A31G213, A31G212
User's Manual

32-bit Cortex-M0+ based
General Purose Microcontroller
With Touch & LED Driver

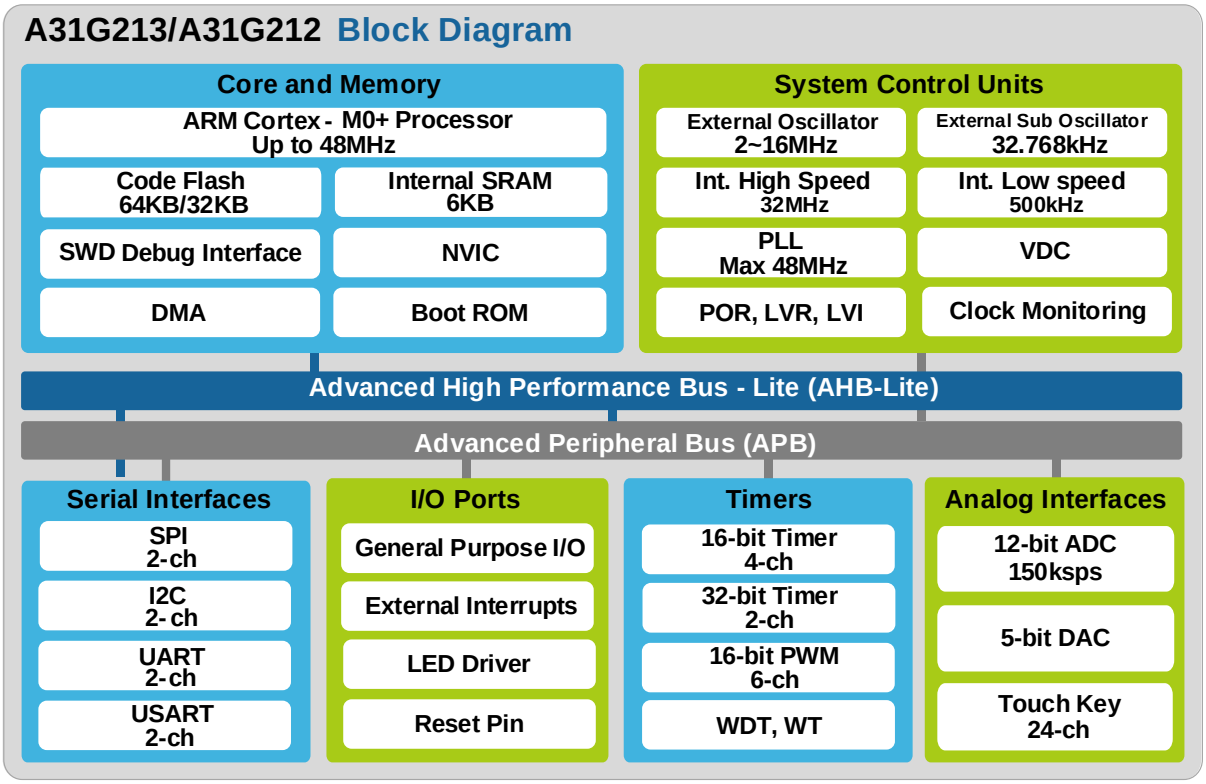
User's Manual Version 1.10

Introduction

This user's manual contains complete information for application developers who use A31G213 or A31G212 for their specific needs.

A31G21x devices are 32-bit general purpose microcontrollers for various appliances. To meet the requirements for the complexity and high performance in consumer electronics, A31G21x incorporate ARM's high-speed 32-bit Cortex-M0+ Cores.

Building on the very successful Cortex-M0+ processor, the ARM® Cortex®-M0+ retains full instruction set and tool compatibility, while further reducing energy consumption and increasing performance.



Reference document

- Document 'DDI 0484C' is provided by ARM and contains information of Cortex-M+0.

Contents

List of figures

List of tables

1 Description

A31G21x lines are 32-bit general purpose microcontrollers with up to 64 Kbytes of flash memory.

