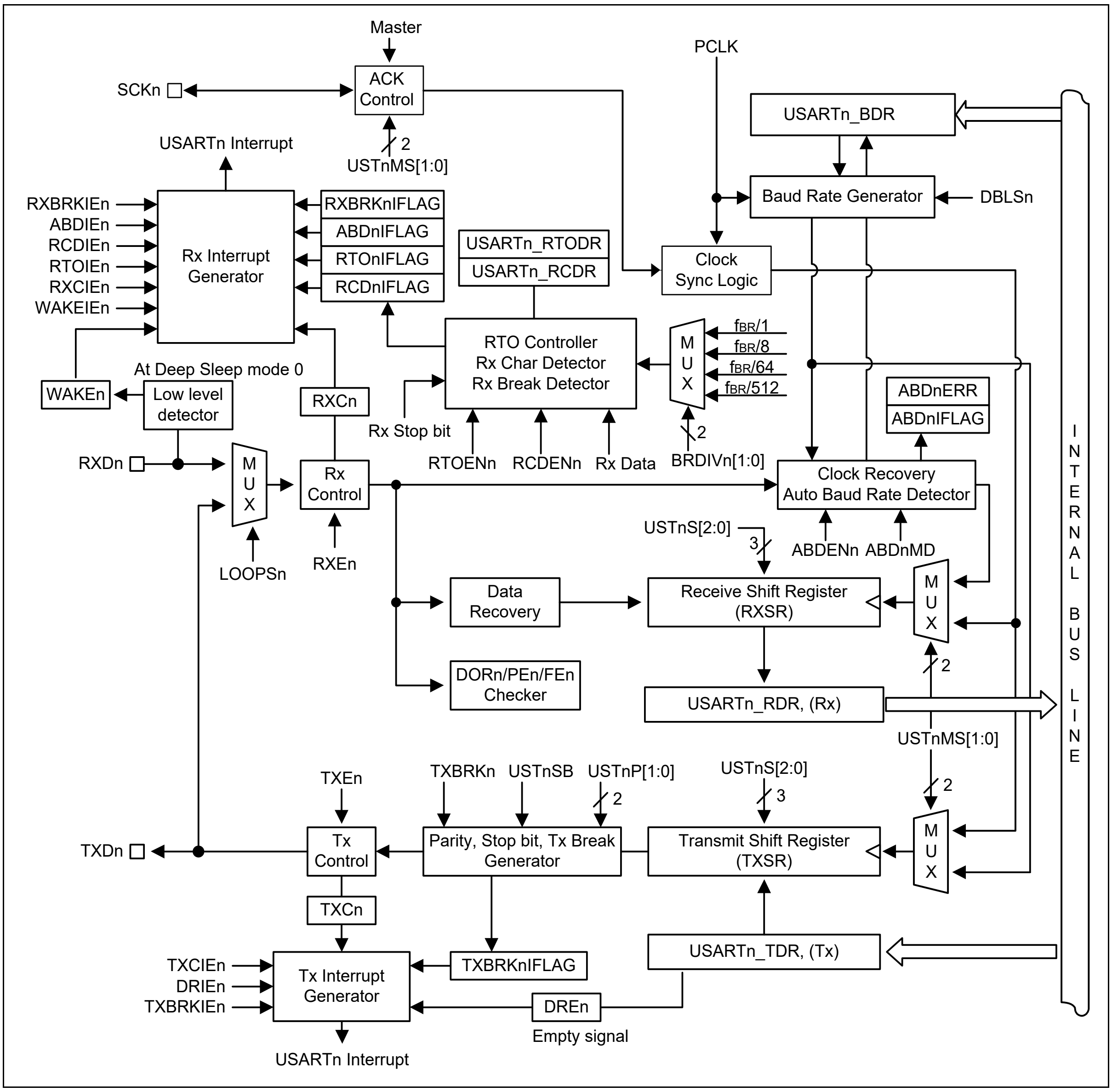
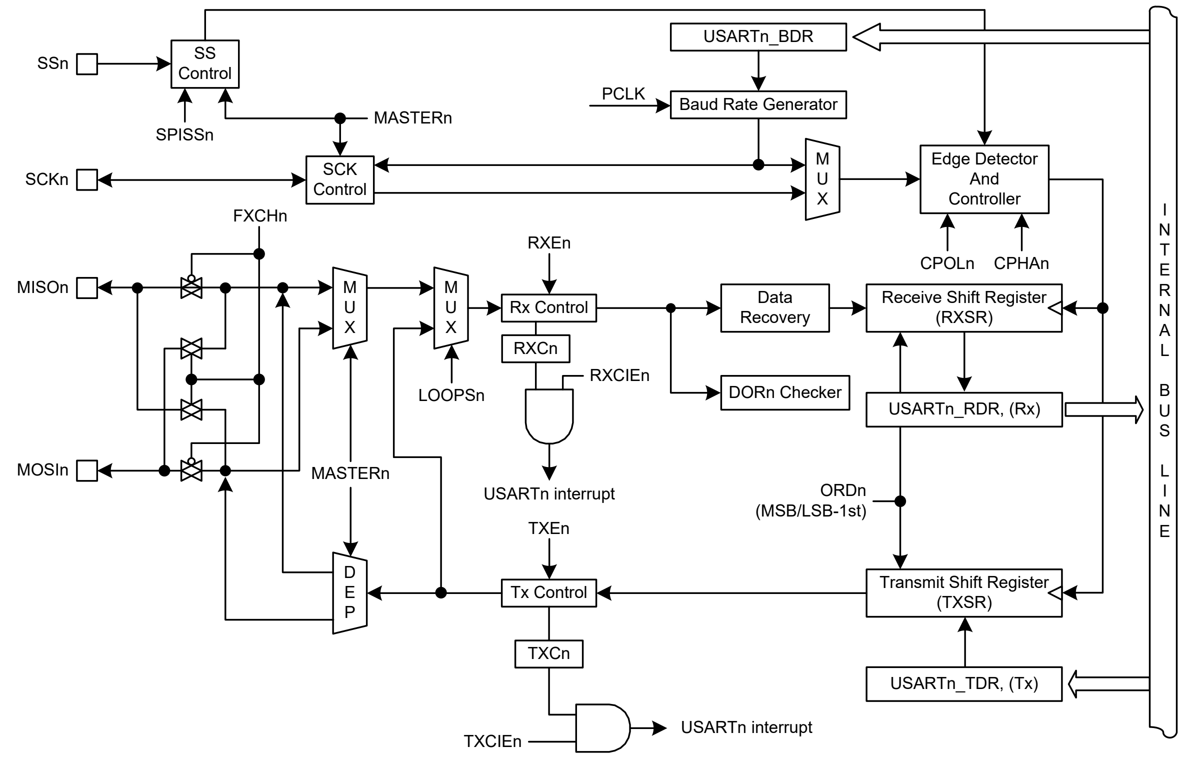


Figure 52 shows a block diagram of the SPI block.

Figure 52. SPIn Block Diagram of USART (n = 10, 11, and 12)



NOTES:

1. The clock frequency of SCKn should be less than or equal to PCLK/2.
2. When DMA is used, the interval of data to be received requires at least 32 clocks of PCLK.

12.4 Pin Description for USART 10/11/12

Table 61. Pins and External Signals for USART 10/11/12

Pin Name	Type	Description
TXDn	O	UART channel n transmit output
RXDn	I	UART channel n receive input
SSn	I/O	SPIn slave select input /output
SCKn	I/O	SPIn serial clock input/output
MOSIn	I/O	SPIn serial data (master output, slave input)
MISOn	I/O	SPIn serial data (master input, slave output)

12.5 UART Functional Description

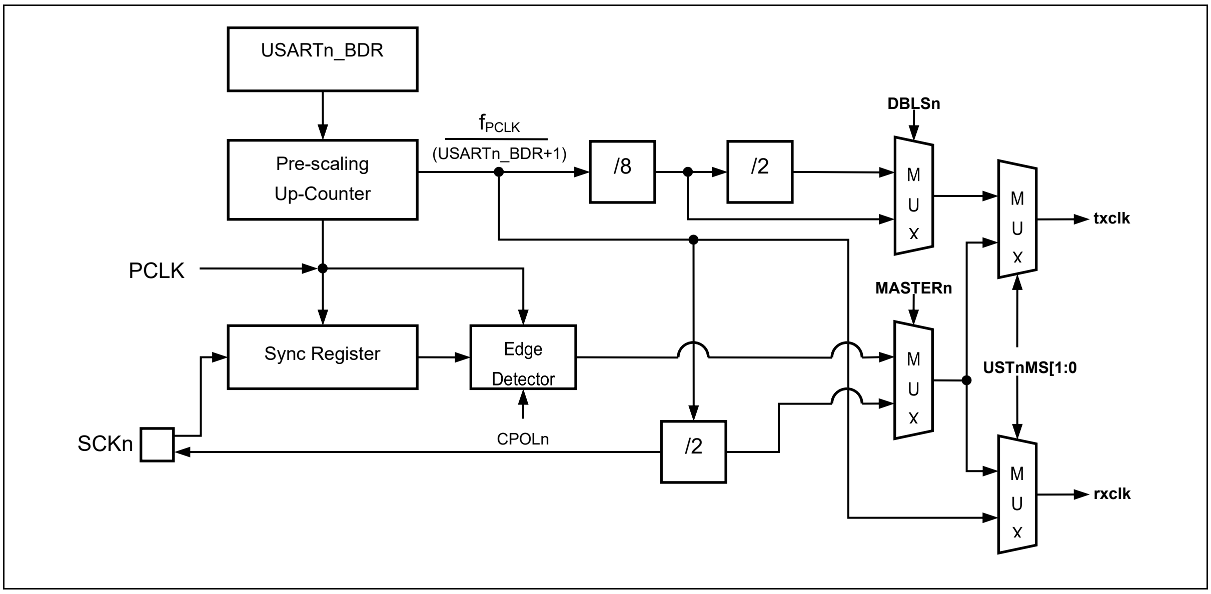
The USART comprises a clock generator, a transmitter, and a receiver. The clock generation logic includes synchronization logic for external clock input, which is used for synchronizing or SPI slave operation. Baud rate generator in the clock generation logic is for asynchronous or master (synchronous or SPI) operation.

The Transmitter consists of a write buffer, a serial shift register, a parity generator, and a control logic. Using DMA allows continuous transfer of data without any software involvement between frames.

The receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition, the receiver has a parity checker, a shift register, and a control logic. The receiver supports the same frame formats as the transmitter and can detect frame errors, data overrun and parity errors (n = 10, 11 and 12).

12.5.1 USART Clock Generation

Figure 53. Clock Generation Block Diagram (USART, n = 10, 11, and 12)



The clock generation logic generates the base clock for the transmitter and receiver. The USART supports four modes of clock operation, which are Normal asynchronous mode, Double speed asynchronous mode, Master synchronous mode and Slave synchronous mode.

The clock generation scheme for master SPI and slave SPI mode is the same as master synchronous and slave synchronous operation mode. The USTnMS[1:0] bit field in the USARTn_CR1 register selects asynchronous or synchronous operation. Asynchronous double speed mode is controlled by the DBLS bit in the USARTn_CR2 register.

The MASTER bit in USARTn_CR2 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCKn pin is active only when the USART operates in synchronous or SPI mode.

Table 62 shows the equations for calculating the baud rate (in bps).

Table 62. Equations for Calculating USART Baud Rate Register Settings (n = 10, 11, and 12)

Operating Mode	Equation for Calculating Baud-rate
Asynchronous Normal Mode (DBLSn=0)	Baud Rate = PCLK/(16(USARTn_BDR+1))
Asynchronous Double Speed Mode (DBLSn=1)	Baud Rate = PCLK/(8(USARTn_BDR+1))
Synchronous or SPI Master Mode	Baud Rate = PCLK/(2(USARTn_BDR+1))

12.5.2 External Clock (SCKn)

External clock is used in the Synchronous mode or in the SPI slave mode. External clock input from the SCKn pin is sampled by the synchronization logic to remove meta-stability. The output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and the receiver.

This process describes two CPU clock period delay. The maximum frequency of the external SCKn pin is limited to 8 MHz.

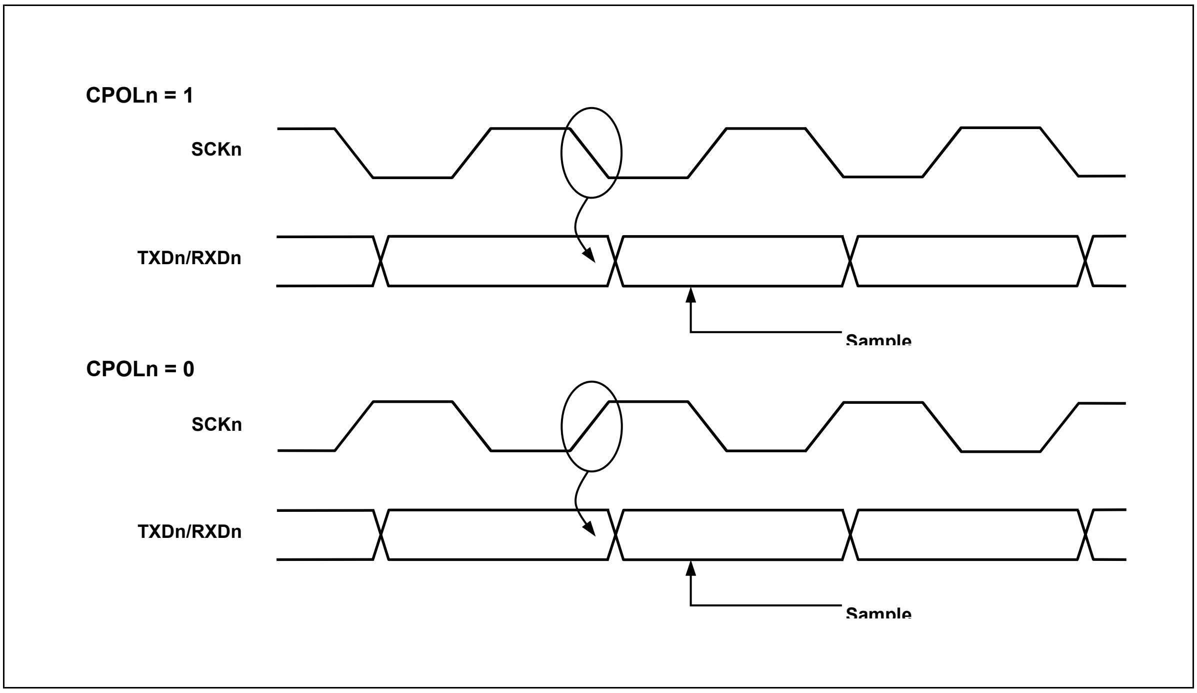
12.5.3 Synchronous Mode Operation

External clock is used in the Synchronous mode or in the SPI slave mode. When the Synchronous or the SPI mode is used, the SCKn pin will be used as either clock input (slave) or clock output (master).

Data sampling and transmission are issued on different edges of SCKn clock respectively. For example, if data input on RXDn (MISOn in SPI mode) pin is sampled on the rising edge of SCKn clock, data output on TXDn (MOSIn in SPI mode) pin is altered on the falling edge.

The CPOLn bit in the USARTn_CR1 register selects which SCKn clock edge is used for data sampling and which is used for data change. As shown in Figure 54 below, when the CPOLn bit is ‘0’, the data will be changed at rising SCKn edge and sampled at falling SCKn edge.

Figure 54. Synchronous Mode SCKn Timing (USART, n = 10, 11, and 12)



12.5.4 UART Data Format

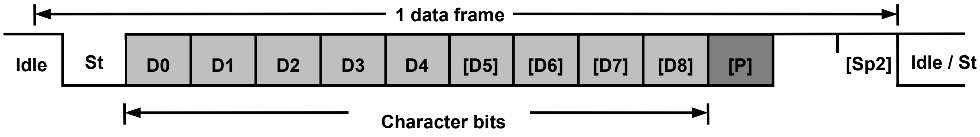
A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection. USART supports all 30 combinations of the following as valid frame formats.

- One start bit
- 5, 6, 7, 8 or 9 data bits
- No, even or odd parity bit.
- One or two stop bits.

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to nine, follow, ending with the most significant bit (MSB). If a parity function is enabled, the parity bit is inserted between the last data bit and the stop bit. A high-to-low transition on data pin is considered as a start bit.

When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle state means high state of data pin. Figure 55 shows a possible combination of the frame formats. Bits inside brackets marked as [Sp2] are option of corresponding bits.

Figure 55. Frame Format (UART)



A data frame consists of the following bits:

- Idle: No signal on communication line (TXDn/RXDn)
- St: Start bit (low-state)
- Dm: Data bits (bit-0 to -8)
- P: Parity bit (even parity, odd parity, no parity)
- Sp: Stop bit (one bit or two bits)

The frame format used by the UART is set by the USTnS[2:0], USTnP[1:0] bits in the USARTn_CR1 register and the USTnSB bit in the USARTn_CR2 register. The transmitter and the receiver use the same values (n = 10, 11 and 12).

12.5.5 UART Parity Bit

The parity bit is calculated by doing an exclusive-OR of all data bits. If odd parity is used, the result of the exclusive-OR is inverted. The parity bit is located between the MSB and the first stop bit of a serial frame.

- Peven = Dm-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 0
- Podd = Dm-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 1
- Peven: Parity bit using even parity
- Podd: Parity bit using odd parity
- Dm: Data bit n of the character

12.5.6 UART Transmitter

The UART transmitter is enabled by setting the TXEn bit in the USARTn_CR1 register. When the Transmitter is enabled, the TXDn pin should be set to TXDn function for the serial output pin in UART mode by the GPIO registers. Baud-rate, operation mode and frame format must be set up before starting any transmission. In Synchronous operation mode, the SCKn pin is used for transmission clock, so it should be selected to do SCKn function by the GPIO registers (n = 10, 11 and 12).

12.5.6.1 UART Sending TX Data

A data transmission is initiated by loading data to the transmit data register (USARTn_TDR). The data to be written in the transmit data register is moved to the shift register when the shift register is ready to send a new frame.

The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded to the new data, it will transfer one complete frame according to the settings of the control registers.

If 9-bit characters are used in the Asynchronous or the Synchronous operation mode, the 9th bit must be written to the TDATA[8] bit in the USARTn_TDR register (n = 10, 11, and 12).

12.5.6.2 UART Transmitter Flag and Interrupt

The UART transmitter has two flags that indicate its state. One is USART data register empty flag (DREn) and the other is transmit completion flag (TXCn). Both flags can be used as interrupt sources.

DREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register.

When the data register empty interrupt enable (DRIEn) bit in the USARTnCR1 register is set and the global interrupt is enabled, the USARTn_ST status register empty interrupt is generated while the DREn flag is set.

Transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out. The TXCn flag can be cleared by writing ‘1’ to the TXCn bit in the USARTn_ST register.

When transmit complete interrupt enable (TXCIEn) bit in the USARTn_CR1 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while the TXCn flag is set. (n = 10, 11 and 12)

12.5.6.3 UART Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USTnP1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent (n = 10, 11, and 12).

12.5.6.4 UART Disabling Transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until the current transmission is completed. When the transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (GPIO) (n = 10, 11, and 12).

12.5.7 UART Receiver

The UART receiver is enabled by setting the RXEn bit in the USARTn_CR1 register. When the receiver is enabled, the RXDn pin should be set to RXDn function for the serial input pin in the UART mode by the GPIO registers. Baud-rate, operation mode and frame format must be set before serial reception. In Synchronous or SPI operation mode, the SCKn pin is used as a transfer clock input, so it should be selected to do SCKn function by the GPIO registers (n = 10, 11, and 12).

12.5.7.1 UART Receiving RX Data

When the UART is in Synchronous mode or in Asynchronous mode, the receiver starts data reception if it detects a valid start bit (LOW) on RXDn pin. Each bit after the start bit is sampled at predefined baud-rate (asynchronous) or at sampling edge of SCKn (synchronous) and shifted into the receive shift register until the first stop bit of a frame is received.

Even if there is a second stop bit in the frame, the second stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is presented in the receiver shift register and contents of the shift register are to be moved into the receive data register (USARTn_RDR) (n = 10, 11, and 12).

12.5.7.2 UART Receiver Flag and Interrupt

The UART receiver has a flag that indicates the receiver’s state. The receive complete (RXCn) flag indicates whether there is unread data in the receive buffer. This flag is set when there is unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When the receive complete interrupt enable (RXCIEn) bit in the USARTn_CR1 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXCn flag is set.

The UART receiver has three error flags, which are frame error (Fen), data overrun (DORn) and parity error (Pen). These error flags can be read from the USARTn_ST register.

The frame error (Fen) flag indicates the state of the first stop bit. The Fen flag is ‘0’ when the stop bit was correctly detected as ‘1’, while the Fen flag is ‘1’ when the stop bit was incorrect, i.e. detected as ‘0’. This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DORn) flag indicates data loss due to a full receive buffer condition. DORn occurs when the receive buffer is full, and another new data is presented in the receive shift register to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To avoid data loss or to clear this flag, the receive buffer must be read.

The parity error (Pen) flag indicates that the frame in the receive buffer had a parity error during reception. If parity check function is not enabled (USTnP1=0), the Pen bit is always read as ‘0’. (n = 10, 11, and 12)

12.5.7.3 UART Parity Checker

If parity bit is enabled (USTnP1=1), the parity checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame. (n = 10, 11 and 12)

12.5.7.4 UART Disabling Receiver

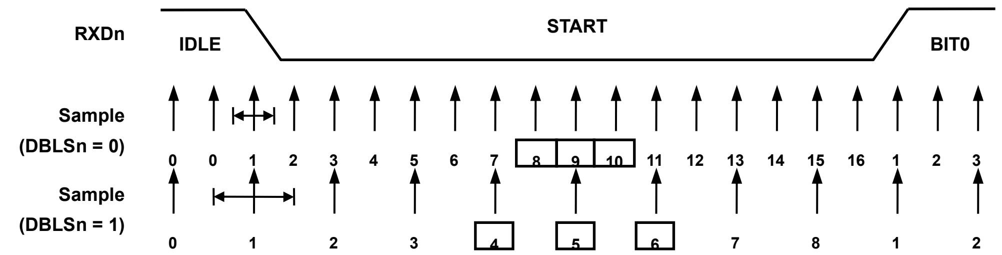
Unlike the transmitter, the receiver becomes inactive immediately after it is disabled by clearing RXEn bit. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (GPIO). (n = 10, 11, and 12)

12.5.7.5 Asynchronous Data Reception

To receive asynchronous data frame, the USART includes a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXDn pin. The data recovery logic samples and filters the incoming bits with a low pass filter and removes the noise of RXDn pin.

Figure 56 illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud-rate in normal mode and 8 times the baud-rate for double speed mode (DBLSn=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is seen using the double speed mode. (n = 10, 11, and 12)

Figure 56. Asynchronous Start Bit Sampling (n = 10, 11, and 12)



When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, which is the start bit condition. After detecting the high-to-low transition on RXDn line, the clock recovery logic uses samples 8, 9 and 10 for normal mode to detect whether valid start bit is received.

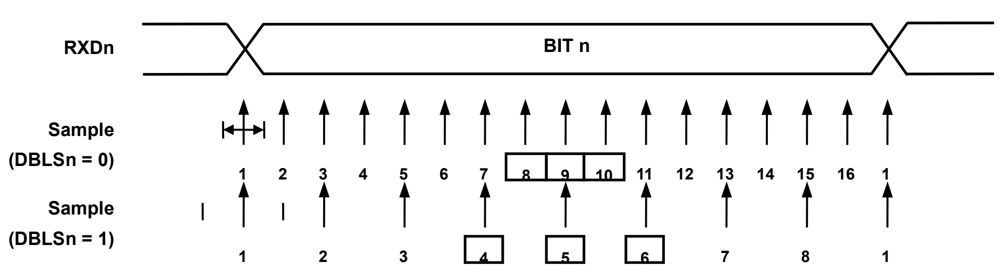
If more than 2 samples have a logical low level, it is considered that a valid start bit is detected, and the internally generated clock is synchronized to the incoming data frame. Then the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost the same as clock recovery process.

The data recovery logic samples each incoming bit 16 times for normal mode and 8 times for double speed mode, and uses sample 8, 9 and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered as a logic ‘0’ and if more than 2 samples have high levels, the received bit is considered as a logic ‘1’.

The data recovery process is then repeated until a complete frame is received, including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the receiver is in idle state and waits to find the start bit (n = 10, 11, and 12).

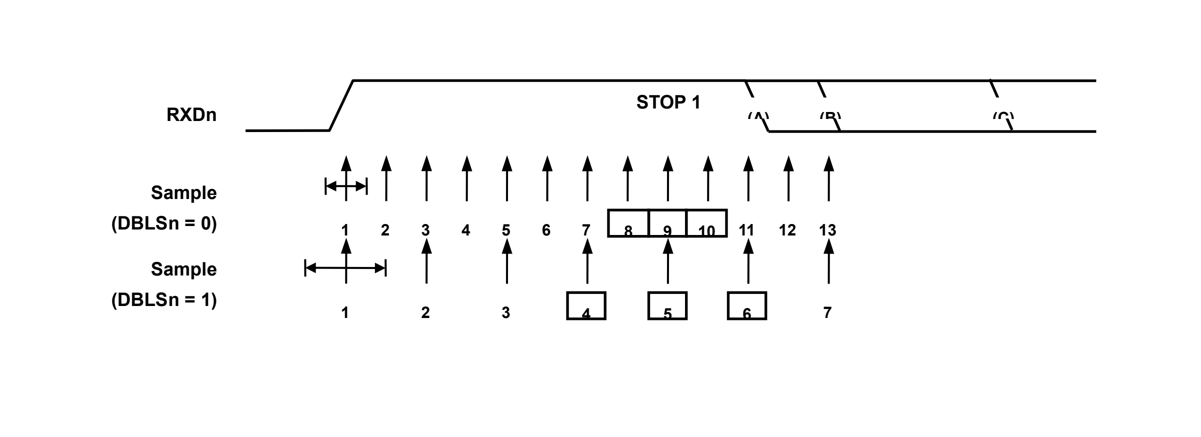
Figure 57. Asynchronous Sampling of Data and Parity Bit (n = 10, 11, and 12)



The process for detecting stop bit is the same as clock and data recovery process. That is, if two or more samples of three center values have high level, correct stop bit is detected, or else a frame error (Fen) flag is set.

After deciding whether the first stop bit is valid, the receiver goes to idle state and monitors the RXDn line to check whether a valid high to low transition is detected (start bit detection) (n = 10, 11, and 12).

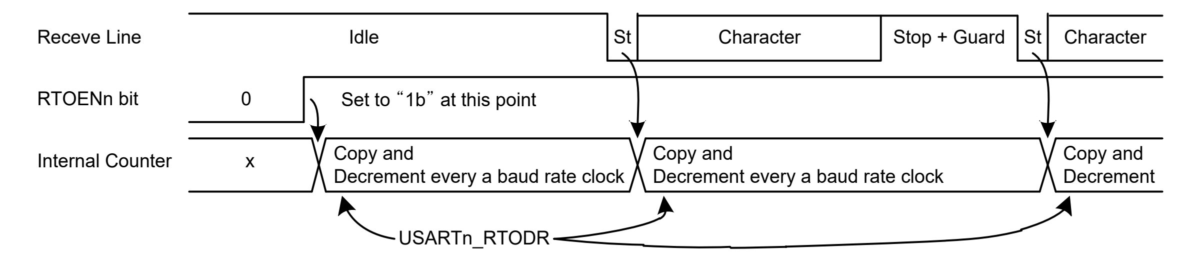
Figure 58. Stop Bit Sampling and Next Start Bit Sampling (n = 10, 11 and 12)



12.5.7.6 Receive Timeout Function

The receive timeout function is used for checking a frame finish. This function is to count time with baud rate unit between the last start bit and a new start bit, and between setting the RTOENn bit of the USARTn_CR3 register and a new start bit. The USARTn_RTODR register should have duration time value before using the receive time out function (n = 10, 11, and 12).

Figure 59. Receive Timeout Function (n = 10, 11, and 12)



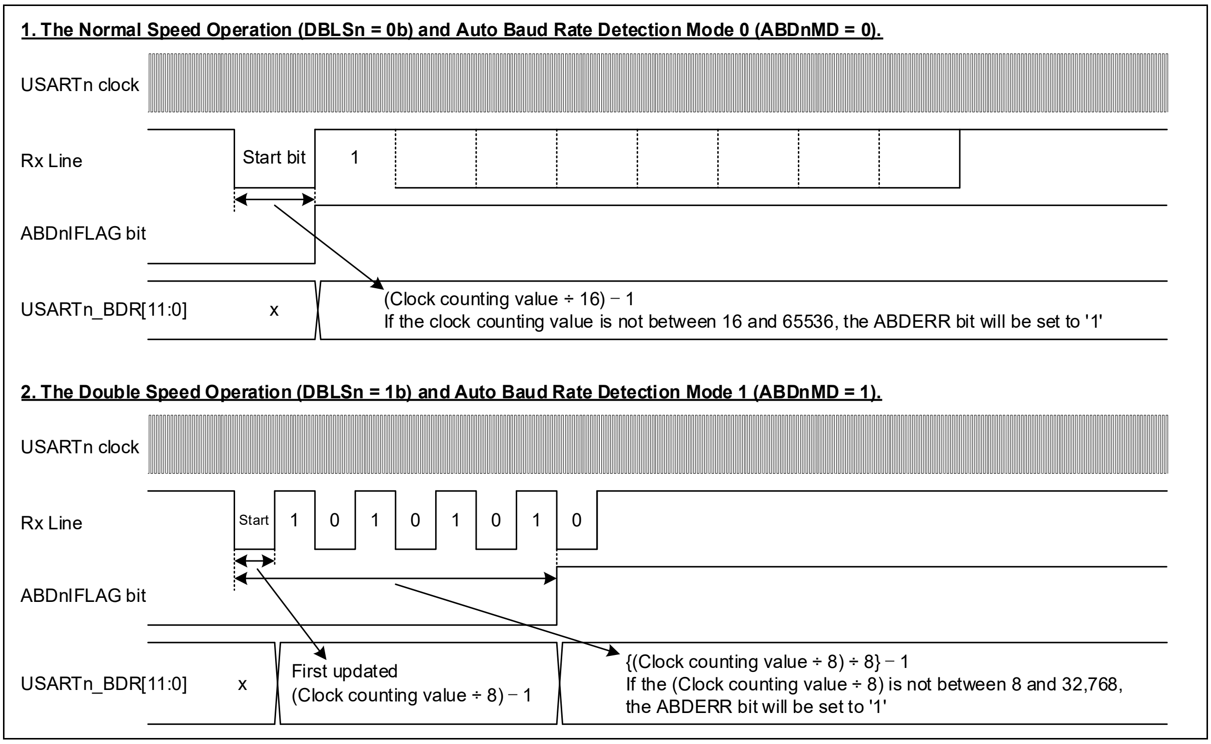
NOTES:

1. The value of the USARTn_RTODR register is transferred to an internal counter when the RTOENn bit is set to '1' or when a start bit is received to.
2. If the RTOENn bit is set to '1', the internal counter counts down until underflow.
3. The RTOnIFLAG bit in the USARTn_CR3 register is set to '1' on the internal counter underflow and the receive time out function is finished. So, to re-use the receive time out function, the RTOENn bit must be cleared to '0' and then set to '1' again.

12.5.7.7 UART Auto Baud Rate Detection

The auto baud rate detection is enabled by setting ‘1’ to the ABDENn bit of the USARTn_CR3 register. The function is useful when using clock sources with relatively low accuracy. There are two auto baud rate detection modes, “Start bit to measure” and “0x55 character to measure” (n = 10, 11, and 12).

Figure 60. Auto Baud Rate Detection Timing Diagram (n = 10, 11, and 12)


1. The Normal Speed Operation (DBLSn = 0b) and Auto Baud Rate Detection Mode 0 (ABDnMD = 0).
USARTn clock
Rx Line
Start bit
1
ABDnIFLAG bit
USARTn_BDR[11:0]
x
(Clock counting value ÷ 16) − 1
If the clock counting value is not between 16 and 65536, the ABDERR bit will be set to '1'
2. The Double Speed Operation (DBLSn = 1b) and Auto Baud Rate Detection Mode 1 (ABDnMD = 1).
USARTn clock
Rx Line
Start
1
0
1
0
1
0
1
0
ABDnIFLAG bit
USARTn_BDR[11:0]
x
First updated
(Clock counting value ÷ 8) − 1
{(Clock counting value ÷ 8) ÷ 8} − 1
If the (Clock counting value ÷ 8) is not between 8 and 32,768,
the ABDERR bit will be set to '1'

12.5.8 SPI Mode

The USART can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full-Duplex, Three-wire synchronous data transfer.
- Mater and Slave Operation.
- Supports all four SPI modes of operation (mode 0, 1, 2, and 3).
- Selectable LSB first or MSB first data transfer.
- Double buffered transmit and receive.
- Programmable transmit bit rate.
- Up to 8 MHz data transfer for SPI

When the SPI mode is enabled by configuring the USTnMS[1:0] bits as ”11”, the slave select (SSn) pin becomes active LOW input in Slave mode operation if the USTnSSEN bit is set to ‘1’. The SSn function is not automatically controlled in master mode operation even if the USTnSSEN bit is set to ‘1’.

Note that during SPI mode of operation, the pin RXDn is renamed as MISOn and TXDn is renamed as MOSIn for compatibility to other SPI devices (n = 10, 11, and 12).

12.5.9 SPI Clock Formats and Timing

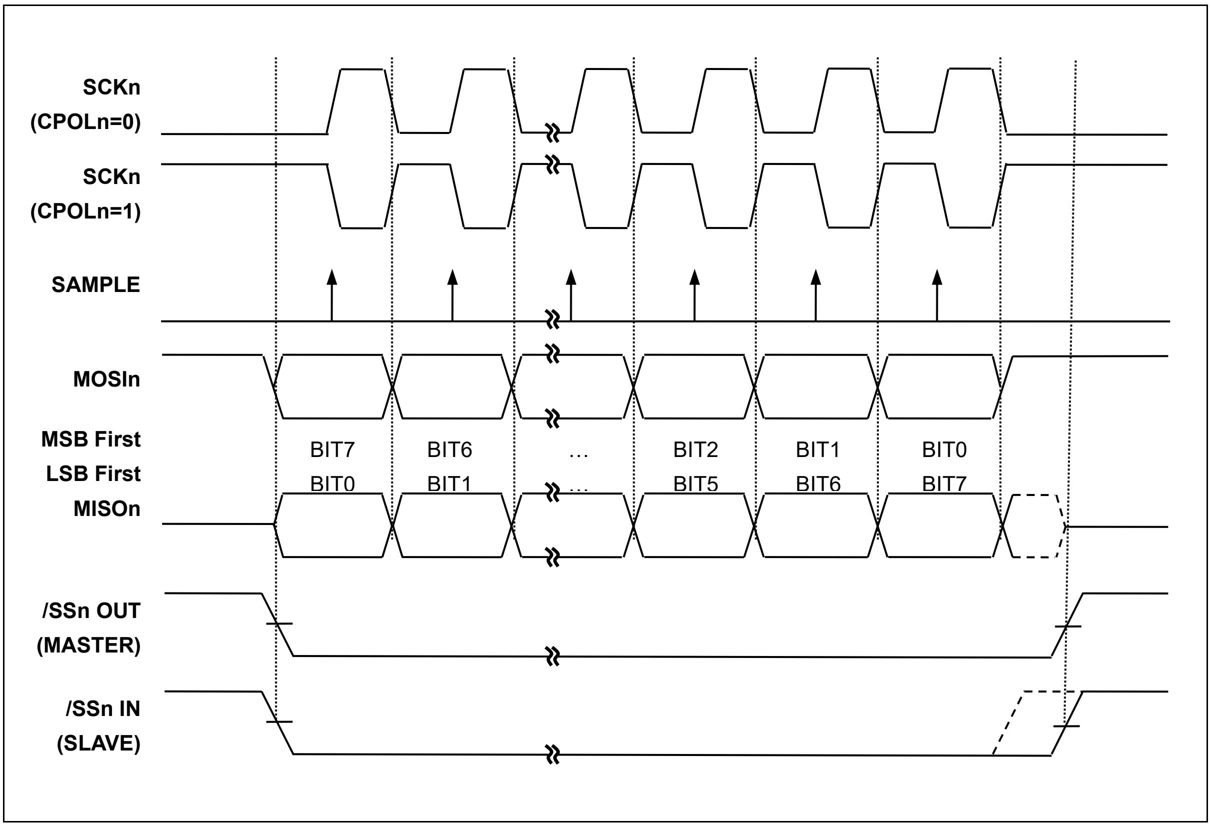
To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USART has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. The CPOLn selectively inserts an inverter in series with the clock. The CPHAn chooses between two different clock phase relationships between the clock and data. Note that the CPHAn and CPOLn bits in the USTnCR0 register have different meanings according to the USTnMS[1:0] bits, which decide the operating mode of USART.

Table 63 shows four combinations of the CPOLn and CPHAn for SPI mode 0, 1, 2, and 3 (n = 10, 11, and 12).

Table 63. CPOL Functionality (n = 10, 11, and 12)

SPIn Mode	CPOLn	CPHAn	Leading Edge	Trailing Edge
0	0	0	Sample (rising)	Setup (falling)
1	0	1	Setup (rising)	Sample (falling)
2	1	0	Sample (falling)	Setup (rising)
3	1	1	Setup (falling)	Sample (rising)

Figure 61. USART SPIn Clock Formats when CPHAn=0 (n = 10, 11, and 12)

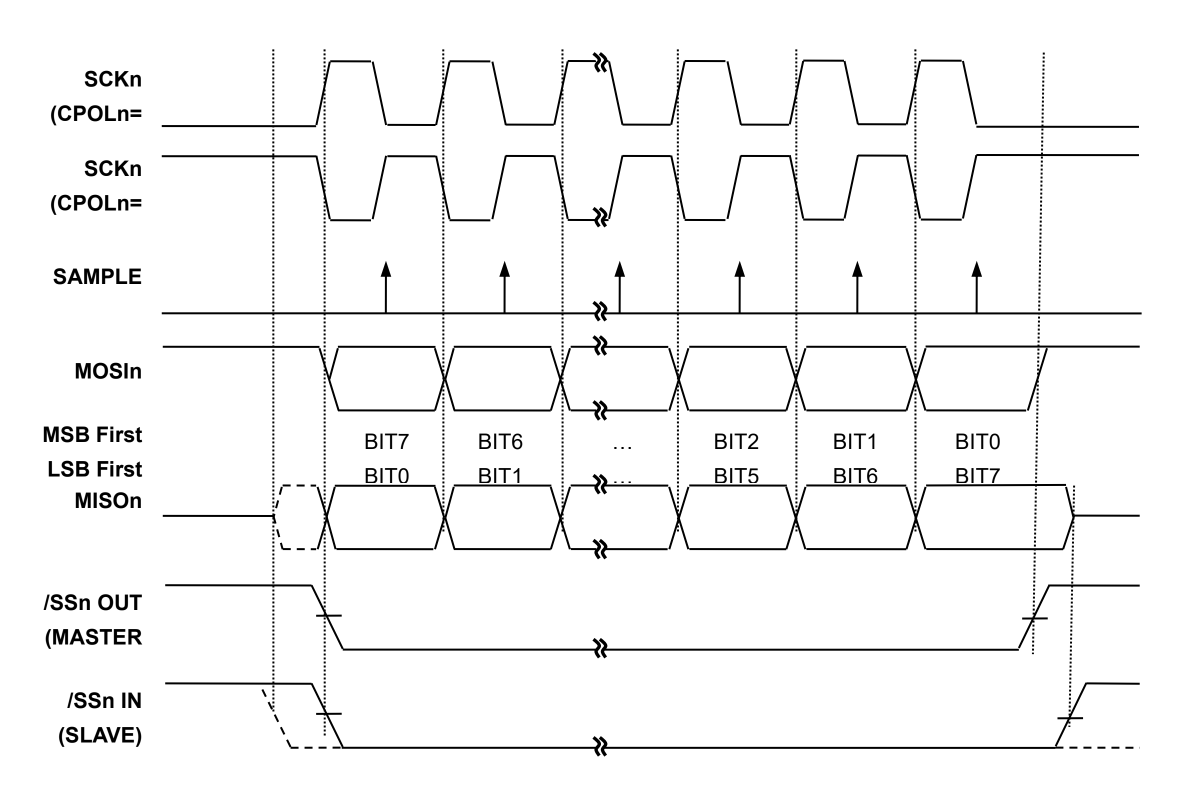


When CPHAn=0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low.

The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively.

At the second SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively (n = 10, 11, and 12).

Figure 62. USART SPIn Clock Formats when CPHAn=1 (n = 10, 11, and 12)



When CPHAn=1, the slave begins to drive its MISOn output when SSn goes active low, but the data is not defined until the first SCKn edge.

The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave.

The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively.

At the third SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively.

Because the SPIn logic reuses USART resources, SPIn mode of operation is similar to that of synchronous or asynchronous operation.

A SPIn transfer is initiated by checking for the USART Data Register Empty flag (DREn=1) and then writing a byte of data to the USARTn_TDR Register. In master mode of operation, even when transmission is not enabled (TXEn=0), writing data to the USARTn_TDR register is necessary because the clock SCKn is generated from the transmitter block.

12.5.10 Local Interconnection Network (LIN) Mode

The LIN mode is selected by writing “10” to the USTnMS[1:0] bits in the USARTn_CR1 register. The LIN transmission is fixed as start bit, 8-bits data length, 1-stop bit, and no parity. So, it should be set as follows.

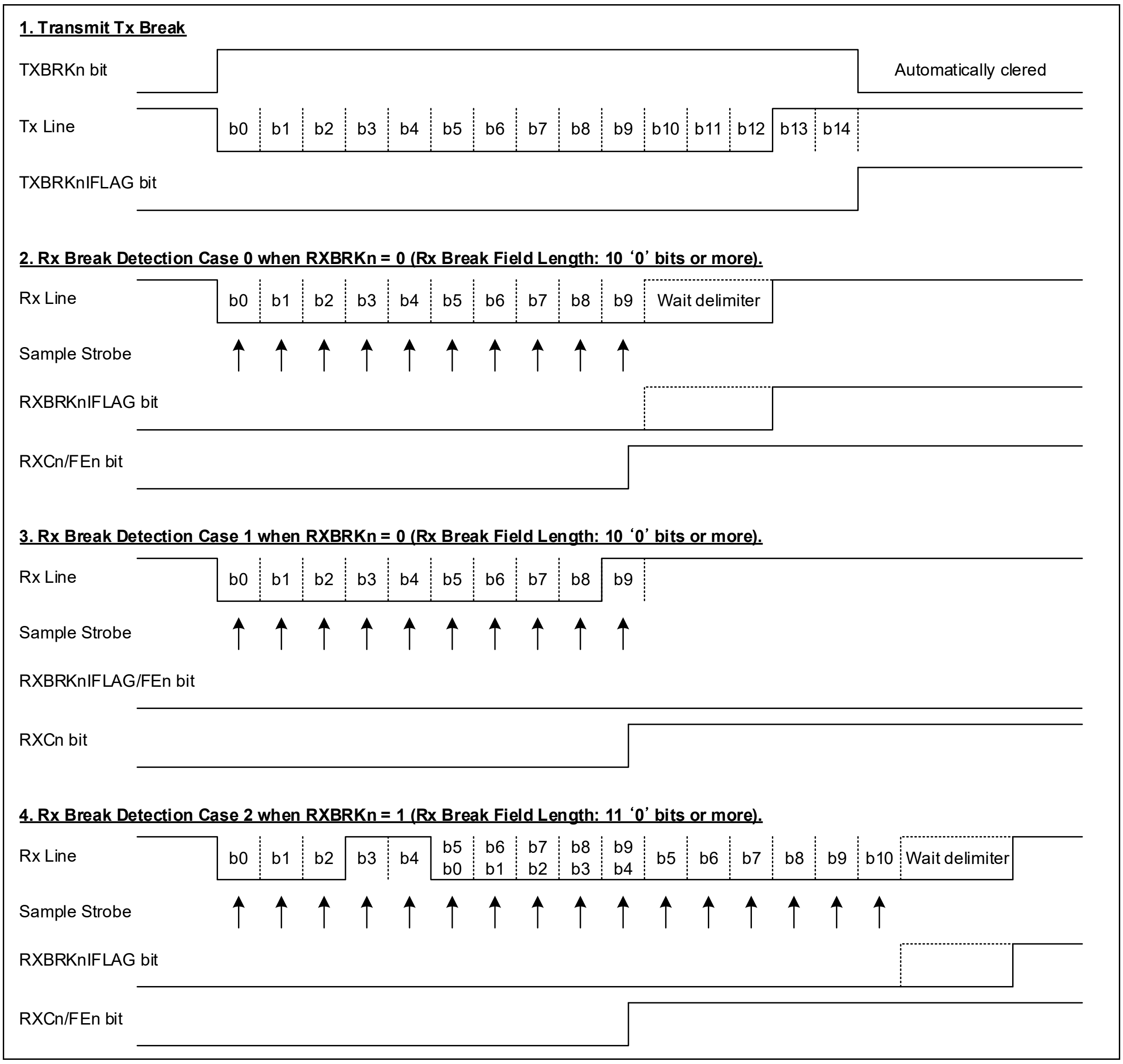
- USTnP[1:0] bits of USARTn_CR1: cleared to “00” for no parity.
- USTnS[2:0] bits of USARTn_CR1: set to “011” for 8-bit data length.
- USTnSB and LOOPSn bits USARTn_CR2: cleared to ‘0’ for one stop bit and normal operation.