In this section, features of A31G21x and peripheral counts are introduced.

Table 1. A31G213 and A31G212 Device Features and Peripheral Counts

Classification	Module/ Peripherals	Description
Core	CPU	• ARM Cortex-M0+ Core Maximum operating frequency: up to 48MHz
	Interrupt	• NVIC (Nested-Vectored Interrupt Controller) • Up to 32 peripheral interrupts supported
Memory	Code Flash	• A31G213 64KB Code Flash Memory • A31G212 32KB Code Flash Memory
	SRAM	• 6KB SRAM
System Control Units (SCU)	Operating Frequency	• Up to 48 MHz
	Clock	• High Speed Internal oscillator (HSI) : 32MHz, • Low Speed Internal oscillator (LSI) : 500KHz • External oscillator(HSE) : 2MHz ~16MHz • External sub-oscillator (LSE) : 32.768 KHz • Phase-locked loop (PLL) frequency generator Generates a high-speed clock (up to 48 MHz)
	Clock Monitoring	• System Fail-Safe function by Clock Monitoring - External oscillator(HSE) - External sub-oscillator (LSE) - Main system clock (MCLK)
	Operating Mode	• RUN mode • SLEEP mode • Power-down mode
	Reset	• nRESET pin reset • Core reset • Software reset • POR (Power-on Reset) • LVR (Low Voltage-Reset) • Reset due to clock oscillating error

Table 1. A31G213 and A31G212 Device Features and Peripheral Counts (continued)

Classification	Module/ Peripherals	Description
Timer	TIMER1x	• 16-bit : 4-ch - Periodic timer mode, One-shot timer mode, - PWM pulse mode, Capture mode
	Timer2x	• 32-bit : 2-ch - Periodic timer mode, One-shot timer mode, - PWM pulse mode, Capture mode
PWM	Timer3x	• 16-bit : 6-ch - Periodic timer mode, Back-to-Back mode, Capture mode
Communication Function	USART	• 48-pin, 44-Pin, 32-pin : 2-ch • 28-pin : 1-ch
	UART	• 2 channels
	SPI	• 48-pin, 44-pin : 2 channels • 32-pin, 28-pin : 1 channel
	I2C	• 2 channels
A/D converter		•12-bit ADC : 150Ksps - 48-pin : 14-ch - 44-pin : 12-ch - 32-pin : 8-ch - 28-pin : 6-ch
D/A converter		• 1 channel • 5-bit resolution
Capacitive Touch Switch		• Capacitive Touch Switch - 48-Pin : 24-ch - 44-Pin : 21-ch - 32-Pin : 13-ch - 28-Pin : 11-ch
LED Driver		• 10 SEG x 16 COM LED Driver
CRC calculator (CRC)		• CRC-CCITT, CRC-16
Operating Voltage		• 1.8V to 5.5V
Operating temperature		• Commercial grade(-40℃ ~ +85℃) • Industrial grade(-40℃ ~ +105℃)

Table 1. A31G213 and A31G212 Device Features and Peripheral Counts (continued)

Classification	Module/Peripherals	Description
Package		• Four types of package options - 48-pin LQFP (7 x 7, 0.5mm pitch) - 44-pin MQFP (10 x 10, 0.8mm pitch) - 32-pin LQFP (7 x 7, 0.8mm pitch) - 32 pin QFN (5 x 5 , 0.5mm pitch) - 28-pin TSSOP (9.7x 4.4, 0.65mm pitch)

1.1 Block diagram

In this section, A31G21x devices with peripherals are described in block diagram.