During LIN mode is enabled, the break field detection circuit is activated, and it is independent from the UART receiver. A break field can be detected whenever it occurs during idle state or during a frame.

When the auto baud rate detection is enabled by setting ‘1’ to the ABDENn bit in the USARTn_CR3 register, the Rx break field may not be detected. So, the auto baud rate detection function should be enabled after the Rx break field detection if needed.

Figure 63 shows the timing diagram for LIN break field.

Figure 63. LIN Break Field Detection and Transmit Timing Diagram (n = 10, 11, and 12)


1. Transmit Tx Break
TXBRKn bit
Automatically clered
Tx Line
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14
TXBRKnIFLAG bit
2. Rx Break Detection Case 0 when RXBRKn = 0 (Rx Break Field Length: 10 '0' bits or more).
Rx Line
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 Wait delimiter
Sample Strobe
RXBRKnIFLAG bit
RXCn/FEn bit
3. Rx Break Detection Case 1 when RXBRKn = 0 (Rx Break Field Length: 10 '0' bits or more).
Rx Line
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9
Sample Strobe
RXBRKnIFLAG/FEn bit
RXCn bit
4. Rx Break Detection Case 2 when RXBRKn = 1 (Rx Break Field Length: 11 '0' bits or more).
Rx Line
b0 b1 b2 b3 b4 b5 b0 b6 b1 b7 b2 b8 b3 b9 b4 b5 b6 b7 b8 b9 b10 Wait delimiter
Sample Strobe
RXBRKnIFLAG bit
RXCn/FEn bit

12.6 USART Registers

The base address and register map of the USART 10/11/12 are described in the following tables:

Table 64. Base Address of USART 10/11/12

Name	Base Address
USART10	0x4000_3800
USART11	0x4000_3900
USART12	0x4000_3A00

Table 65. USARTn Register Map

Name	Offset	Type	Description	Reset Value	Reference
USARTn_CR1	0x0000	RW	USARTn Control Register 1	0x0000_0000	12.6.1
USARTn_CR2	0x0004	RW	USARTn Control Register 2	0x0000_0000	12.6.2
USARTn_CR3	0x0008	RW	USARTn Control Register 3	0x0000_0000	12.6.3
USARTn_ST	0x000C	RW	USARTn Status Register	0x0000_0080	12.6.4
USARTn_BDR	0x0010	RW	USARTn Baud-rate Generation Register	0x0000_0FFF	12.6.5
USARTn_RDR	0x0014	RO	USARTn Receive Data Register	0x0000_0000	12.6.6
USARTn_TDR	0x0018	RW	USARTn Transmit Data Register	0x0000_0000	12.6.7
USARTn_RTODR	0x000C	RW	USARTn Receive Time-out Data Register	0x0000_00FF	12.6.8
USARTn_RCDR	0x0020	RW	USARTn Receive Character Detection Data Register	0x0000_0000	12.6.9

NOTE:

1. n = 10, 11, and 12

12.6.1 USARTn_CR1: USARTn Control Register 1

USART module should be configured properly before running.

The USARTn_CR1 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_CR1=0x4000_3800, USART11_CR1=0x4000_3900, USART12_CR1=0x4000_3A00

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14	13 12	11 10 9	8	7	6	5	4	3	2	1	0
Reserved		USTnMS[1:0]	USTnP[1:0]	USTnS[2:0]	ORDn	CPOLn	CPHAn	DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn
-		00	00	000	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15 14	USTnMS[1:0]	USARTn Operation Mode Selection.			
		00	Asynchronous Mode. (UART)		
		01	Synchronous Mode.		
		10	Local Interconnection Network (LIN)		
		11	SPI mode		
		NOTE:	1. The LIN transmission is fixed as "Start, D0, D1, D2, D3, D4, D5, D6, D7, Stop1". So, the USTnP[1:0], LOOPSn, and USTnSB bits must be cleared to '0' and the USTnS[2:0] bits should be set to "011".		
13 12	USTnP[1:0]	Selects Parity Generation and Check method. (only UART mode)			
		00	No parity.		
		01	Reserved.		
		10	Even parity.		
		11	Odd parity.		
11 9	USTnS[2:0]	Selects the length of data bit in a frame at Asynchronous or Synchronous mode.			
		000	5-bit.		
		001	6-bit.		
		010	7-bit.		
		011	8-bit.		
		111	9-bit.		
		Others	Reserved.		
8	ORDn	Selects the first data bit to be transmitted. (only SPI mode)			
		0	LSB-first.		
		1	MSB-first.		
7	CPOLn	Selects the clock polarity of SCK in synchronous or SPI mode.			
		0	SCK to 'L' when idle.		
		1	SCK to 'H' when idle.		
6	CPHAn	CPOLn and this bit determine if data are sampled on the leading or the trailing edge of SCK. (only SPI mode)			
		CPOLn	CPHAn	Leading edge	Trailing edge
		0	0	Sample (rising)	Setup (falling)
		0	1	Setup (rising)	Sample (falling)
		1	0	Sample (falling)	Setup (rising)
		1	1	Setup (falling)	Sample (rising)
5	DRIEn	Transmit Data Register Empty Interrupt Enable.			
		0	Disable transmit data empty interrupt.		

		1	Enable transmit data empty interrupt.
4	TXCIEn	Transmit Complete Interrupt Enable.	
		0	Disable transmit complete interrupt.
		1	Enable transmit complete interrupt.
3	RXCIEn	Receive Complete Interrupt Enable.	
		0	Disable receive complete interrupt.
		1	Enable receive complete interrupt.
2	WAKEIEn	Asynchronous Wake-up Interrupt Enable in DEEP SLEEP Mode. When the device is in DEEP SLEEP mode, if RXDn goes to low level, an interrupt can be requested to wake-up system (only UART and LIN mode). This bit should be cleared to '0' to receive Rx data.	
		0	Disable asynchronous wake-up interrupt.
		1	Enable asynchronous wake-up interrupt. (Only used for wake-up)
1	TXEn	Enables the Transmitter unit.	
		0	Transmitter is disabled.
		1	Transmitter is enabled.
0	RXEn	Enables the Receiver unit.	
		0	Receiver is disabled.
		1	Receiver is enabled.

NOTE:

1. The CPOLn and CPHAn bits should be changed while TXEn and RXEn bits are '0'.

12.6.2 USARTn_CR2: USARTn Control Register 2

USART module should be configured properly before running.

The USARTn_CR2 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_CR2=0x4000_3804, USART11_CR2=0x4000_3904, USART12_CR2=0x4000_3A04

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10	9	8	7	6	5	4	3	2	1 0
Reserved	USTnEN	DBLSn	MASTERn	LOOPSn	DISSCKn	USTnSSEN	FXCHn	USTnSB	Reserved
-	0	0	0	0	0	0	0	0	-
-	RW	RW	RW	RW	RW	RW	RW	RW	-

9	USTnEN	Enable USARTn block. This bit can be cleared to '0' during the corresponding TXEn and RXEn bits are all '0'. **NOTE**: 1. This bit should be set to '1' after setting the related registers.	
		0	Disable USARTn block.
		1	Enable USARTn block.
8	DBLSn	Selects receiver sampling rate. (only asynchronous and LIN mode)	
		0	Normal asynchronous operation.
		1	Double speed asynchronous operation.
7	MASTERn	Selects master or slave in SPIn or Synchronous mode and controls the direction of SCKn pin.	
		0	Slave operation. (External clock for SCKn)
		1	Master operation. (Internal clock for SCKn)
6	LOOPSn	1. 1-wire Half-Duplex Communication on Asynchronous Mode.	
		0	Normal operation.
		1	1-wire half-duplex communication (The TXD and RXD lines are internally connected, the RXD pin is not used, and the TXD pin is always an input when no transmitted. So, the TXD pin must be configured to open-drain with an external pull-up resistor)
		2. Loop Back for Test on SPI and Synchronous Mode	
		0	Normal operation.
		1	Loop back (The "MOSI and MISO"/"TXD and RXD" lines are internally connected and the receive input is not used).
5	DISSCKn	In synchronous mode operation, selects the waveform of SCKn output.	
		0	SCKn continues to output the clock while USARTn is enabled in synchronous master mode.
		1	SCKn is active while any frame is transferring.
4	USTnSSEN	This bit controls the SSn pin operation. (only SPI mode)	
		0	Disable.
		1	Enable. (The SS pin should be configured as an alternative function)
3	FXCHn	SPIn port function exchange control. (only SPI mode)	
		0	No effect.
		1	Exchange MOSIn and MISOn function.
2	USTnSB	Selects the length of stop bit in Asynchronous or Synchronous mode.	
		0	1 stop bit.
		1	2 stop bits.

12.6.3 USARTn_CR3: USARTn Control Register 3

USART module should be configured properly before running. This register is used only for UART and LIN mode. The USARTn_CR3 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 10, 11, and 12)

USART10_CR3=0x4000_3808, USART11_CR3=0x4000_3908, USART12_CR3=0x4000_3A08

31 30 29 28 27 26 25 24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0
Reserved	RXBRKn	TXBRKn	RXBRKIEn	TXBRKIEn	RXBRKnIFLAG	TXBRKnIFLAG	Reserved	ABDENn	ABDnMD	ABDIEn	ABDnIFLAG	ABDnERR	Reserved	RCDENn	RTOENn	Reserved	RCDIENn	RTOIENn	Reserved	RCDnIFLAG	RTOnIFLAG	Reserved	BRDIVn
-	0	0	0	0	0	0	-	0	0	0	0	0	-	0	0	-	0	0	-	0	0	-	00
-	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW	RW	-	RW	RW	-	RW	RW	-	RW

23	RXBRKn	Rx Break Field Length (Only LIN mode)	
		0	Ten or more consecutive ‘0’ bits
		1	Eleven or more consecutive ‘0’ bits
22	TXBRKn	Tx Break Field (only LIN mode)	
		0	No effect
		1	Transmit Tx break (thirteen ‘0’ bits + two ‘1’ bits (“000000000000011”), automatically cleared)
21	RXBRKIEn	Rx Break Field Detection Interrupt Enable bit (only LIN mode)	
		0	Disable Rx break detection interrupt
		1	Enable Rx break detection interrupt
20	TXBRKIEn	Tx Break Field Completion Interrupt Enable bit (only LIN mode)	
		0	Disable Tx break completion interrupt
		1	Enable Tx break completion interrupt
19	RXBRKnIFLAG	Rx Break Field Detection Interrupt flag. This bit is set when LIN break is detected.	
		0	No request occurred
		1	Request occurred. This bit is cleared to ‘0’ when write ‘1’.
18	TXBRKnIFLAG	Tx Break Field Completion Interrupt flag. This bit is set when Tx break field is completely transmitted.	
		0	No request occurred
		1	Request occurred. This bit is cleared to ‘0’ when write ‘1’.
16	ABDENn	Auto Baud Rate Detection Enable bit.	
		0	Disable auto baud rate detection
		1	Enable auto baud rate detection (this bit is automatically cleared after operation)
		NOTE: 1. In the LIN mode, the Rx break field may not be detected while this bit is ‘1’. So, it is recommended to set this bit to ‘1’ for auto baud rate detection after the Rx break field detection.	
15	ABDnMD	Auto Baud Rate Detection Mode	
		0	Mode 0, The start bit is used to measure the baud rate (1ST bit must be ‘1’)
		1	Mode 1, The 0x55 character is used to measure the baud rate detection
14	ABDIEn	Auto Baud Rate Detection Interrupt Enable bit	
		0	Disable auto baud rate detection interrupt

		1	Enable auto baud rate detection interrupt
13	ABDnIFLAG		Auto Baud Rate Detection Interrupt Flag. This bit is set to ‘1’ when the auto baud rate detection finishes, whether an error occurs or not.
		0	No request occurred
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written
12	ABDnERR		Auto Baud Rate Detection Error bit. This bit is set to ‘1’ if the clock counting values are not between 16 and 65536 on the normal speed operation (DBLSn = 0) or between 8 and 32,768 on the double speed operation (DBLSn = 1).
		0	No error occurs
		1	An error occurs. This bit is cleared to ‘0’ when write ‘1’.
			NOTE: 1. If an error occurs, the USARTn_BDR register will not be updated.
10	RCDENn		Receive Character Detection Function Enable bit. This function is to compare the value of USARTn_RCDR register with the value just received.
		0	Disable receive detection function.
		1	Enable receive detection function.
9	RTOENn		Receive Time Out Function Enable bit. This function is to count time with baud rate units from the leading edge of a start bit to a new start bit. The receive time out controller counts down from the value of USARTn_RTODR register every start bit and set this bit. The RTOnIFLAG bit is set to ‘1’ at the counter underflow (only asynchronous mode).
		0	Disable receive time out function.
		1	Enable receive time out function.
7	RCDIEn		Receive Character Detection Interrupt Enable.
		0	Disable receive character detection interrupt
		1	Enable receive character detection interrupt
6	RTOIEn		Receive Time Out Interrupt Enable.
		0	Disable receive time out interrupt.
		1	Enable receive time out interrupt
4	RCDnIFLAG		Receive Character detection Interrupt Flag. This bit is set to ‘1’ if the value in the USARTn_RCDR register matches the value received in the non-error state of frame and parity. On match of them, the bit may be set even if data overrun occurs.
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written.
3	RTOnIFLAG		Receive Time Out Interrupt Flag
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written.
1 0	BRDIVn		Baud Rate Clock Dividing Selection for Receive Time Out. (only asynchronous and LIN mode).
		00	f_{BR}/1.
		01	f_{BR}/8
		10	f_{BR}/64
		11	f_{BR}/512

12.6.4 USARTn_ST: USARTn Status Register

The USARTn_ST register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_ST=0x4000_380C, USART11_ST=0x4000_390C, USART12_ST=0x4000_3A0C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			DREn	TXCn	RXCn	WAKEn	Reserved	DORn	FEn	PEn
-			1	0	0	0	-	0	0	0
-			RO	RW	RO	RW	-	RW	RW	RW

7	DREn	Transmit Data Register Empty Interrupt Flag. The flag is set to ‘1’ when the data in the USARTn_TDR register has been transferred to the transmit shift register. This bit is cleared by a write to the USARTn_TDR register (only UART mode).	
		0	Not transferred to the transmit shift register.
		1	Transferred to the transmit shift register.
6	TXCn	Transmit Complete Interrupt Flag. This flag is set to ‘1’ when the data in the transmit shift register has been shifted out and when the DREn = 1.	
		0	No request occurred.
		1	The data in the transmit shift register is shifted out completely. This bit is cleared to ‘0’ when write ‘1’.
5	RXCn	Receive Data Register Not Empty Interrupt Flag. This bit is set to ‘1’ when the data in the receive shift register has been transferred to the USARTn_RDR register. The bit is cleared by a read to the USARTn_RDR register.	
		0	No request occurred.
		1	There is data in the receive data register. This bit is cleared to ‘0’ when write ‘1’.
4	WAKEn	Asynchronous Wake-up Interrupt Flag. This flag is set when the RXD pin is detected low while the CPU is in DEEP SLEEP mode (only UART mode)	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when write ‘1’.
2	DORn	Data Overrun bit. This bit is set when the receive shift register is transferred to the USARTn_RDR register while the RXCn=1. The data of the shift register is ignored. This bit must be cleared by software to receive new data (only UART mode).	
		0	No Data OverRun.
		1	Data overrun detected. This bit is cleared to ‘0’ when write ‘1’.
1	FEn	Frame Error bit. This bit is set when the received data have not a valid stop bit. That is, the stop bit following the last data bit is detected as ‘0’. The bit will be cleared by H/W if new data are received (only UART mode).	
		0	No Frame Error.
		1	Frame error detected. This bit is cleared to ‘0’ when write ‘1’.
0	PEn	Parity Error bit. This bit is set when the received data has a parity error on parity enable. The bit will be cleared by H/W if new data are received (only UART mode).	
		0	No Parity Error.
		1	Parity error detected. This bit is cleared to ‘0’ when write ‘1’.

12.6.5 USARTn_BDR: USARTn Baud Rate Generation Register

The USARTn_BDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_BDR=0x4000_3810, USART11_BDR=0x4000_3910, USART12_BDR=0x4000_3A10

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="20">Reserved</td><td colspan="12">BDATA[11:0]</td></tr>
<tr><td colspan="20">-</td><td colspan="12">0xFFF</td></tr>
<tr><td colspan="20">-</td><td colspan="12">RW</td></tr>
</table>

11 0	BDATA[11:0]	The value in this register is used to generate internal baud rate in UART mode or to generate SCK clock in SPI mode. The range is 0x000 to 0xFFF in asynchronous UART and SPI mode, but the range is 0x002 to 0xFFF in synchronous mode.

12.6.6 USARTn_RDR: USARTn Receive Data Register

The USARTn_RDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_RDR=0x4000_3814, USART11_RDR=0x4000_3914, USART12_RDR=0x4000_3A14

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">Reserved</td><td colspan="9">RDATA[8:0]</td></tr>
<tr><td colspan="23">-</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="23">-</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td></tr>
</table>

8 0	RDATA[8:0]	Receive Data bits. A receive shift register is moved to this register after stop bit.

NOTE:

1. When asynchronous or synchronous mode, the RDATA[8] bit is the received 9th bit.

12.6.7 USARTn_TDR: USARTn Transmit Data Register

The USARTn_TDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_TDR=0x4000_3818, USART11_TDR=0x4000_3918, USART12_TDR=0x4000_3A18

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9	8	7	6	5	4	3	2	1	0
Reserved			TDATA[8:0]								
-			0	0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW	RW

8 0	TDATA[8:0]	Transmit Data bits. This register is moved to the transmit shift register after a previous character is completely shifted out. In SPI master mode, the SCK clock is generated when data is moved to the shift register. Do not write to this transmit data register while transmitting in SPI mode.

NOTES:

1. When in asynchronous or synchronous mode, the TDATA[8] bit is the 9th bit to be transmitted.
2. The data to be transmitted should be written after all control registers are set.

12.6.8 USARTn_RTODR: USARTn Receive Timeout Data Register

The USARTn_RTODR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_RTODR=0x4000_381C, USART11_RTODR=0x4000_391C, USART12_RTODR=0x4000_3A1C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RTOD[7:0]
-			0xFF
-			RW

7 0	RTOD[7:0]	USARTn Receive Timeout Data. Counting number: RTOD[7:0] +1

12.6.9 USARTn_RCDR: USARTn Receive Character Detection Data Register

The USARTn_CR3.RCDENn bit sets the comparison value when using the receive character auto-detect function. When this function is enabled, the result is reflected in the USARTn_CR3.RCDnIFLAG bit.

The USARTn_RCDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 10, 11, and 12).

USART10_RCDR=0x4000_3820, USART11_RCDR=0x4000_3920, USART12_RCDR=0x4000_3A20

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">Reserved</td><td colspan="8">RCDD[7:0]</td></tr>
<tr><td colspan="24">-</td><td colspan="8">0x00</td></tr>
<tr><td colspan="24">-</td><td colspan="8">RW</td></tr>
</table>

7 0	RCDD[7:0]	USARTn Receive Character Detection Data. When the USARTn_CR3.RCDENn bit is enabled, the result is reflected in the USARTn_CR3.RCDnIFLAG bit when the same character as the set value is received.

12.6.10 USART Register Map Summary

Table 66. USART Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USARTn_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USTnMS[1:0]		USTnP[1:0]		USTnS[2:0]			ORDn	CPOLn	CPHAn	DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USARTn_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USTnEN	DBLSn	MASTERn	LOOPSn	DISSCKn	USTnSSEN	FXCHn	USTnSB	Res.	Res.
	Reset value																							0	0	0	0	0	0	0	0		
0x08	USARTn_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXBRKn	TXBRKn	RXBRKIEn	TXBRKIEn	RXBRKnIFLA(	TXBRKnIFLA(	Res.	ABDENn	ABDnMD	ABDIEn	ABDnIFLAG	ABDnERR	Res.	RCDENn	RTOENn	Res.	RCDIEn	RTOIEn	Res.	RCDnIFLAG	RTOnIFLAG	Res.	BRDIVn[1:0]	
	Reset value									0	0	0	0	0	0		0	0	0	0	0		0	0		0	0		0	0		0	0
0x0C	USARTn_ST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DREn	TXCn	RXCn	WAKEn	Res.	DORn	FEn	PEn
	Reset value																									1	0	0	0		0	0	0
0x10	USARTn_BDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDATA[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	USARTn_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDATA[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	USARTn_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDATA[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x1C	USARTn_RTODR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTOD[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x20	USARTn_RCDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCDD[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

13. Universal Asynchronous Receiver/Transmitter (UART)

13.1 Introduction

The A31S134 has a two-channel UART module. This built-in UART module transmits and receives data according to user-specified settings and reads the current UART status.

The UART status information includes type and conditions of the current UART transmission / reception process and can be used to check for errors (Parity, overrun, framing, or break interrupts) that may occur during data reception.

Each UART channel has a programmable baud-rate generator that generates an internal clock for the corresponding UART unit by dividing the prescaled clock using a baud-rate divisor (ranging from 4 to 65,535) and then dividing the result by 16.

Users can program interrupts that can control the UART communication.

Using Direct Memory Access (DMA), users can perform high-speed data communication.

13.2 Main Features

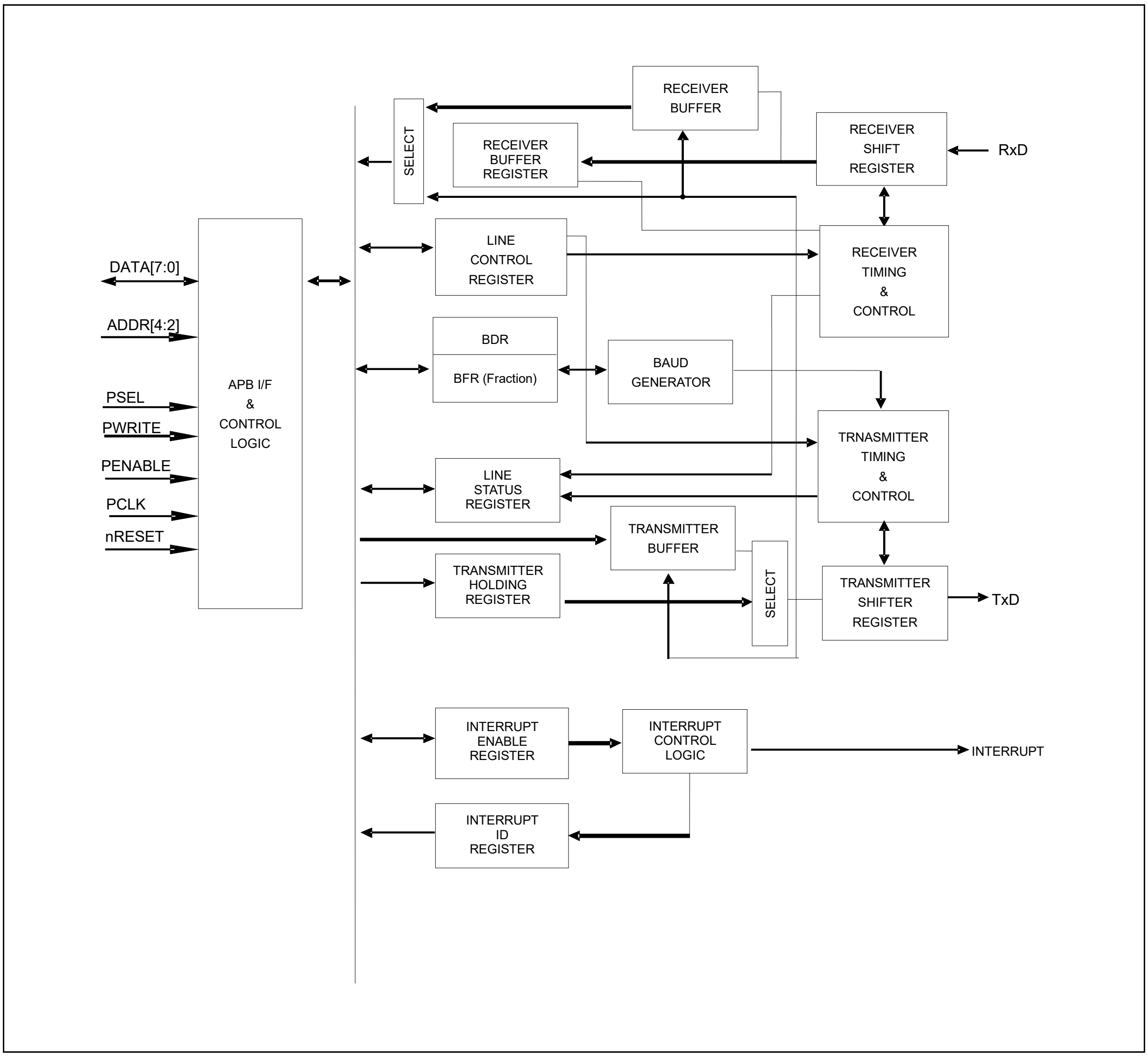
The UART 0/1 of the A31S134 has the following features:

- Compatible with 16450 UART
- Configurable standard asynchronous control bit (Start, Stop, and Parity)
- Programmable 16-bit fractional baud generator
- Programmable serial communication
- 5-, 6-, 7- or 8- bit data transfer
- Even, odd, or no-parity bit insertion and detection
- 1-, 1.5- or 2-Stop bit-insertion and detection
- 16-bit baud rate generation with 8-bit fraction control
- Hardware inter-frame delay function
- Stop bit error detection
- Line status register

13.3 UART 0/1 Block Diagram

Figure 64 shows a block diagram of the UART block.

Figure 64. UARTn Block Diagram (n = 0 and 1)



13.4 Pin Description for UART 0/1

Table 67. Pins and External Signals for UARTn (n = 0 and 1)

Pin Name	Type	Description
TXDn	O	UART Channel n transmit output
RXDn	I	UART Channel n receive input

13.5 UART Functional Description

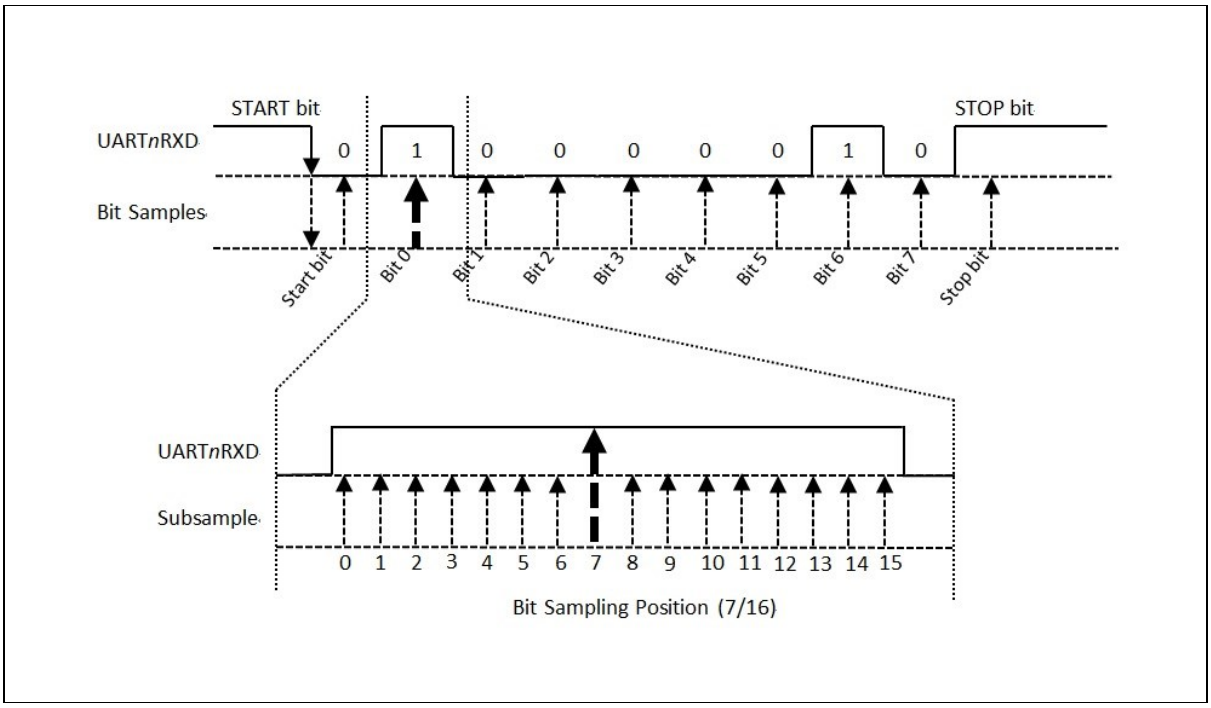
The UART module is compatible with 16450 UART. Additionally, fractional baud rate compensation logic is provided. It does not have an internal FIFO block.

13.5.1 Receiver Sampling Timing

The UART of the A31S134 operates at the following timing as shown in Figure 65.

If a falling-edge is detected on the receive line, the UART considers it as a start bit. From then on, the UART oversamples 1-bit 16 times and detects the bit value at the 7th sample.

Figure 65. Sampling Timing of UART Receiver



It is recommended to enable debounce settings in the PCU block to enhance immunity to external glitch noise.

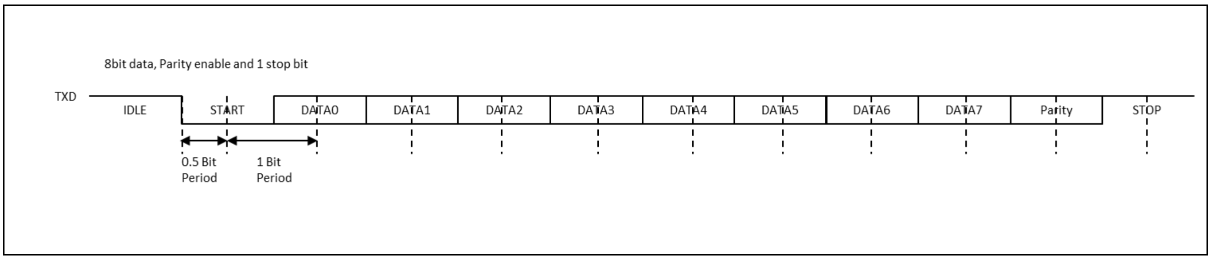
13.5.2 Transmitter

The transmitter has a data transmission function. The start bit, data bits, optional parity bit, and stop bit shift in series, the least significant bit shifting first.

The number of data bit is selected in DLEN[1:0] in the UARTn_LCR register. The parity bit is set according to the PARITY and PEN bits in the UARTn_LCR register. If the parity type is even, the parity bit depends on one-bit sum of all data bits. For odd parity, the parity bit is an inverted sum of all data bits. The number of stop bits is selected in the STOPBIT in the UARTn_LCR register.

The example of transmission data format is described in Figure 66.

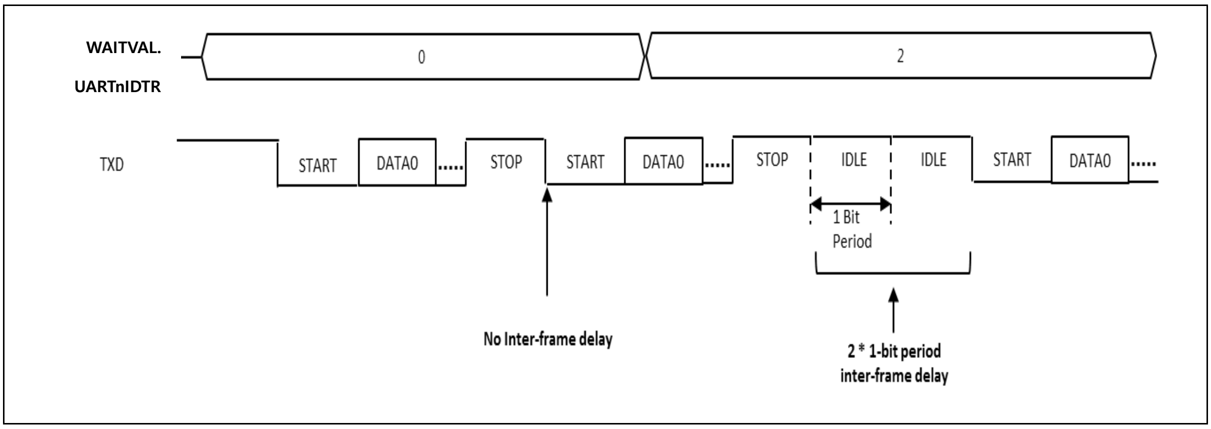
Figure 66. Transmission Data Format Example



13.5.3 Inter-Frame Delay Transmission

The inter-frame delay function allows the transmitter to insert an idle state on the TXD line between 2 characters. The width of the idle state is defined in WAITVAL field of the UARTn_IDTR register. When this field is set as 0, no time-delay is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted character during the number of bit periods defined in WAITVAL field.

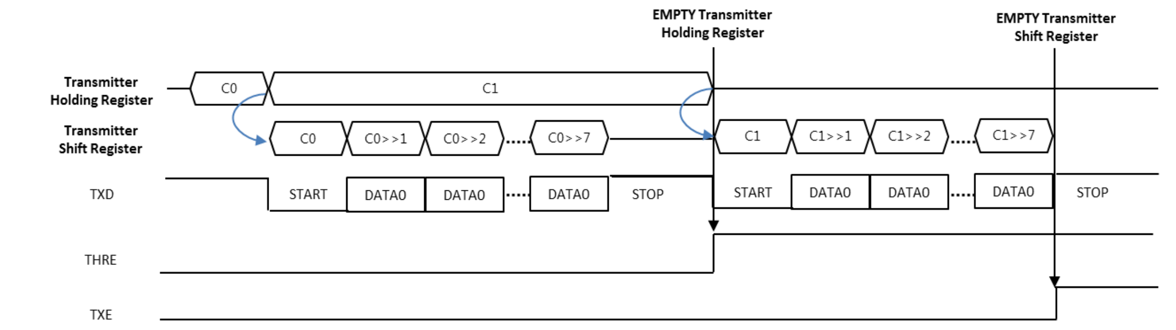
Figure 67. Inter-frame Delay Timing Diagram



13.5.4 Transmit Interrupt

The transmission operation makes some kinds of interrupt flags. When transmitter hold register is empty, the THRE interrupt flag will be raised. When transmitter shifter register is empty, the TXE interrupt flag will be raised. User can select an interrupt timing that works the best for the application.

Figure 68. Transmit Interrupt Timing Diagram



13.6 UART Registers

The base address and register map of the UART 0/1 are described in the following tables:

Table 68. Base Address of UART 0/1

Name	Base Address
UART0	0x4000_4000
UART1	0x4000_4100

Table 69. UARTn Register Map

Name	Offset	Type	Description	Reset Value	Reference
UARTn_RBR	0x00	RO	UARTn Receive Data Buffer Register	0x00000000	13.6.1
UARTn_THR	0x00	WO	UARTn Transmit Data Hold Register	0x00000000	13.6.2
UARTn_IER	0x04	RW	UARTn Interrupt Enable Register	0x00000000	13.6.3
UARTn_IIR	0x08	RO	UARTn Interrupt ID Register	0x00000001	13.6.4
UARTn_LCR	0x0C	RW	UARTn Line Control Register	0x00000000	13.6.5
UARTn_DCR	0x10	RW	UARTn Data Control Register	0x00000000	13.6.6
UARTn_LSR	0x14	RO	UARTn Line Status Register	0x00000060	13.6.7
UARTn_BDR	0x20	RW	UARTn Baud Rate Divisor Latch Register	0x00000000	13.6.8
UARTn_BFR	0x24	RW	UARTn Baud Rate Fractional Counter Value	0x00000000	13.6.9
UARTn_IDTR	0x30	RW	UARTn Inter-frame Delay Time Register	0x000000C0	13.6.10

NOTE:

1. n = 0 and 1

13.6.1 UARTn_RBR: UARTn Receive Data Buffer Register

Received data will be read from UARTn_RBR register. The maximum length of data is 8 bits. The last data received will stay in this register until a new byte is received. The UARTn_RBR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_RBR=0x4000_4000, UART1_RBR=0x4000_4100

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RBR[7:0]
-			0x00
-			RO

7 0	RBR[7:0]	UARTn Receive Data Buffer.

13.6.2 UARTn_THR: UARTn Transmit Data Hold Register

The UARTn_THR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_THR=0x4000_4000, UART1_THR=0x4000_4100

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			THR[7:0]
-			0x00
-			WO

7 0	THR[7:0]	UARTn Transmit Data Hold.

13.6.3 UARTn_IER: UARTn Interrupt Enable Register

The UARTn_IER register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_IER=0x4000_4004, UART1_IER=0x4000_4104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												TXEIE	RLSIE	THREIE	DRIE
-																												0	0	0	0
-																												RW	RW	RW	RW

3	TXEIE	Transmit Register Empty Interrupt Enable.	
		0	Disable transmit register empty interrupt.
		1	Enable transmit register empty interrupt.
2	RLSIE	Receiver Line Status Interrupt Enable.	
		0	Disable receiver line status interrupt.
		1	Enable receiver line status interrupt.
1	THREIE	Transmit Holding Register Empty Interrupt Enable.	
		0	Disable transmit hold register empty interrupt.
		1	Enable transmit hold register empty interrupt.
0	DRIE	Data Receive Interrupt Enable.	
		0	Disable data receive interrupt.
		1	Enable data receive interrupt.

13.6.4 UARTn_IIR: UARTn Interrupt ID Register

The UARTn_IIR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_IIR=0x4000_4008, UART1_IIR=0x4000_4108

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5	4	3	2 1	0
Reserved	TXE	Reserved	IID[1:0]	DRIE
-	0	-	00	1
-	RO	-	RO	RO

4	TXE	Transmit Complete Interrupt Source ID.
2 1	IID[1:0]	UARTn Interrupt ID.

NOTE:

1. The UARTn supports 3-priority interrupt generation and the interrupt source ID register shows one interrupt source which has highest priority among pending interrupts.
   The priority is defined as below.
   - Receive line status interrupt.
   - Receive data ready interrupt and Character timeout interrupt.
   - Transmit hold register empty interrupt.

0	IPEN	Interrupt Pending.	
		0	Interrupt is pending.
		1	No interrupt is pending.

Table 70. Interrupt ID and Control of UARTn_IIR

Priority	TXE	IID		IPEN	Interrupt Sources		
	bit-4	bit-2	bit-1	bit-0	Interrupt	Interrupt Condition	Interrupt Clear
-	0	0	0	1	None	-	-
1	0	1	1	0	Receiver Line Status	Overrun, Parity, Framing or Break Error	Read LSR register
2	0	1	0	0	Receiver Data Available	Receive data is available.	Read receive register or read IIR register
3	0	0	1	0	Transmitter Holding Register Empty	Transmit buffer empty	Write transmit hold register or read IIR register
4	1	X	X	X	Transmitter Register Empty	Transmit register empty	Write transmit hold register or read IIR register

NOTE:

1. After checking the above bits, Read data buffer to avoid losing interrupt source.

13.6.5 UARTn_LCR: UARTn Line Control Register

The UARTn_LCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_LCR=0x4000_400C, UART1_LCR=0x4000_410C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved				BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN[1:0]	
-				0	0	0	0	0	00	
-				RW	RW	RW	RW	RW	RW	

6	BREAK	Transfer Break Control. The TXDn pin will be driven at low state to notice the alert to the receiver.	
		0	Normal transfer mode.
		1	Break transmit mode.
5	STICKP	Force Parity. This bit is effective when the PEN bit is set to '1'.	
		0	Disable parity stuck.
		1	Enable parity stuck. A parity is always the value of the PARITY bit.
4	PARITY	Parity Mode and Parity Stuck Selection.	
		0	Odd parity mode.
		1	Even parity mode.
3	PEN	Parity Bit Transfer Enable.	
		0	Disable parity transfer.
		1	Enable parity transfer.
2	STOPBIT	Stop Bit Length Selection.	
		0	1 stop bit.
		1	1.5 or 2 stop bit. 1.5 stop bit in case of 5-bit data and 2 stop bit in case of 6-/7-/8-bit data.
1 0	DLEN[1:0]	Data Length Selection.	
		00	5-bit data length
		01	6-bit data length
		10	7-bit data length
		11	8-bit data length

Parity bit will be generated according to bit-3, -4, -5 of UARTn_LCR register.

Table 71 shows the variation of parity bit generation.

Table 71. Parity Bit Setting

STICKP	PARITY	PEN	Parity
X	X	0	No Parity
0	0	1	Odd Parity
0	1	1	Even Parity
1	0	1	Force parity as '1'
1	1	1	Force parity as '0'

13.6.6 UARTn_DCR: UARTn Data Control Register

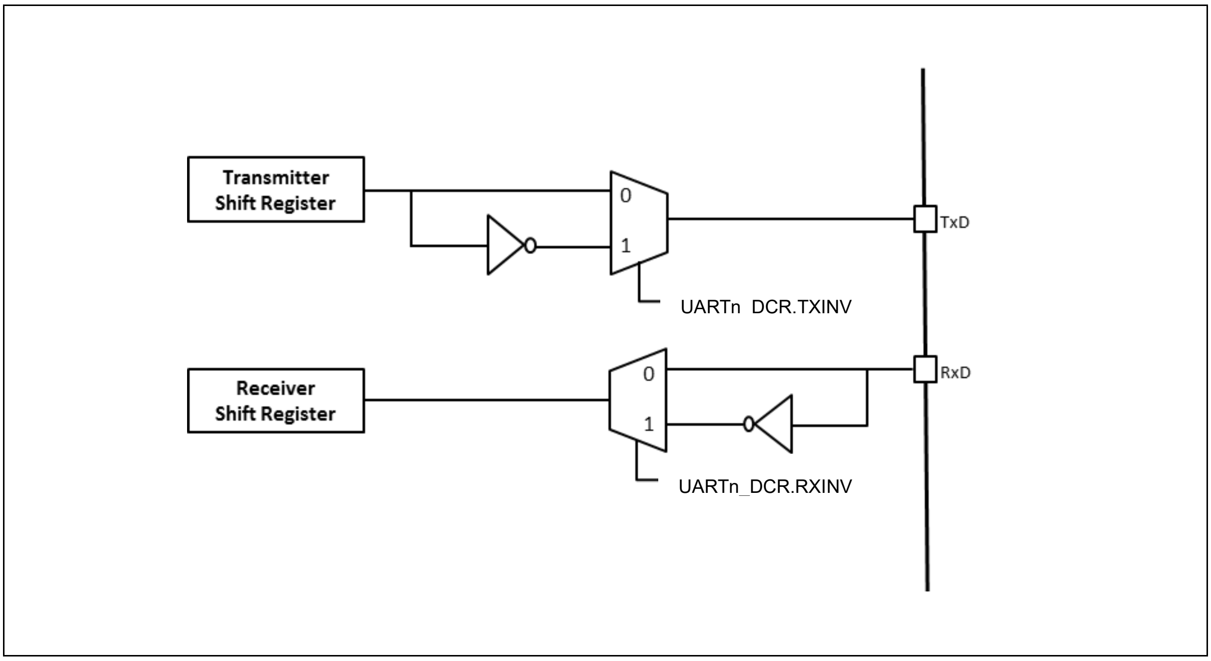
The UARTn_DCR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_DCR=0x4000_4010, UART1_DCR=0x4000_4110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											LBON	RXINV	TXINV	Reserved	
-																											0	0	0	-	
-																											RW	RW	RW	-	

4	LBON	Local Loopback Test Mode Enable.
		0 Normal mode.
		1 Local loopback mode. TXDn connected to RXDn internally.
3	RXINV	Receive Data Inversion Selection.
		0 Normal receive data input.
		1 Inverted receive data input.
2	TXINV	Transmit Data Inversion Selection.
		0 Normal transmit output.
		1 Inverted transmit output.

Figure 69. Data Inversion Control Diagram



13.6.7 UARTn_LSR: UARTn Line Status Register

The UARTn_LSR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_LSR=0x4000_4014, UART1_LSR=0x4000_4114

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved				TEMT	THRE	BI	FE	PE	OE	DR
-				1	1	0	0	0	0	0
-				RO	RO	RO	RO	RO	RO	RO

6	TEMT	Transmit Empty.	
		0	Transmit register has data or is transmitting.
		1	Transmit register is empty.
5	THRE	Transmit Holding Empty.	
		0	Transmit hold register is not empty.
		1	Transmit hold register is empty
		NOTE: 1. This bit will be set to ‘1’ when it starts transmission.	
4	BI	Break Condition Indication.	
		0	Normal status.
		1	Break condition is detected.
3	FE	Frame Error Indicator.	
		0	No frame error.
		1	Frame error takes place. The receive character did not have a valid stop.
2	PE	Parity Error Indicator.	
		0	No parity error.
		1	Parity error takes place. The receive character does not have correct parity information.
1	OE	Overrun Error Indicator.	
		0	No overrun error.
		1	Overrun error takes place. Additional data arrived while RHR is full.
0	DR	Data Receive Indicator.	
		0	No data in receive hold register.
		1	Data has been received and is saved in the receive hold register.