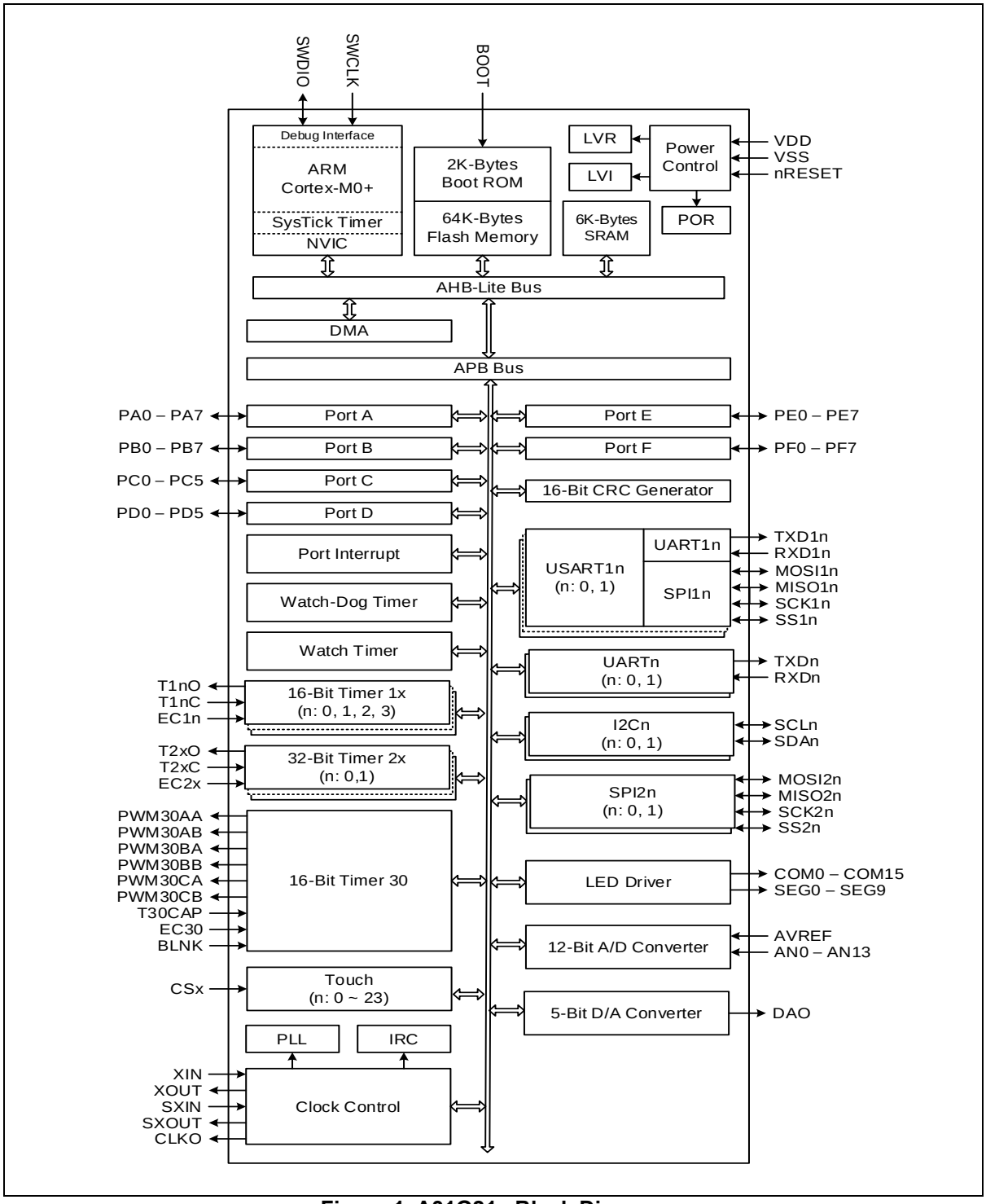


Figure 1. A31G21x Black Diagram

2 Pinouts and pin descriptions

In this chapter, A31G21x devices pinouts and pin descriptions are introduced.