This register provides the status of data transfers between transmitter and receiver. Users can check the line status from this register. Bit-1, -2, -3, -4 will raise the line status interrupt when the RLSIE bit in the UARTn_IER register is set. Other bits can generate interrupts when their interrupt enable bits in the UARTn_IER register are set.

13.6.8 UARTn_BDR: UARTn Baud Rate Divisor Latch Register

The UARTn_BDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_BDR=0x4000_4020, UART1_BDR=0x4000_4120

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved	BDR[15:0]
-	0x0000
-	RW

15 0	BDR[15:0]	Baud Rate Divider Latch Value Baud rate = PCLK/(16 x (BDR[15:0] + 1)). The range is 0x0000 to 0xFFFF.

To establish communication with the UART channel, baud rate should be set properly. The programmable baud rate generator provides divider number from 0 to 65,535. The expected baud-rate should be written to the 16-bit divider register (UARTn_BDR). UART$_{clock}$ is PCLK.

Baud rate calculation formula is as follows:

$$\mathrm{BDR} = \frac{UART_{clock}}{16 \times BaudRate} - 1$$

In the case of 32 MHz UART$_{clock}$ speed, the divider value and error rate are shown in table.

Table 72. Example of Baud Rate Calculation (without BFR)

UART$_{clock}$= 32 MHz		
Baud-rate	**Divider**	**Error Rate (%)**
1,200	1,665	0.04%
2,400	832	0.04%
4,800	415	0.16%
9,600	207	0.16%
19,200	103	0.16%
38,400	51	0.16%
57,600	33	2.12%
115,200	16	2.12%

13.6.9 UARTn_BFR: UARTn Baud Rate Fraction Counter Register

The UARTn_BFR register is a 32-bit size register for the baud rate decimal adjustment function. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_BFR=0x4000_4024, UART1_BFR=0x4000_4124

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	BFR[7:0]
-	0x00
-	RW

7
0 BFR[7:0] Fraction Counter value.

0 Disable fraction counter.

N Fraction compensation mode under operation. Fraction counter is incremented by FCNT.
FCNT = Float × 256

NOTE:

1. 8-bit fractional counter will count up by FCNT value every (baud rate)/16 period and whenever fractional counter overflow happened, the divisor value will increment by one. So, this period will be compensated. Then next period, the divisor value will return to original set value.

Table 73. Example of Baud Rate Calculation

$\text{UART}_{\text{clock}}$ = 32 MHz			
Baud-rate	**Divider**	**FCNT**	**Error Rate(%)**
1,200	1,665	170	0.00%
2,400	832	85	0.00%
4,800	415	170	0.00%
9,600	207	85	0.00%
19,200	103	42	0.00%
38,400	51	21	0.00%
57,600	33	184	0.01%
115,200	16	92	0.01%

The FCNT value can be calculated using the equation below:

$$\text{FCNT} = \text{Float} \times 256$$

For example, when the target baud rate is 4,800 bps and $\text{UART}_{\text{clock}}$ is 32 MHz, the BDR value is 415.6666. The integer 415 is the BDR value and floating number 0.6666 leads to the FNCT value as follows:

FCNT = 0.6666 × 256 = 170.6496, and thus the FCNT value is 170.

8-bit fractional counter will count up by FCNT value every (baud-rate)/16 periods and whenever fractional counter overflow takes place; the divisor value will increment by one and compensate this period. Then, the divisor value will return to its original value.

13.6.10 UARTn_IDTR: UARTn Inter-Frame Delay Time Register

The UARTn_IDTR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

UART0_IDTR =0x4000_4030, UART1_IDTR=0x4000_4130

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7	6	5 4 3	2 1 0
Reserved	SMS	DMS	Reserved	WAITVAL[2:0]
-	1	1	000	000
-	RW	RW	-	RW

7	SMS	Start Bit Multi Sampling Enable.
		0 Multi sampling is disabled for start bit, Single sampling will be done at 8/16 baud rate for the start bit.
		1 Multi sampling is enabled for start bit. Sampling is done three times at 7/16, 8/16, and 9/16 baud rate. Dominant value among three samples will be selected for the start bit.
6	DMS	Data Bit Multi sampling enable.
		0 Multi sampling is disabled for data bit, Single sampling will be done at 8/16 baud rate for the data bit.
		1 Multi sampling is enabled for data bit. Sampling is done three times at 7/16, 8/16, and 9/16 baud-rate. Dominant value among three samples will be selected for the data bit.
2 0	WAITVAL[2:0]	Wait Time Value. Dummy delay can be inserted between two continuous transmits.
		Wait Time = WAITVAL[2:0]/(Baud-rate)
		See 13.5.3 Inter-Frame Delay Transmission for a detailed description of the operation.

13.6.11 UART Register Map Summary

Table 74. UARTn Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	UARTn_RBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x00	UARTn_THR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	THR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x04	UARTn_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXEIE	RLSIE	THREIE	DRIE
	Reset value																													0	0	0	0
0x08	UARTn_IIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXE	Res.	IID[1:0]		IPEN
	Reset value																												0		0	0	1
0x0C	UARTn_LCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN[1:0]	
	Reset value																										0	0	0	0	0	0	0
0x10	UARTn_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LBON	RXINV	TXINV	Res.	Res.
	Reset value																												0	0	0		
0x14	UARTn_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMT	THRE	BI	FE	PE	OE	DR
	Reset value																										1	1	0	0	0	0	0
0x20	UARTn_BDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	UARTn_BFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BFR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x30	UARTn_IDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS	DMS	Res.	Res.	Res.	WAITVAL[2:0]		
	Reset value																									0	0				0	0	0

NOTE:

1. n = 0 and 1.

14. Serial Peripheral Interface (SPI)

14.1 Introduction

SPI interface enables synchronous serial data transfer between external serial devices. It allows full-duplex communication using four-wires (MOSIn, MISOn, SCKn, SSn).

It supports master and slave modes and selects serial clock (SCKn) polarity. In addition, for the data transmission, it selects whether to transfer LSB first or MSB first.

14.2 Main Features

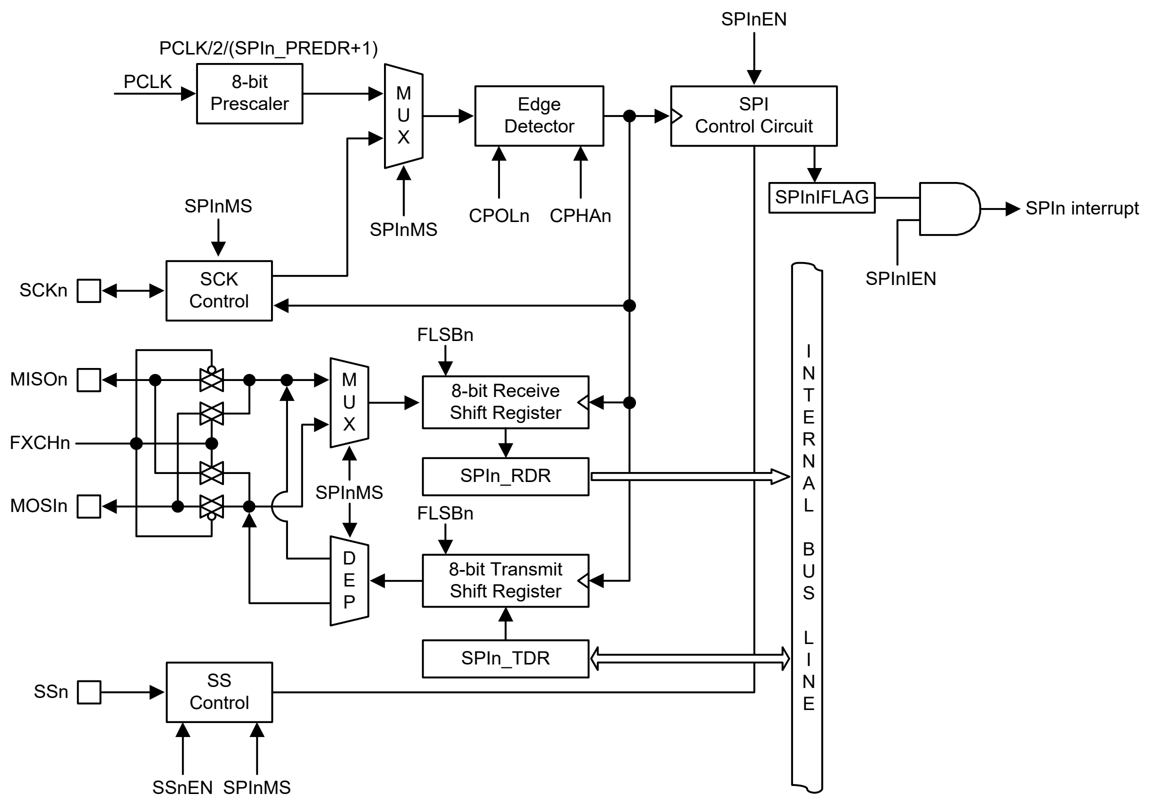
The SPI of the A31S134 features the followings:

- Selectable between the master and slave operations
- Full-duplex and four-wire synchronous transfers supported
  - SS: Slave Select
  - SCK: Serial Clock
  - MOSI: Master Output, Slave Input
  - MISO: Master Input, Slave Output
- SPI clock speed and polarity adjustable
- Up to 8 MHz of SPI clock supported
- Selectable between MSB first transfer or LSB first transfer
- DMA Tx, Rx transfers

14.3 SPI 0/1 Functional Description

14.3.1 Block Diagram

Figure 70. SPIn Block Diagram (n = 0 and 1)



NOTES:

1. The clock frequency of the SCKn should be less than or equal to the PCLK/2.
2. Interrupts of the SPIn occur only when one byte transmission/reception finishes. So, the status (SPInMS and SSnHIGH) bits should be checked by software.
3. When DMA is used, at least 32 clocks of PCLK are required for the interval between data-receptions.

14.4 Pin Description for SPI 0/1

Table 75. Pins and External Signals for SPI (n = 0 and 1)

Pin Name	Type	Description
SSn	I/O	SPIn Slave select input/output
SCKn	I/O	SPIn Serial clock input/output
MOSIn	I/O	SPIn Serial data (master output, slave input)
MISOn	I/O	SPIn Serial data (master input, slave output)

14.5 Functional Description

When SPIn block is enabled (SPInEN = ’1’), the slave select (SSn) pin becomes active-low input in slave mode operation if the SSnEN bit is set to ‘1’. The SSn function is not automatically controlled in master mode operation even if the SSnEN bit is set to ‘1’ (n = 0 and 1).

14.5.1 SPI Clock Formats and Timing

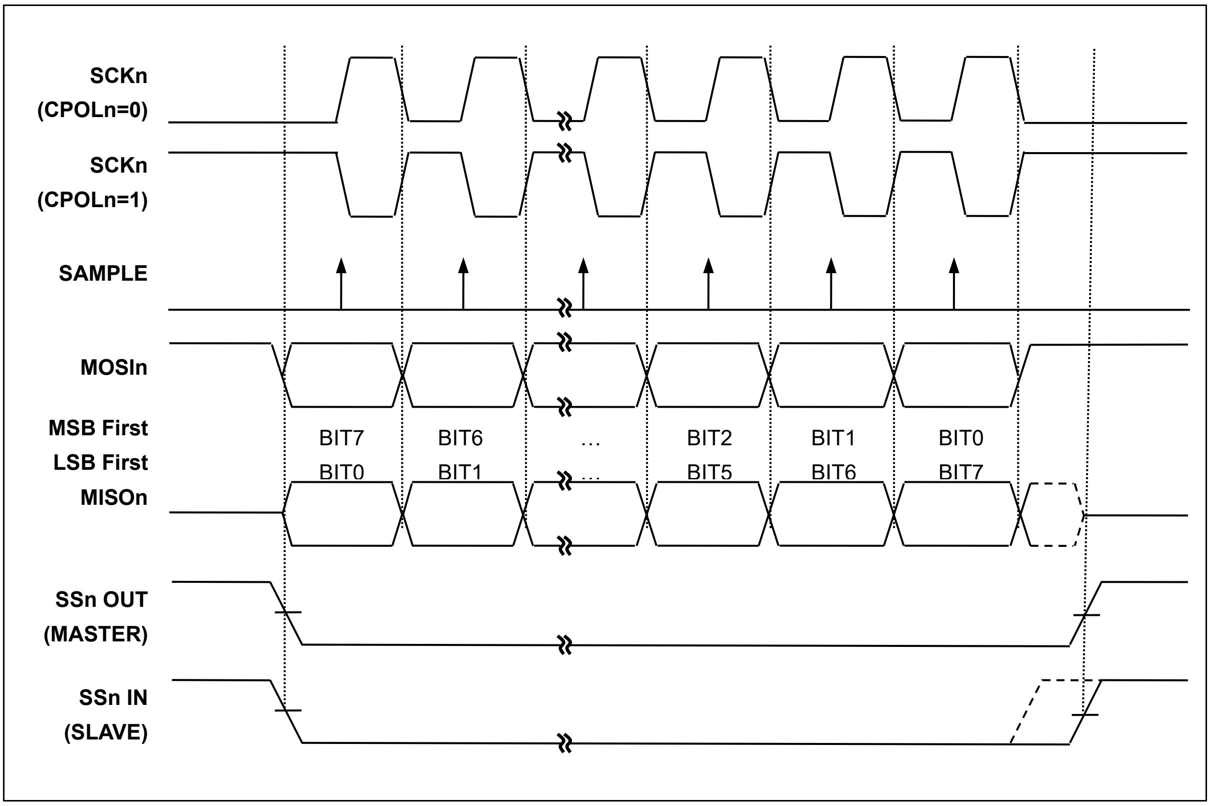
To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPIn has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. The CPOLn selectively inserts an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data.

Table 76 shows the four combinations of the CPOLn and CPHAn for SPIn (n = 0 and 1).

Table 76. CPOL Functionality (n = 0 and 1)

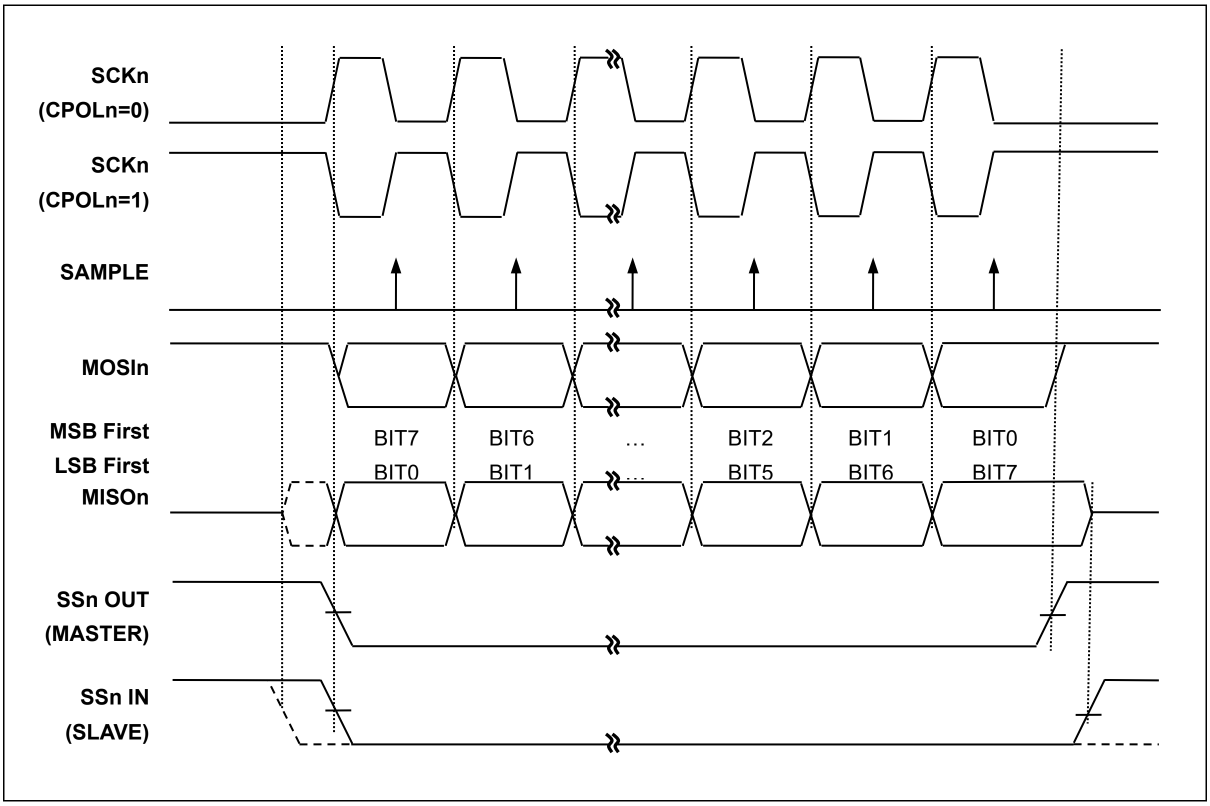
SPIn Mode	CPOLn	CPHAn	Leading Edge	Trailing Edge
0	0	0	Sample (rising)	Setup (falling)
1	0	1	Setup (rising)	Sample (falling)
2	1	0	Sample (falling)	Setup (rising)
3	1	1	Setup (falling)	Sample (rising)

Figure 71. SPIn Clock Formats when CPHAn=0 (n = 0 and 1)



When CPHAn=0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low. The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the second SCKn edge, the SPIn shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively (n = 0 and 1).

Figure 72. SPIn Clock Formats when CPHAn=1 (n = 0 and 1)



When CPHAn=1, the slave begins to drive its MISOn output when SSn input goes active low, but the data is not defined until the first SCKn edge.

The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave. The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the third SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively.

When CPHAn=1, the slave's SSn input is not required to go to its inactive high level between transfers.

14.6 SPI 0/1 Registers

The base address and register map of the SPI 0/1 are described in the following tables:

Table 77. Base Address of SPI 0/1

Name	Base Address
SPI0	0x4000_5800
SPI1	0x4000_5880

Table 78. SPIn Register Map

Name	Offset	Type	Description	Reset Value	Reference
SPIn_CR	0x00	RW	SPIn Control Register	0x0000_0000	14.6.1
SPIn_SR	0x04	RW	SPIn Status Register	0x0000_0000	14.6.2
SPIn_RDR	0x08	RO	SPIn Receive Data Register	0x0000_0000	14.6.3
SPIn_TDR	0x0C	RW	SPIn Transmit Data Register	0x0000_0000	14.6.4
SPIn_PREDR	0x10	RW	SPIn Prescaler Data Register	0x0000_03FF	14.6.5

NOTE:

1. n = 0 and 1.

14.6.1 SPIn_CR: SPIn Control Register

SPI module should be configured properly before running.

The SPIn_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

SPI0_CR=0x4000_5800, SPI1_CR=0x4000_5880

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved	SPInEN	FLSBn	SPInMS	Reserved	SPInIEN	Reserved	CPOLn	CPHAn
-	0	0	0	-	0	-	0	0
-	RW	RW	RW	-	RW	-	RW	RW

7	SPInEN	SPIn Operation Control. **NOTE:** 1. This bit should be set to '1' after setting the related registers.			
		0	Disable SPIn operation.		
		1	Enable SPIn operation.		
6	FLSBn	Data Transmission sequence selection.			
		0	MSB first.		
		1	LSB first.		
5	SPInMS	Master/Slave Selection.			
		0	Slave mode.		
		1	Master mode.		
3	SPInIEN	SPIn Interrupt Enable.			
		0	Disable SPIn interrupt.		
		1	Enable SPIn interrupt.		
1	CPOLn	Selects the clock polarity of SCK.			
		0	SCK to 0 when idle.		
		1	SCK to 1 when idle.		
0	CPHAn	The CPOLn and this bit determine if data are sampled on the leading or the trailing edge of SCK.			
		CPOLn	CPHAn	Leading edge	Trailing edge
		0	0	Sample (rising)	Setup (falling)
		0	1	Setup (rising)	Sample (falling)
		1	0	Sample (falling)	Setup (rising)
		1	1	Setup (falling)	Sample (rising)

14.6.2 SPIn_SR: SPIn Status Register

The SPIn_SR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

SPI0_SR=0x4000_5804, SPI1_SR=0x4000_5884

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			SPInIFLA	Reserved		SSnHIGH	Reserved		FXCHn	SSnEN
-			0	-	-	0	-	-	0	0
-			RW	-	-	RW	-	-	RW	RW

7	SPInIFLAG	SPIn Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when write ‘1’.
4	SSnHIGH	This bit is set when the SSn pin goes high level during the pin is the corresponding function.	
		0	No effect when ‘0’ is written.
		1	The SSn pin has gone from low level to high. This bit is cleared to ‘0’ when write ‘1’.
1	FXCHn	SPIn Pin Function Exchange Control.	
		0	No effect.
		1	Exchange MOSIn and MISOn function.
0	SSnEN	SSn Pin Operation Control.	
		0	Disable SSn pin operation.
		1	Enable SSn pin operation. The corresponding SSn Pin should be configured to the alternative function.

14.6.3 SPIn_RDR: SPIn Receive Data Register

The SPIn_RDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

SPI0_RDR=0x4000_5808, SPI1_RDR=0x4000_5888

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RDATA[7:0]
-			0x00
-			RO

7 0	RDATA[7:0]	SPIn Receive Data.

14.6.4 SPIn_TDR: SPIn Transmit Data Register

The SPIn_TDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

SPI0_TDR=0x4000_580C, SPI1_TDR=0x4000_588C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								TDATA[7:0]							
-																								0x00							
-																								RW							

7 0	TDATA[7:0]	SPIn Transmit Data. When it is written a byte to this data register, clock and data are sent out from SCKn and MOSIn pins of SPIn, respectively. **NOTE**: The data to be transmitted should be written after all control registers are set.

14.6.5 SPIn_PREDR: SPIn Prescaler Data Register

The SPIn_PREDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 and 1).

SPI0_PREDR=0x4000_5810, SPI1_PREDR=0x4000_5890

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						PRED[9:0]									
-																						0x3FF									
-																						RW									

9 0	PRED[9:0]	The value in this register is used to generate an SCK clock. SCKn clock: PCLK/2/(PRED[9:0] +1). The SCKn clock must be less than or equal to 8 MHz. The range is 0x00 to 0x3FF.

14.6.6 SPIn Register Map Summary

Table 79. SPIn Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPIn_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPInEN	FLSBn	SPInMS	Res.	SPInIEN	Res.	CPOLn	CPHAn
	Reset value																									0	0	0		0		0	0
0x04	SPIn_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPInIFLAG	Res.	Res.	SSnHIGH	Res.	Res.	FXCHn	SSnEN
	Reset value																									0			0			0	0
0x08	SPIn_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x0C	SPIn_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x10	SPIn_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[9:0]									
	Reset value																							1	1	1	1	1	1	1	1	1	1

NOTE:

1. n = 0 and 1.

15. Inter-Integrated Circuit (I2C)

15.1 Introduction

The I2C 0/1/2 (Inter-integrated Circuit) interface built in the A31S134 satisfies the standard I2C communication protocol and is used for serial communication with internal and external devices via the I2C protocol.

Equipped with three units, it supports both master and slave modes, and is capable of transmitting and receiving data in bytes by using interrupts or polling.

The I2C of the A31S134 operates in Standard mode (100 kHz), Fast mode (400 kHz) or Fast-Plus mode (1 MHz) and supports General call.

It helps communicate with various peripherals that have the same bus type. To use the I2C, it is recommended to set the SCL and SDA pins to open-drain and then connect external pull-up resistors to render their output signals ‘High’.

15.2 Main Features

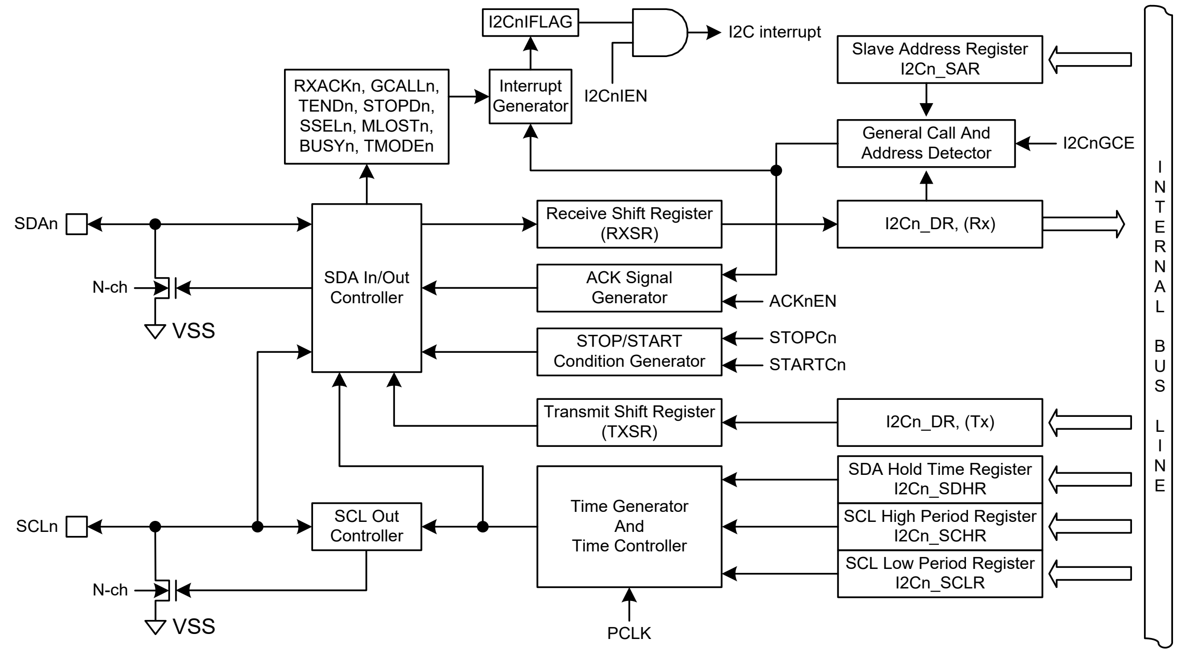
The I2C 0/1/2 of the A31S134 have the following features:

- Compatible with I2C bus standard
- Multi-master operation
- Up to 1 MHz data transfer read speed
- 7-bit address
- Two slave addresses supported
- Master and slave operations
- Bus busy detection

15.3 I2C 0/1/2 Block Diagram

Figure 73 shows a block diagram of the I2C module.

Figure 73. I2Cn Block Diagram (n = 0, 1 and 2)


I2CnIFLAG
I2C interrupt
Interrupt Generator
I2CnIEN
RXACKn, GCALLn, TENDn, STOPDn, SSELn, MLOSTn, BUSYn, TMODEn
Slave Address Register I2Cn_SAR
General Call And Address Detector
I2CnGCE
SDAn
N-ch
VSS
SDA In/Out Controller
Receive Shift Register (RXSR)
I2Cn_DR, (Rx)
ACK Signal Generator
ACKnEN
STOP/START Condition Generator
STOPCn
STARTCn
Transmit Shift Register (TXSR)
I2Cn_DR, (Tx)
SDA Hold Time Register I2Cn_SDHR
SCLn
SCL Out Controller
Time Generator And Time Controller
SCL High Period Register I2Cn_SCHR
SCL Low Period Register I2Cn_SCLR
N-ch
VSS
PCLK
INTERNAL BUS LINE

NOTE:

1. When the corresponding port of I2C is a alternative function for SCLn and SDAn pins, the SCLn and SDAn pins are automatically set as N-channel open-drain outputs and the input latch is read when reading the pins. The corresponding pull-up resistor is determined by the control register.

15.4 Pin Description for I2C 0/1/2

Table 80. Pins and External Signals for I2Cn (n = 0, 1, and 2)

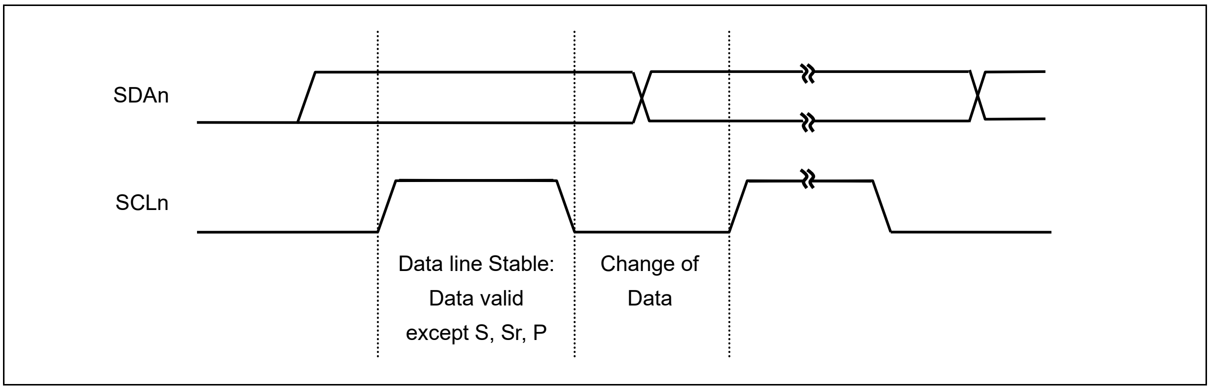
Pin Name	Type	Description
SCLn	I/O	I2C channel n Serial clock bus line (open-drain)
SDAn	I/O	I2C channel n Serial data bus line (open-drain)

15.5 I2C 0/1/2 Functional Description

15.5.1 I2C Bit Transfer

The data on the SDAn line must be stable during HIGH period of the clock, SCLn. The high or low state of the data line can only change when the clock signal on the SCLn line is low. The exceptions are START(S), repeated START(Sr), and STOP(P) condition, where data line changes when clock line is high.

Figure 74. I2C Bus Bit Transfer (n = 0, 1 and 2)



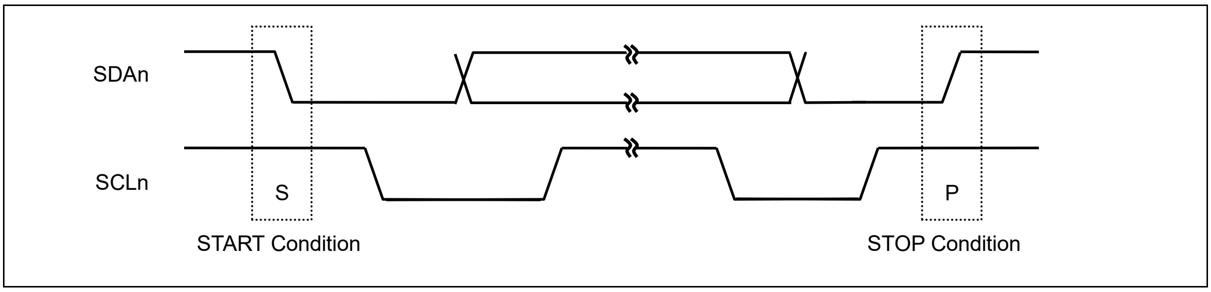
15.5.2 START/Repeated START/STOP

One master can issue a START (S) condition to detect other devices connected to the SCLn, SDAn lines that will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

- A high-to-low transition on the SDAn line while SCLn is high defines a START (S) condition.
- A low-to-high transition on the SDAn line while SCLn is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, i.e., the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays in busy mode. So, the START and repeated START conditions are functionally identical.

Figure 75. START and STOP Conditions (n = 0, 1, and 2)



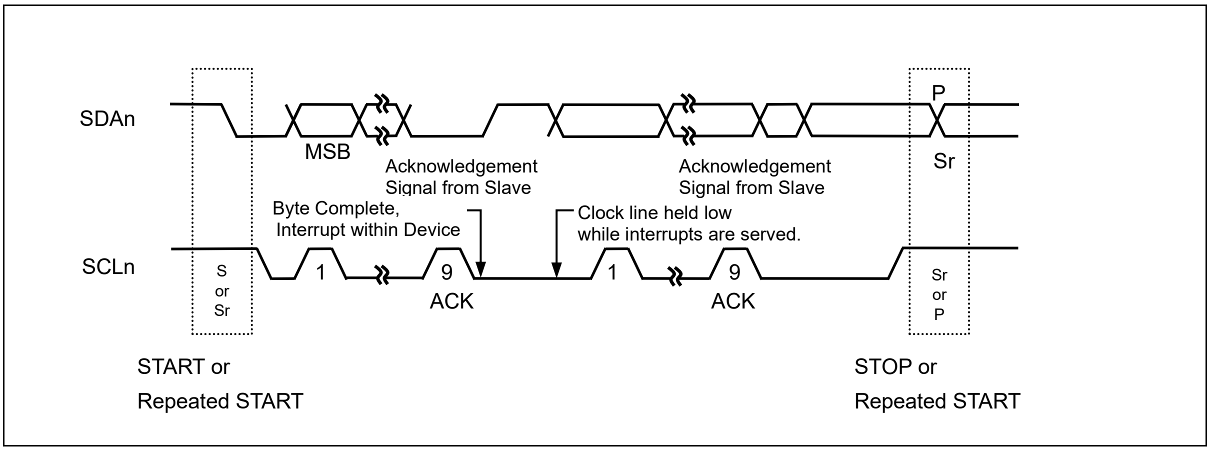
15.5.3 Data Transfer

Every byte on the SDAn line must be 8-bits long, but the number of bytes that can be transmitted per transfer is unlimited.

Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave cannot receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCLn LOW to force the master into a wait state.

Data transfer then continues when the slave is ready for another byte of data and releases clock line SCLn.

Figure 76. I2C Bus Data Transfer (n = 0, 1 and 2)



15.5.4 Acknowledge

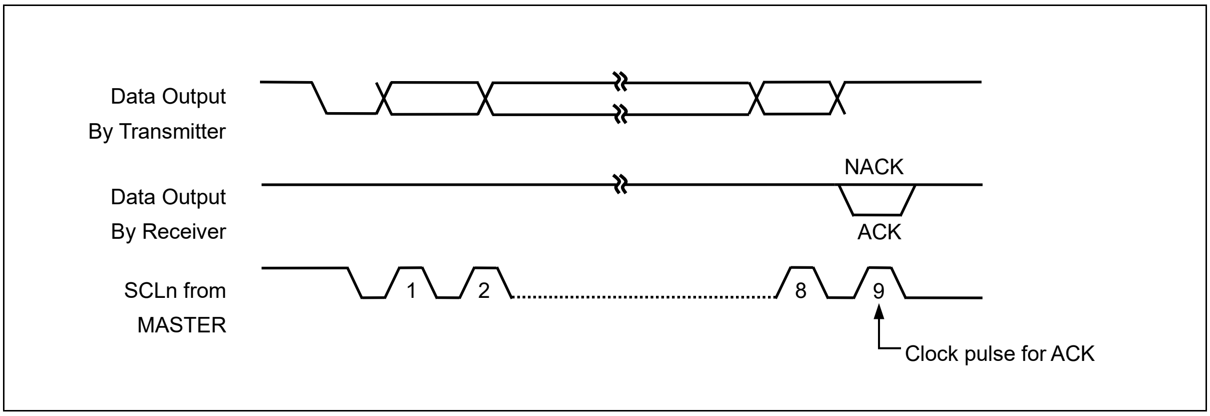
An acknowledge clock pulse is generated by the master. The transmitter releases the SDAn line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDAn line during the acknowledge clock pulse so that it remains stable at LOW during the HIGH period of this clock pulse.

When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it is performing some real time function, the data line must be left HIGH by the slave.

In addition, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDAn line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

Figure 77. I2C Bus Acknowledge (n = 0, 1, and 2)



15.5.5 Synchronization and Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCLn line. This means that a HIGH to LOW transition on the SCLn line will cause the devices concerned to start counting off their LOW period and it will hold the SCLn line in that state until the clock HIGH state is reached.

However, the LOW to HIGH transition of this clock may not change the state of the SCLn line if another clock is still within its LOW period. In this way, a synchronized SCLn clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period. A master may start a transfer only if the bus is free. Two or more masters may generate a START condition.

Arbitration takes place on the SDAn line, while the SCLn line is at the HIGH level, in such a way that a master that transmits a HIGH level, while another master that transmits a LOW level, will switch off its DATA output state because the level on the bus does not correspond to its own level.

Arbitration continues for many bits until a winning master gets the ownership of I2C bus. Its first stage is comparison of the address bits.

Figure 78. Clock Synchronization during Arbitration Procedure (n = 0, 1, and 2)

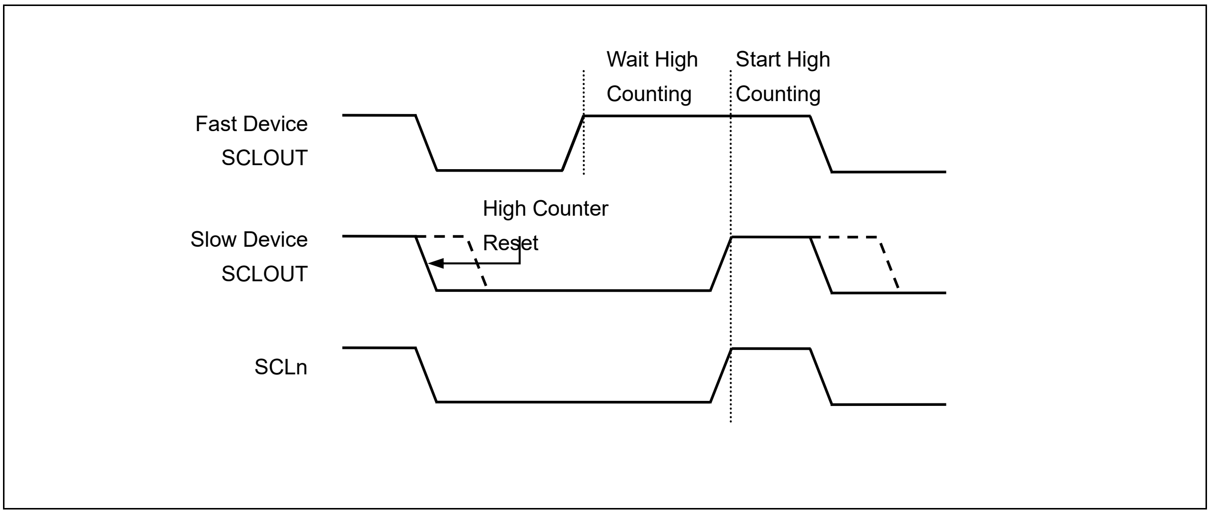
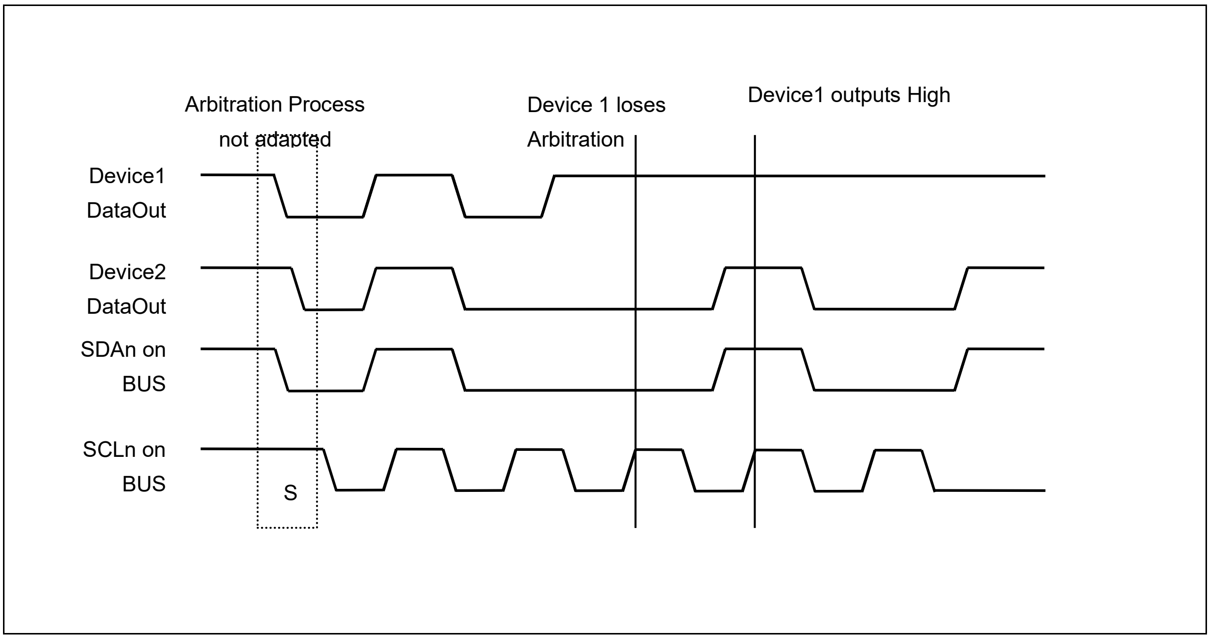


Figure 79. Arbitration Procedure between Two Masters (n = 0, 1, and 2)



15.6 I2C Operation

The I2C is byte-oriented and interrupt-based. Interrupts are issued after all bus events except for the transmission of a START condition. Since I2C is interrupt based, the application software is free to carry on with other operations during an I2C byte transfer.

Note that when an I2C interrupt is generated, I2CnIFLAG flag in I2Cn_CR register is set, and it is cleared when all interrupt source bits in the I2Cn_ST register are cleared to ‘0’. When I2C interrupt occurs, the SCLn line is held at LOW until all interrupt source bits in I2Cn_ST register are cleared to ‘0’. When the I2CnIFLAG flag is set, the I2Cn_ST contains a value indicating the current state of the I2C bus. According to the value in I2Cn_ST, software can decide what to do next.

I2C can operate in four modes: master/slave, transmitter/receiver. The operating mode is configured by a winning master.

A more detailed explanation follows below. (n = 0, 1, and 2)

15.6.1 Master Transmitter

To operate I2C as a master transmitter, follow the recommended steps below.

1. Enable I2C by setting I2CnEN bit in I2Cn_CR. This provides the main clock to the peripheral.
2. Load SLA+W into the I2Cn_DR, where SLA is the address of slave device and W is the transfer direction from the viewpoint of master. For master transmitter, W is ‘0’. Note that I2Cn_DR is used for both address and data.
3. Configure baud rate by writing desired value to both I2Cn_SCLR and I2Cn_SCHR for the Low and High period of SCLn line.
4. Configure the I2Cn_SDHR to decide when SDAn changes value from falling edge of SCLn. If SDA should change in the middle of SCLn LOW period, load half the value of I2Cn_SCLR to the I2Cn_SDHR.
5. Set the STARTCn bit in I2Cn_CR. This transmits a START condition. Also, configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in I2Cn_DR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of receiving ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in I2Cn_ST is set, and I2C waits in idle state or can be operated as an addressed slave.

To operate as a slave when the MLOSTn bit in I2Cn_ST is set, the ACKnEN bit in I2Cn_CR must be set and the received 7-bit address must match the SLAn bits in I2Cn_SAR1/2. In this case, I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. The reason for this is to decide whether I2C should continue serial transfer or stop communication. The following steps continue, assuming that I2C does not lose mastership during the first data transfer.

I2C (Master) can choose one of the following cases regardless of receiving ACK signal from slave.

A. Master receives an ACK signal from the slave. Thereby, the master can send data to the slave while continuing data transfer. In this case, the data to be transmitted is written to I2Cn_DR.

B. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC bit in I2Cn_CR.

C. Master transmits repeated START condition without checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in I2Cn_CR.

After doing any of the actions above, clear all interrupt source bits in I2Cn_ST to ‘0’ to release SCLn line. In the case of A, move to step 7. In case of B, move to step 9 to handle STOP interrupt. In case of C, move to step 6 after transmitting the data in I2Cn_DR, and if transfer direction bit is ‘1’, go to master receiver section.

7. 1-Byte of data is transmitted. During data transfer, bus arbitration continues.

8. This is ACK signal processing stage for data packets transmitted by master. I2C holds the SCLn LOW. When I2C loses bus mastership while transmitting data to arbitrate other masters, the MLOSTn bit in I2Cn_ST is set. If then, I2C waits in idle state. When the data in I2Cn_DR is transmitted completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases regardless of receiving ACK signal from slave.

   A. Master receives ACK signal from slave, so continues data transfer since slave can receive more data from master. In this case, load data to transmit to I2Cn_DR.

   B. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in I2Cn_CR.

   C. Master transmits repeated START condition without checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set the STARTCn bit in I2Cn_CR.

After doing any of the actions above, clear all interrupt source bits in the I2Cn_ST register to ‘0’ to release SCL line. In the case of A, move to step 7. In case of B, move to step 9 to handle STOP interrupt. In case of C, move to step 6 after transmitting the data in the I2CDR register, and if transfer direction bit is ‘1’, go to master receiver section.

9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write “0xff” to the I2Cn_ST register. After this, I2C enters idle state.

15.6.2 Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

1. Enable I2C by setting I2CnEN bit in the I2Cn_CR register. This provides the main clock to the peripheral.

2. Load SLA+R into the I2Cn_DR register, where SLA is the address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is ‘1’. Note that the I2Cn_DR register is used for both address and data.

3. Configure baud rate by writing desired value to both the I2Cn_SCLR and I2Cn_SCHR registers for the low and high period of SCLn line.

4. Configure the I2Cn_SDHR register to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of I2Cn_SCLR to the I2Cn_SDHR register.

5. Set the STARTCn bit in I2Cn_CR. This transmits a START condition. Also, configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in the I2Cn_DR register is transmitted out according to the baud-rate.

6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of receiving ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in the I2Cn_ST register is set, and I2C waits in idle state or can be operated as an addressed slave.

   To operate as a slave when the MLOSTn bit in the I2Cn_ST register is set, the ACKnEN bit in the I2Cn_CR register must be set, and the received 7-bit address must equal to the SLAn bits in I2Cn_SAR1/2. In this case, I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. The reason for this is to decide whether I2C should continue serial transfer or stop communication. The following steps continue, assuming that I2C does not lose mastership during the first data transfer.

I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

A. Master receives ACK signal from slave, so continues data transfer since slave can prepare and transmit more data to master. Configure the ACKnEN bit in the I2Cn_CR register to decide whether I2C should check Acknowledges for the subsequent data to be received or not.

B. Master stops data transfer since it receives no ACK signal from slave. In this case, set the STOPCn bit in the I2Cn_CR register.

C. Master transmits repeated START condition due to lack of ACK signal from slave. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in the I2Cn_CR register.

After doing any of the actions above, clear all interrupt source bits in the I2Cn_ST register to ‘0’ to release SCLn line. In the case of A, move to step 7. In the case of B, move to step 9 to handle STOP interrupt. In case of C, move to step 6 after transmitting the data in the I2Cn_DR register, and if transfer direction bit is ‘0’, go to master transmitter section.

7. 1-byte of data is received.

8. This is ACK signal processing stage for data packets transmitted by slave. I2C holds the SCLn LOW. When 1-Byte of data is received completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases according to the RXACKn flag in the I2Cn_ST register.

   A. Master continues receiving data from slave. To do this, set the ACKnEN bit in the I2Cn_CR register to acknowledge the next data to be received.

   B. Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing the ACKnEN bit in the I2Cn_CR register.

   C. Since no ACK signal is detected, the master terminates data transfer. In this case, set the STOPCn bit in the I2Cn_CR register.

   D. No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA+R/W into the I2Cn_DR register and set the STARTCn bit in the I2Cn_CR register.

   After doing any of the actions above, clear all interrupt source bits in the I2Cn_ST register to ‘0’ to release SCLn line. In the cases of A and B, move to step 7. In case of C, move to step 9 to handle STOP interrupt. In the case of D, move to step 6 after transmitting the data in I2Cn_DR, and if transfer direction bit is ‘0’, go to master transmitter section.

9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear the I2Cn_ST register, write “0xff” value to the I2Cn_ST register. After this, I2C enters idle state.

15.6.3 Slave Transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

1. If the operating clock (HCLK) of the system is slower than that of SCLn, load value 0x00 into the I2Cn_SDHR register to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of HCLK where SDAH is multiple of number of HCLK coming from I2Cn_SDHR. When the hold time of SDAn is longer than the period of HCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting the I2CnIEN bit and I2CnEN bit in the I2Cn_CR register. This provides the main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with the SLAn bit fields in the I2Cn_SAR1/2 registers. If the GCALLnEN bit in the I2Cn_SAR1/2 registers is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not match the SLAn bit field in the I2CnSAR register, I2C enters idle state, i.e., waits for another START condition. Otherwise, if the address equals SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address matches SLAn bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs, load transmit data to I2Cn_DR and clear all interrupt source bits in I2Cn_ST to ‘0’ to release SCLn line.
5. 1-byte of data is transmitted.
6. In this step, I2C generates a TENDn interrupt and holds the SCLn line LOW regardless of receiving ACK signal from master. Slave can select one of the following cases.

   A. No ACK signal is detected and I2C waits for STOP or repeated START condition.

   B. ACK signal from the master is detected and load data to transmit into the I2Cn_DR register.

   After doing any of the actions above, clear all interrupt source bits in the I2Cn_ST register to ‘0’ to release SCLn line. In the case of A, move to step 7 to terminate communication. In the case of B, move to step 5. In either case, a repeated START condition can be detected. For that case, move to step 4.
7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear the I2Cn_ST register, write “0xff” to the I2Cn_ST register. After this, I2C enters idle state.

15.6.4 Slave Receiver

To operate I2C in slave receiver, follow the recommended steps below.

1. If the operating clock (HCLK) of the system is slower than that of SCLn, load value 0x00 into I2Cn_SDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by (SDAH × period of HCLK) where SDAH is multiple of number of HCLK coming from the I2Cn_SDHR register. When the hold time of SDAn is longer than the period of HCLK, I2C (slave) cannot transmit serial data properly.

2. Enable I2C by setting I2CnIEN bit in the I2Cn_CR register. This provides the main clock to the peripheral.

3. When a START condition is detected, I2C receives one byte of data and compares it with the SLA value in the I2CSAR register. If the GCALLnEN bit in the I2Cn_SAR1/2 registers is enabled, I2C compares the received data with value 0x00, the general call address.

4. If the received address does not match the SLAn values in the I2Cn_SAR1/2 registers, I2C enters idle state i.e., waits for another START condition. Otherwise, if the address matches the SLAn value and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held low. Note that even if the address equals to SLA value, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs and I2C is ready to receive data, clear all interrupt source bits in the I2Cn_ST register to ‘0’ to release SCLn line.

5. 1-byte of data is received.

6. In this step, I2C generates a TENDn interrupt and holds the SCLn line low regardless of receiving ACK signal from master. Slave can select one of the following cases.

   A. No ACK signal is detected (ACKnEN=0) and I2C waits STOP or repeated START condition.

   B. ACK signal is detected (ACKnEN=1) and I2C can continue to receive data from master.

   After doing any of the actions above, clear all interrupt source bits in the I2Cn_ST register to ‘0’ to release SCLn line. In the case of A, move to step 7 to terminate communication. In the case of B, move to step 5. In either case, a repeated START condition can be detected. For that case, move to step 4.

7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear the I2Cn_ST register, write “0xff” to the I2Cn_ST register. After this, I2C enters idle state.

15.7 I2C registers

The base address and register map of the I2C module are described in the following tables:

Table 81. Base Address of I2Cn Interface

Name	Base Address
I2C0	0x4000_4800
I2C1	0x4000_4900
I2C2	0x4000_4A00

Table 82. I2C Register Map

Name	Offset	Type	Description	Reset Value	Reference
I2Cn_CR	0x00	RW	I2Cn Control Register	0x0000_0000	15.7.1
I2Cn_ST	0x04	RW	I2Cn Status Register	0x0000_0000	15.7.2
I2Cn_SAR1	0x08	RW	I2Cn Slave Address Register 1	0x0000_0000	15.7.3
I2Cn_SAR2	0x0C	RW	I2Cn Slave Address Register 2	0x0000_0000	15.7.4
I2Cn_DR	0x10	RW	I2Cn Data Register	0x0000_0000	15.7.5
I2Cn_SDHR	0x14	RW	I2Cn SDA Hold Time Register	0x0000_0001	15.7.6
I2Cn_SCLR	0x18	RW	I2Cn SCL Low Period Register	0x0000_003F	15.7.7
I2Cn_SCHR	0x1C	RW	I2Cn SCL High Period Register	0x0000_003F	15.7.8

NOTE:

1. n = 0, 1, and 2.

15.7.1 I2Cn_CR: I2Cn Control Register

The register can be set to configure I2C operation mode to activate I2C transactions.

The I2Cn_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_CR=0x4000_4800, I2C1_CR=0x4000_4900, I2C2_CR=0x4000_4A00

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			I2CnEN	TXDLYENBn	I2CnIEN	I2CnIFLAG	ACKnEN	IMASTERn	STOPCn	STARTCn
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RO	RW	RO	RW	RW

7	I2CnEN	Activate I2Cn Block.	
		0	Disable I2Cn block.
		1	Enable I2Cn block.
6	TXDLYENBn	I2Cn_SDHR Register Control.	
		0	Enable I2Cn_SDHR register.
		1	Disable I2Cn_SDHR register.
5	I2CnIEN	I2Cn Interrupt Enable.	
		0	Disable I2Cn interrupt.
		1	Enable I2Cn interrupt.
4	I2CnIFLAG	I2Cn Interrupt Flag. This bit is cleared when all interrupt source bits in the I2Cn_ST register are cleared to ‘0’.	
		0	No request occurred.
		1	Request occurred.
3	ACKnEN	Controls ACK signal generation at ninth SCL period.	
		0	No ACK signal is generated. (SDA = 1)
		1	ACK signal is generated. (SDA = 0)
		NOTES: 1. ACK signal is output (SDA = 0) for the following 3 cases. 2. When received address packet is equal to SLAn[6:0] bits in I2Cn_SAR1/I2Cn_SAR2 register. 3. When received address packet is equal to value 0x00 with GCALLn enabled. 4. When I2Cn operates as a receiver (master or slave)	
2	IMASTERn	Represents Operation Mode of I2Cn. This bit is cleared to ‘0’ on STOP condition.	
		0	I2Cn is in slave mode.
		1	I2Cn is in master mode.
1	STOPCn	STOP Condition Generation When I2Cn is master.	
		0	No effect.
		1	Generate STOP condition.
0	STARTCn	START Condition Generation When I2Cn is master.	
		0	No effect.
		1	Generate START or Repeated START condition.

15.7.2 I2Cn_ST: I2Cn Status Register

The I2Cn_ST register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_ST=0x4000_4804, I2C1_ST=0x4000_4904, I2C2_ST=0x4000_4A04

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RO	RW

7	GCALLn	This bit has different meaning depending on whether I2C is master or slave. When I2C is a master, this bit represents whether it received AACK (address ACK) from slave.	
		0	No AACK is received. (Master mode)
		1	AACK is received (Master mode). It may be set to '1' after address transmission.
		When I2C is a slave, this bit is used to indicate general call.	
		0	General call address is not detected. (Slave mode)
		1	General call address is detected. (Slave mode)
6	TENDn	This bit is set when 1-byte of data is transferred completely.	
		0	1 byte of data is not completely transferred.
		1	1 byte of data is completely transferred.
5	STOPDn	This bit is set when a STOP condition is detected.	
		0	A STOP condition is not detected.
		1	A STOP condition is detected.
		NOTE: 1. A STOP condition is not detected on unaddressed slaves.	
4	SSELn	This bit is set when I2C is addressed by other master.	
		0	I2C is not selected as a slave.
		1	I2C is addressed by other master and acts as a slave.
3	MLOSTn	This bit represents the result of bus arbitration in master mode.	
		0	I2C maintains bus mastership.
		1	I2C has lost bus mastership during arbitration process.
2	BUSYn	This bit reflects bus status.	
		0	I2C bus is idle, so a master can issue a START condition.
		1	I2C bus is busy.
1	TMODEn	This bit is used to indicate whether I2C is transmitter or receiver.	
		0	I2C is a receiver.
		1	I2C is a transmitter.
0	RXACKn	This bit shows the state of ACK signal.	
		0	No ACK is received.
		1	ACK is received at ninth SCL period.

NOTES:

1. The GCALLn, TENDn, STOPDn, SSELn, and MLOSTn bits can be source of interrupt.
2. When an I2C interrupt occurs except for DEEP SLEEP mode, the SCL line is held low. To release SCL, Clear to '0' all interrupt source bits in I2Cn_ST register.
3. The GCALLn, TENDn, STOPDn, SSELn, MLOSTn, and RXACKn bits are cleared when '1' is written to the corresponding bit.

15.7.3 I2Cn_SAR1: I2Cn Slave Address Register 1

The I2Cn_SAR1 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_SAR1=0x4000_4808, I2C1_SAR1=0x4000_4908, I2C2_SAR1=0x4000_4A08

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			SLAn[6:0]							GCALLnEN
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

7 1	SLAn[6:0]	These bits configure the slave address 1 in slave mode.	
0	GCALLnEN	This bit decides whether I2Cn allows general call address 1 or not in I2Cn slave mode.	
		0	Ignore general call address 1.
		1	Allow general call address 1.

15.7.4 I2Cn_SAR2: I2Cn Slave Address Register 2

The I2Cn_SAR2 register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_SAR2=0x4000_480C, I2C1_SAR2=0x4000_490C, I2C2_SAR2=0x4000_4A0C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			SLAn[6:0]							GCALLnEN
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

7 1	SLAn[6:0]	These bits configure the slave address 2 in slave mode.	
0	GCALLnEN	This bit decides whether I2Cn allows general call address 2 or not in I2Cn slave mode.	
		0	Ignore general call address 2.
		1	Allow general call address 2.

15.7.5 I2Cn_DR: I2Cn Data Register

The I2Cn_DR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_DR=0x4000_4810, I2C1_DR=0x4000_4910, I2C2_DR=0x4000_4A10

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved	DATA[7:0]							
-	0	0	0	0	0	0	0	0
-	RW	RW	RW	RW	RW	RW	RW	RW

7 0	DATA[7:0]	The I2Cn_DR Transmit buffer and Receive buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the I2Cn_DR register. Reading the I2Cn_DR register returns the contents of the Receive Buffer.

15.7.6 I2Cn_SDHR: I2Cn SDA Hold Time Register

The I2Cn_SDHR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_SDHR=0x4000_4814, I2C1_SDHR=0x4000_4914, I2C2_SDHR=0x4000_4A14

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Reserved	HLDT[11:0]
-	0x001
-	RW

11 0	HLDT[11:0]	This register is used to control SDA output timing from the falling edge of SCL. Note that SDA is changed after tPCLK X (I2Cn_SDHR+2). In master mode, load half the value of I2Cn_SCLR to this register to make SDA switch in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after tPCLK X (I2Cn_SDHR+2) in master mode. So, to ensure proper operation in slave mode, the value tPCLK X (I2Cn_SDHR + 3) must be smaller than the period of SCL.

15.7.7 I2Cn_SCLR: I2Cn SCL Low Period Register

The I2Cn_SCLR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_SCLR=0x4000_4818, I2C1_SCLR=0x4000_4918, I2C2_SCLR=0x4000_4A18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				SCLL[11:0]											
-																				0x03F											
-																				RW											

11 0	SCLL[11:0]	This register defines the low period of SCL in master mode. The base clock is PCLK, and the period is calculated by the formula: tPCLK × (4 × I2Cn_SCLR + 3) where tPCLK is the period of PCLK.

15.7.8 I2Cn_SCHR: I2Cn SCL High Period Register

The I2Cn_SCHR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0, 1, and 2).

I2C0_SCHR=0x4000_481C, I2C1_SCHR=0x4000_491C, I2C2_SCHR=0x4000_4A1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				SCLH[11:0]											
-																				0x03F											
-																				RW											

11 0	SCLH[11:0]	This register defines the high period of SCL in master mode. The base clock is PCLK, and the period is calculated by the formula: tPCLK × (4 × I2Cn_SCHR + 3) where tPCLK is the period of PCLK.

15.7.9 I2C Register Map Summary

Table 83. I2C Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	I2Cn_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2CnEN	TXDLYENBn	I2CnIEN	I2CnIFLAG	ACKnEN	IMASTERn	STOPCn	STARTCn
	Reset value																									0	0	0	0	0	0	0	0
0x04	I2Cn_ST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
	Reset value																									0	0	0	0	0	0	0	0
0x08	I2Cn_SAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLAn[6:0]							GCALLnEN
	Reset value																									0	0	0	0	0	0	0	0
0x0C	I2Cn_SAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLAn[6:0]							GCALLnEN
	Reset value																									0	0	0	0	0	0	0	0
0x10	I2Cn_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x14	I2Cn_SDHR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HLDT[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	1
0x18	I2Cn_SCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCLL[11:0]											
	Reset value																					0	0	1	1	1	1	1	1	1	1	1	1
0x1C	I2Cn_SCHR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCLH[11:0]											
	Reset value																					0	0	1	1	1	1	1	1	1	1	1	1

16. Analog-to-Digital Converters (ADC)

16.1 Introduction

The ADC (Analog-to-Digital Converter) of the A31S134 allows conversion of an analog input signal to a corresponding 12-bit digital value. Its A/D module has fourteen analog inputs as shown in Figure 80. Output of the multiplexer is the input into the converter, which generates the result through successive approximation.

The A/D module has three registers such as a control register (ADC_CR), a data register (ADC_DR), and a prescaler data register (ADC_PREDR). The channels to be converted are selected by setting the ADSEL[3:0] bit field. The ADC_DR register contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADC_DR register, A/D conversion status bit ADCIFLAG is set to ‘1’, and the A/D interrupt is set. During A/D conversion, the ADCIFLAG bit is read as ‘0’.

16.2 Main Features

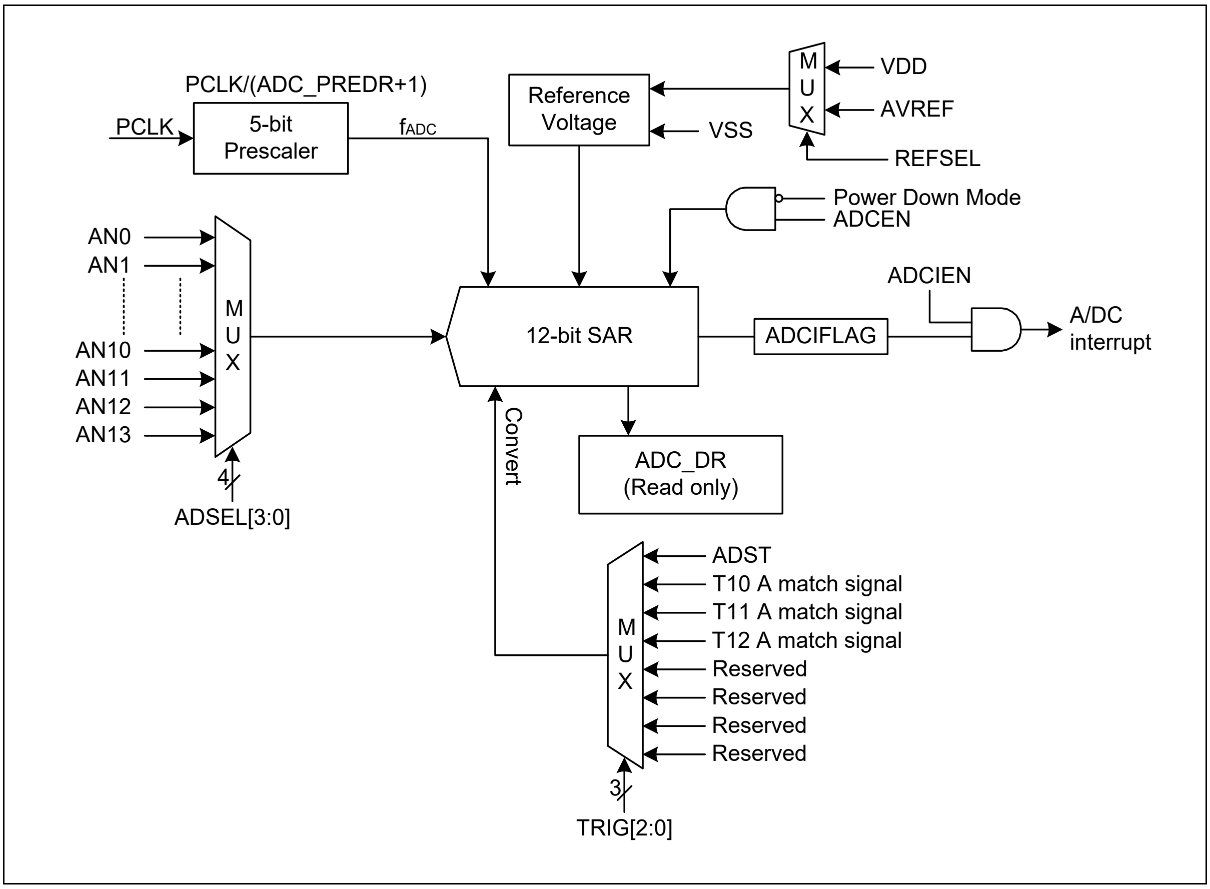
The main features of the ADC module are listed below:

- 14-channel of analog inputs
- Software trigger (ADST) and Timer trigger (T10/11/12 A match) support
- Conversion time: 58 clocks
- 5-bit prescaler

16.3 ADC Block Diagram

Figure 80 shows a block diagram of the 12-bit ADC.

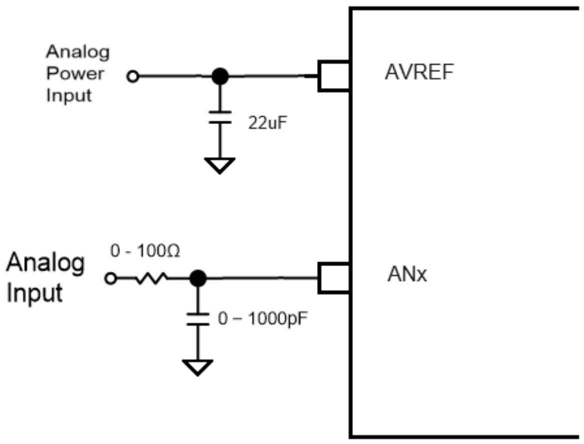
Figure 80. 12-bit ADC Block Diagram



16.4 ADC Recommend Circuit

Figure 81 shows a recommend circuit diagram of the ADC.

Figure 81. Recommend Circuit for ADC Input



16.5 Pin Description for ADC

Pin Name	Type	Description
VDD	P	Analog/Digital Power
VSS	P	Analog/Digital GND
AVREF	P	Analog Reference Voltage
AN0	A	ADC Input 0
AN1	A	ADC Input 1
AN2	A	ADC Input 2
AN3	A	ADC Input 3
AN4	A	ADC Input 4
AN5	A	ADC Input 5
AN6	A	ADC Input 6
AN7	A	ADC Input 7
AN8	A	ADC Input 8
AN9	A	ADC Input 9
AN10	A	ADC Input 10
AN11	A	ADC Input 11
AN12	A	ADC Input 12
AN13	A	ADC Input 13

NOTE:

1. P: Power, A: Analog.

16.6 ADC Functional Description

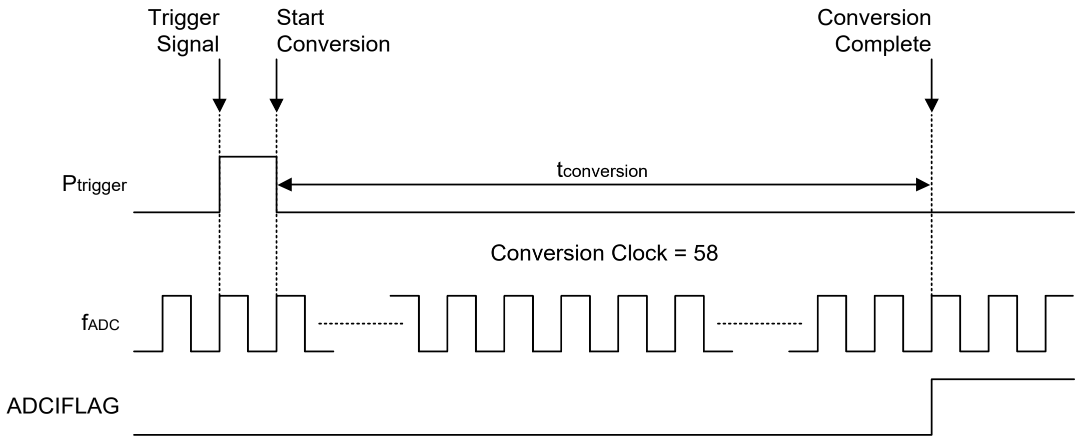
16.6.1 ADC Internal Channel Wiring

The A/D conversion process requires four steps (four clock edges) to convert each bit and 10 clocks to set up A/D conversion. Therefore, a total of 58 clocks are required to complete a 12-bit conversion: When the frequency of A/D converter is 1 MHz, one clock cycle is 1 μs. Each bit conversion requires four clocks. The conversion rate is calculated as follows:

$$4 \text{ clocks/bit} \times 12 \text{ bits} + \text{set-up time} = 58 \text{ clocks},$$

$$58 \text{ clocks} \times 1\ \mu s = 58\ \mu s \text{ at } 1 \text{ MHz}$$

Figure 82. 12-bit ADC Converter Timing Chart



NOTE:

1. The conversion time needs more than 2 0μs when VDD ≥ 2.7 V.

16.7 ADC Registers

The base address and register map of the ADC are described in the followings:

Table 84. Base Address of ADC

Name	Base Address
ADC	0x4000_3000

Table 85. ADC Register Map

Name	Offset	Type	Description	Reset Value	Reference
ADC_CR	0x0000	RW	A/D Converter Control Register	0x0000_0000	16.7.1
ADC_DR	0x0004	RO	A/D Converter Data Register	-	16.7.2
ADC_PREDR	0x0008	RW	A/D Converter Prescaler Data Register	0x0000_000F	16.7.3

16.7.1 ADC_CR: A/D Converter Control Register

The A/D Converter module should be configured properly before running.

The ADC_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_CR=0x4000_3000

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15	14	13 12 11	10	9	8	7 6	5	4	3 2 1 0
Reserved	ADCEN	Reserved	TRIG[2:0]	REFSEL	Reserved	ADST	Reserved	ADCIEN	ADCIFLAG	ADSEL[3:0]
-	0	0	000	0	-	0	-	0	0	0000
-	RW	RW	RW	RW	-	RW	-	RW	RW	RW

15	ADCEN	ADC Module Enable. (The ADC is automatically disabled at power down mode)	
		0	Disable ADC module operation.
		1	Enable ADC module operation.
13 11	TRIG[2:0]	ADC Trigger Signal Selection.	
		000	ADST.
		001	Timer 10 A-match signal.
		010	Timer 11 A-match signal.
		011	Timer 12 A-match signal.
		Others	Reserved
10	REFSEL	ADC Reference Selection.	
		0	Select analog power. (VDD)
		1	Select external reference. (AVREF)
8	ADST	ADC Conversion Start. This bit is automatically cleared to ‘0’ after operation.	
		0	No effect.
		1	Trigger signal generation for conversion start.
5	ADCIEN	ADC Interrupt Enable.	
		0	Disable ADC interrupt.
		1	Enable ADC interrupt.
4	ADCIFLAG	ADC Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
3 0	ADSEL[3:0]	A/D Converter Channel Selection.	
		0000	AN0
		0001	AN1
		0010	AN2
		0011	AN3
		0100	AN4
		0101	AN5
		0110	AN6
		0111	AN7
		1000	AN8
		1001	AN9
		1010	AN10
		1011	AN11
		1100	AN12

1101	AN13
Others	Reserved

16.7.2 ADC_DR: A/D Converter Data Register

The ADC_DR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_DR=0x4000_3004

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="20">Reserved</td><td colspan="12">ADDATA[11:0]</td></tr>
<tr><td colspan="20">-</td><td colspan="12">0xXXX</td></tr>
<tr><td colspan="20">-</td><td colspan="12">RO</td></tr>
</table>

11 0	ADDATA[11:0]	A/D Converter Result Data.

16.7.3 ADC_PREDR: A/D Converter Prescaler Data Register

The ADC_PREDR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_PREDR=0x4000_3008

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="27">Reserved</td><td colspan="5">PRED[4:0]</td></tr>
<tr><td colspan="27">-</td><td colspan="5">01111</td></tr>
<tr><td colspan="27">-</td><td colspan="5">RW</td></tr>
</table>

4 0	PRED[4:0]	A/D Converter Prescaler Data.

NOTES:

1. The prescaler sets the A/D conversion clock. The frequency of A/D converter should be less than 3 MHz because the conversion time needs at least 20 µs.
2. If the A/D frequency is more than 3 MHz, malfunction may occur.

Table 86. ADC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ADCEN	Res	TRIG[2:0]			REFSEL	Res	ADST	Res	Res	ADCIEN	ADCIFLAG	ADSEL[3:0]			
	Reset value																	0		0	0	0	0		0			0	0	0	0	0	0
0x04	ADC_DR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ADDATA[11:0]											
	Reset value																					Unknown											
0x08	ADC_PREDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRED[4:0]				
	Reset value																												0	0	0	0	0

17. Cyclic Redundancy Check (CRC) and Checksum

17.1 Introduction

A CRC (Cyclic Redundancy Check) generator is used to obtain 16-bit CRC code of Flash ROM and any data stream.

Among other applications, the CRC-based techniques are used to verify data transmission or storage integrity. In the scope of functional safety standards, they offer means of verifying Flash memory’s integrity.

The CRC generator helps computing the signature of the software during runtime, comparing with a reference signature.

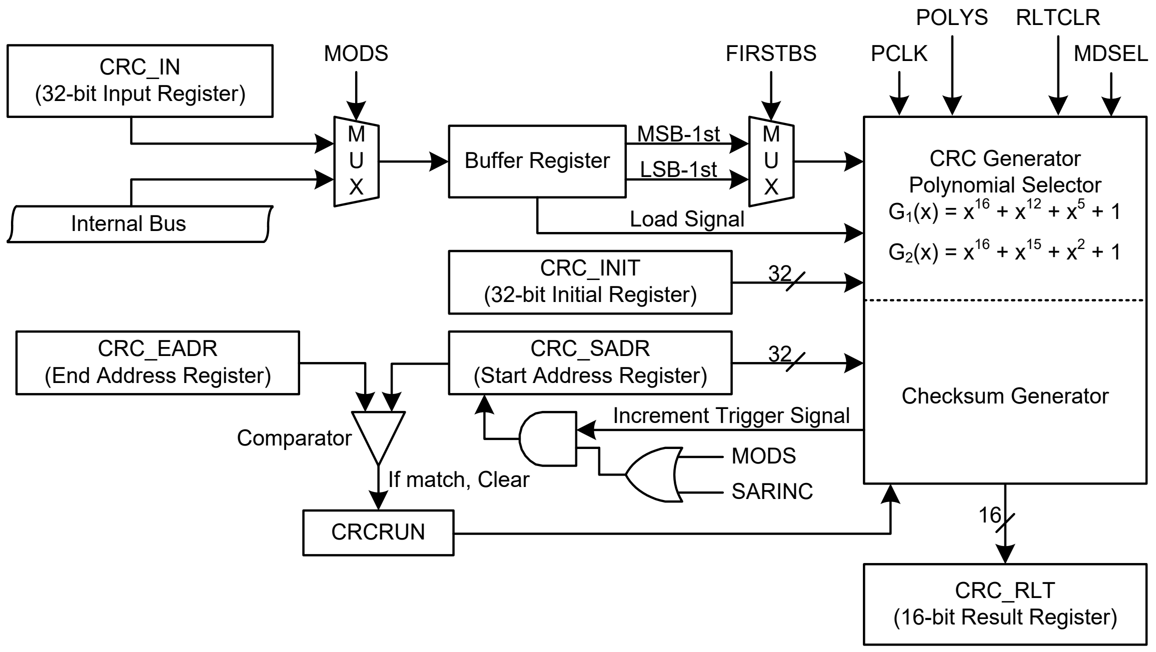
17.2 Main Features

The CRC generator of the A31S134 features the followings:

- Auto CRC and user CRC mode
- CRC-CCITT ($G1(x) = x^{16} + x^{12} + x^5 + 1$) supported
- CRC-16 ($G2(x) = x^{16} + x^{15} + x^2 + 1$) supported
- CRC and checksum mode
- CRC/checksum start address auto increment (user mode only)

17.3 CRC and Checksum Block Diagram

Figure 83. CRC and Checksum Timer Block Diagram



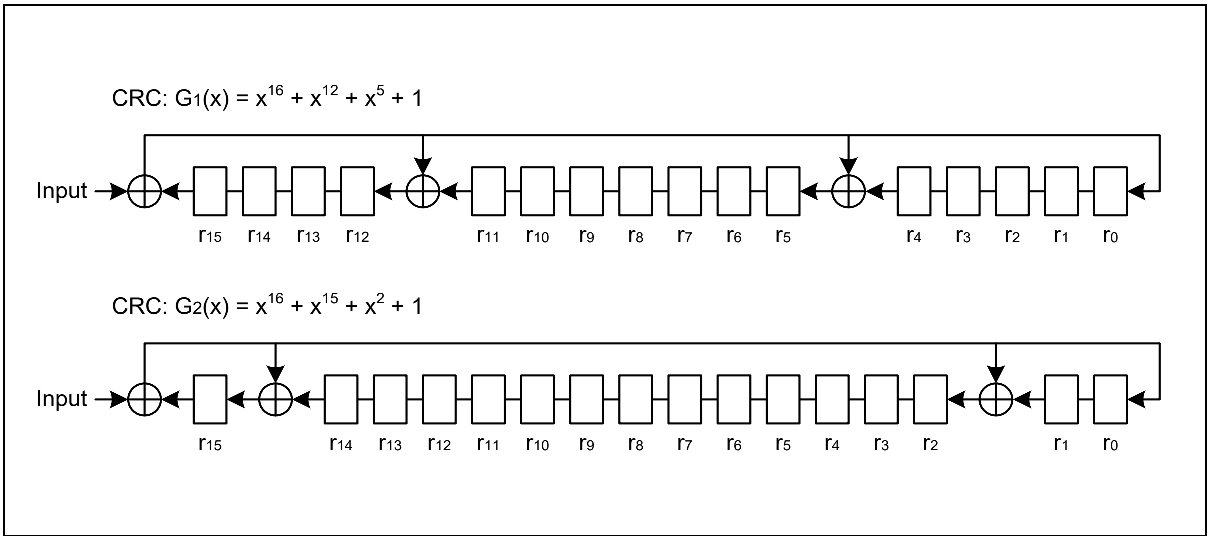
NOTES:

1. The operation is finished after calculation from CRC_SADR to the CRC_EADR on auto mode or SARINC=1.
2. The CRC_SADR and CRC_EADR are the same value after completion.
3. The end address (CRC_EADR) must be at least 4 greater than the start address (CRC_SADR).
4. The end address must be at least 0x7C from the start address on “Auto mode”.
5. The CRC_SADR/CRC_EADR on the “Auto mode” should have any value in flash memory area (0x10000000 to 0x1001FFFF).
6. The CPU will be held on the “Auto Mode” if the CPU is in the flash memory and the global interrupt should be disabled by software.
7. The HCLK frequency should be less than or equal to 20MHz on the “Auto mode” by the CRC_CR.MODS bit.
8. On auto mode run, the CRC_CR register can’t be written and the CRC_RLT register can’t be read.

17.4 CRC and Checksum Functional Description

17.4.1 CRC Polynomial Structure

Figure 84. CRC Polynomial Structure



17.4.2 The CRC Operation Procedure in Auto CRC/Checksum Mode

1. CRC/Checksum Clock Enable
2. Set CRC start address register. (CRC_SADR)
3. Set CRC end address register. (CRC_EADR)
4. Set CRC initial data register. (CRC_INIT)
5. Global interrupt Disable.
6. Select CRC (HCLK) Clock. (HCLK should be less than or equal to 20 MHz during CRC/Checksum auto mode)
7. Select Auto CRC/Checksum Mode and CRC.
8. CRC operation starts. (CRCRUN = 1)
9. Read the CRC result.
10. Global interrupt Enable.

17.4.3 The CRC Operation Procedure in User CRC/Checksum Mode

1. CRC/Checksum Clock Enable
2. Set CRC start address register. (CRC_SADR)
3. Set CRC end address register. (CRC_EADR)
4. Set CRC initial data register. (CRC_INIT)
5. Select User CRC/Checksum Mode and CRC
6. CRC operation starts. (CRCRUN = 1)
7. Input CRC Data at CRC_IN.
8. Check CRC is finished on Start Address Auto Increment or Compare Start address and End address to check CRC end point.
9. Repeat 8 and 9 until CRC end point.
10. CRC Stop and read CRC result.

17.5 CRC and Checksum Registers

The base address and register map of the CRC and checksum block are described in the following tables.

Table 87. Base Address of CRC and Checksum

Name	Base Address
CRC	0x3000_1000

Table 88. CRC and Checksum Register Map

Name	Offset	Type	Description	Reset Value	Reference
CRC_CR	0x0000	RW	CRC/Checksum Control Register	0x0000_0000	17.5.1
CRC_IN	0x0004	RW	CRC/Checksum Input Data Register	0x0000_0000	17.5.2
CRC_RLT	0x0008	RO	CRC/Checksum Result Data Register	0x0000_FFFF	17.5.3
CRC_INIT	0x000C	RW	CRC/Checksum Initial Data Register	0x0000_0000	17.5.4
CRC_SADR	0x0010	RW	CRC/Checksum Start Address Register	0x1000_0000	17.5.5
CRC_EADR	0x0014	RW	CRC/Checksum End Address Register	0x1001_FFFC	17.5.6

17.5.1 CRC_CR: CRC Control Register

The CRC_CR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_CR=0x3000_1000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			MODS	RLTCLR	MDSEL	POLYS	SARINC	Reserved	FIRSTBS	CRCRUN
-			0	0	0	0	0	-	0	0
-			RW	RW	RW	RW	RW	-	RW	RW

7	MODS	User/Auto Mode Selection.	
		0	User mode. (Calculate every data written to the CRC_IN register)
		1	Auto mode. (Calculate till CRC_SADR == CRC_EADR)
6	RLTCLR	CRC/Checksum Result Data Register (CRC_RLT) Initialization.	
		0	No effect.
		1	Initialize the CRC_RLT register with the value of CRC_INIT (This bit is automatically cleared to '0' after operation)
5	MDSEL	CRC/Checksum Selection.	
		0	Select CRC.
		1	Select checksum.
4	POLYS	Polynomial Selection. (CRC only)	
		0	Select CRC-CCITT ($G_1(x) = x^{16} + x^{12} + x^5 + 1$)
		1	Select CRC-16 ($G_2(x) = x^{16} + x^{15} + x^2 + 1$)
3	SARINC	CRC/Checksum Start Address Auto Increment Control. (User mode only)	
		0	No effect.
		1	The CRC/Checksum start address register is incremented by four every data written to the CRC_IN register.
1	FIRSTBS	First Shifted-in Selection. (CRC only)	
		0	MSB-first
		1	LSB-first
0	CRCRUN	CRC/Checksum Start Control and Busy.	
		0	Not busy. The CRC operation can be finished by writing '0' to this bit while running.
		1	Start CRC operation. This bit is automatically cleared to '0' when the value of CRC_SADR register reaches the value of CRC_EADR register.
		NOTE: 1. The five "NOP instruction" should be executed immediately after this bit becomes '1'.	

NOTES:

1. The CRC_RLT register and the CRC/Checksum block should be initialized by writing '1' to the RLTCLR bit before a new CRC/Checksum calculation.
2. The CRCRUN bit should be set to '1' last time after setting appropriate values to the registers.
3. On the user mode, it will be calculated every writing data to the CRC_IN register during CRCRUN==1.
4. On the user mode with SARINC==0, the block is finished by writing '0' to the CRCRUN bit.

5. It is prohibited to write any data to the CRC_IN register during CRCRUN==0.
6. The checksum is calculated by byte unit.
   Ex) On 0x34A7E991, CRC_RLT = 0x34 + 0xA7 + 0xE9 + 0x91.
7. The five “NOP Instruction” should follow immediately after CRCRUN bit is set to ‘1’.

17.5.2 CRC_IN: CRC Input Data Register

The CRC_IN register is a 32-bit size.

CRC_IN=0x3000_1004

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
INDATA[31:0]			
0x00000000			
RW			

31 0	INDATA[31:0]	CRC Input Data.

NOTE:

1. The CRC_IN register should be written in one word (32-bit).

17.5.3 CRC_RLT: CRC Result Data Register

The CRC_RLT register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_RLT=0x3000_1008

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		RLTDATA[15:0]	
-		0xFFFF	
-		RO	

15 0	RLTDATA[15:0]	CRC Result Data.

17.5.4 CRC_INIT: CRC Initial Data Register

The CRC_INIT register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_INIT=0x3000_100C

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">INIDATA[15:0]</td></tr>
<tr><td colspan="16">-</td><td colspan="16">0x0000</td></tr>
<tr><td colspan="16">-</td><td colspan="16">RW</td></tr>
</table>

15 0	INIDATA[15:0]	CRC Initial Data.

17.5.5 CRC_SADR: CRC Start Address Register

The CRC_SADR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_SADR=0x3000_1010

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="30">SADR[31:2]</td><td colspan="2">Don't care</td></tr>
<tr><td colspan="30">0x10000000</td><td colspan="2">00</td></tr>
<tr><td colspan="30">RW</td><td colspan="2">-</td></tr>
</table>

31 2	SADR[31:2]	CRC Start Address.
1 0	-	Don't care.

NOTE:

1. The LSB 7-bit of the start address should be “0x00” on “Auto mode”

17.5.6 CRC_EADR: CRC End Address Register

The CRC_EADR register is a 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_EADR=0x3000_1014

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="30">EADR[31:2]</td><td colspan="2">Don’t care</td></tr>
<tr><td colspan="30">0x1001FFFC</td><td colspan="2">00</td></tr>
<tr><td colspan="30">RW</td><td colspan="2">-</td></tr>
</table>

31 2	EADR[31:2]	CRC End Address.
1 0	-	Don’t care.

NOTE:

1. The LSB-7bits of the end address should be “0x7C” on “Auto mode”.

17.5.7 CRC Register Map Summary

Table 89. CRC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MODS	RLTCLR	MDSEL	POLYS	SARINC	Res	FIRSTBS	CRCRUN
	Reset value																									0	0	0	0	0		0	0
0x04	CRC_IN	INDATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	CRC_RLT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RLTDATA[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	CRC_INIT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	INIDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	CRC_SADR	SADR[31:2]																														Don't care	
	Reset value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	CRC_EADR	EADR[31:2]																														Don't care	
	Reset value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

Abbreviation for Registers

The following abbreviations are used in register descriptions:

Table 90. Abbreviation

Abbreviation	Description
RW (Read / write)	Software can read and write to this bit.
RO (Read only)	Software can only read this bit.
WO (Write-only)	Software can only write to this bit. Reading this bit returns the reset value.
RC (Read / clear)	Software can read to clear this bit.
RWC0 (Read / write 0 clear)	Software can read and clear this bit by writing ‘0’.
RWC1 (Read / write 1 clear)	Software can read and clear this bit by writing ‘1’.
Res. (Reserved)	Reserved bit, must be kept at reset value.

Glossary

This section gives a brief definition of acronyms, abbreviations, and terminology used in this document:

- Word: Data of 32-bit length
- Half-word: Data of 16-bit length
- Byte: Data of 8-bit length
- PGM: Flash Memory Programming
- ERS: Flash Memory Erasing
- AHB: Advanced High-performance Bus
- APB: Advanced Peripheral Bus
- CRC: Cyclic Redundancy Check
- DMA: Direct Memory Access
- FRT: Free-Run Timer
- HSE: High-Speed External Oscillator
- HSI: High-Speed Internal Oscillator
- I2C: Inter-Integrated Circuit
- LSB: Least Significant Bit
- LQFP: Low-profile Quad Flat Package
- LSE: Low-Speed External Oscillator
- LSI: Low-Speed Internal Oscillator
- LVI: Low-Voltage Indicator
- LVR: Low-Voltage Reset
- MPWM: Motor Pulse-Width Modulation
- MSB: Most Significant Bit
- OPAMP: Operational Amplifier
- PGM: Programming
- PLL: Phase-Locked Loop
- POR: Power-On Reset
- SCU: System Control Unit
- SPI: Serial Peripheral Interface
- UART: Universal Asynchronous Receiver Transmitter
- WDT: Watchdog Timer

Revision History

Revision	Date	Notes
1.00	Apr. 26, 2024	Initial release.
1.01	July. 30, 2024	Minor revision.

Korea

Regional Office, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
www.abovsemi.com

HQ, Ochang
R&D, QA, and Test Center
37, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
www.abovsemi.com

Domestic Sales Manager
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

Global Sales Manager
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

China Sales Manager
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr

ABOV Disclaimer

IMPORTANT NOTICE – PLEASE READ CAREFULLY

ABOV Semiconductor ("ABOV") reserves the right to make changes, corrections, enhancements, modifications, and improvements to ABOV products and/or to this document at any time without notice. ABOV does not give warranties as to the accuracy or completeness of the information included herein. Purchasers should obtain the latest relevant information of ABOV products before placing orders. Purchasers are entirely responsible for the choice, selection, and use of ABOV products and ABOV assumes no liability for application assistance or the design of purchasers' products. No license, express or implied, to any intellectual property rights is granted by ABOV herein. ABOV disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of ABOV products in such unauthorized applications. ABOV and the ABOV logo are trademarks of ABOV. All other product or service names are the property of their respective owners. Information in this document supersedes and replaces the information previously supplied in any former versions of this document.