2.1 Pinouts

2.1.1 A31G212CLN, A31G213CLN, A31G212CL2N, A31G213CL2N (48-LQFP)

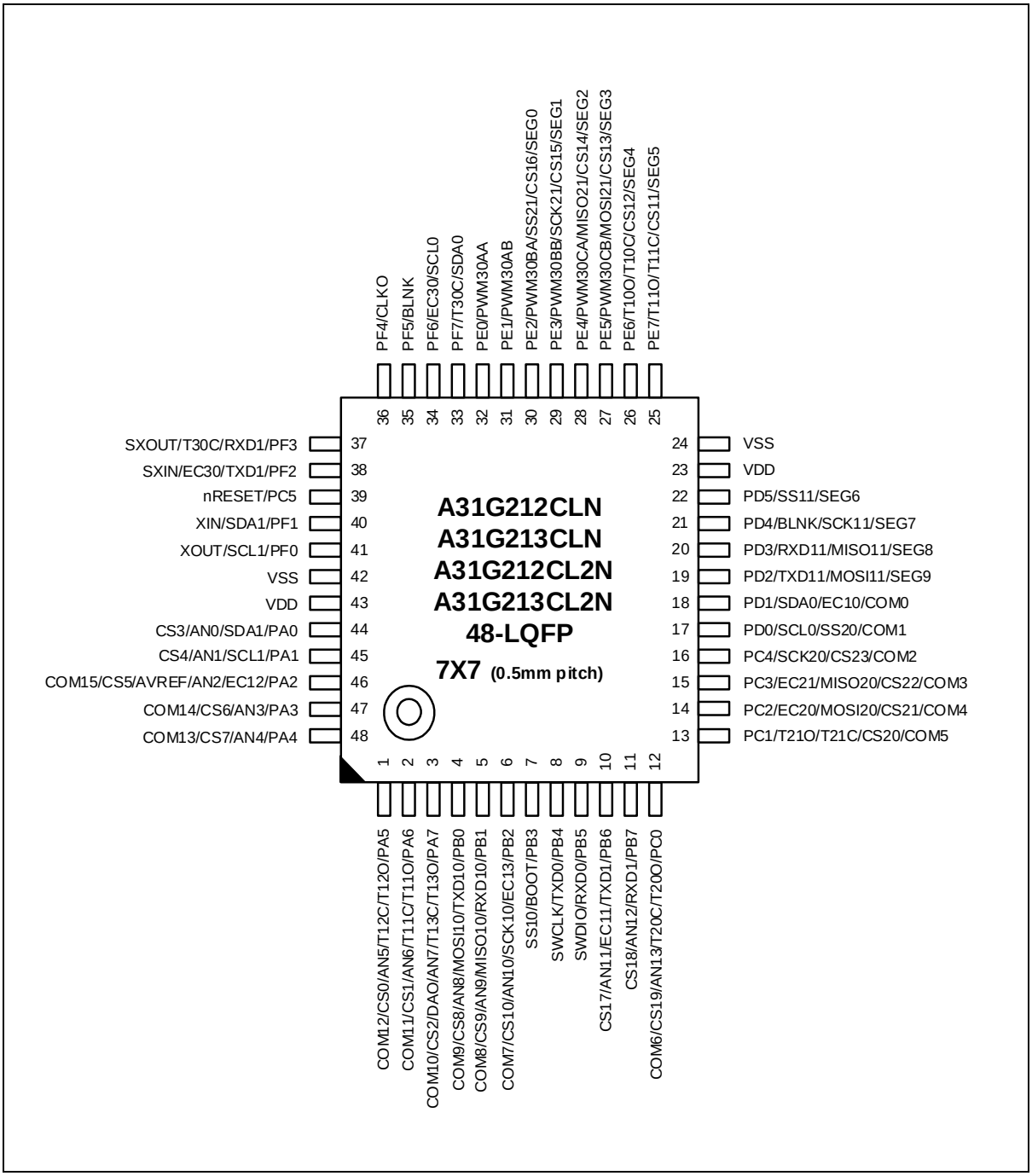


Figure 2. LQFP 48 Pinouts

2.1.2 A31G212SQN, A31G213SQN, A31G212SQ2N, A31G213SQ2N (44-MQFP)

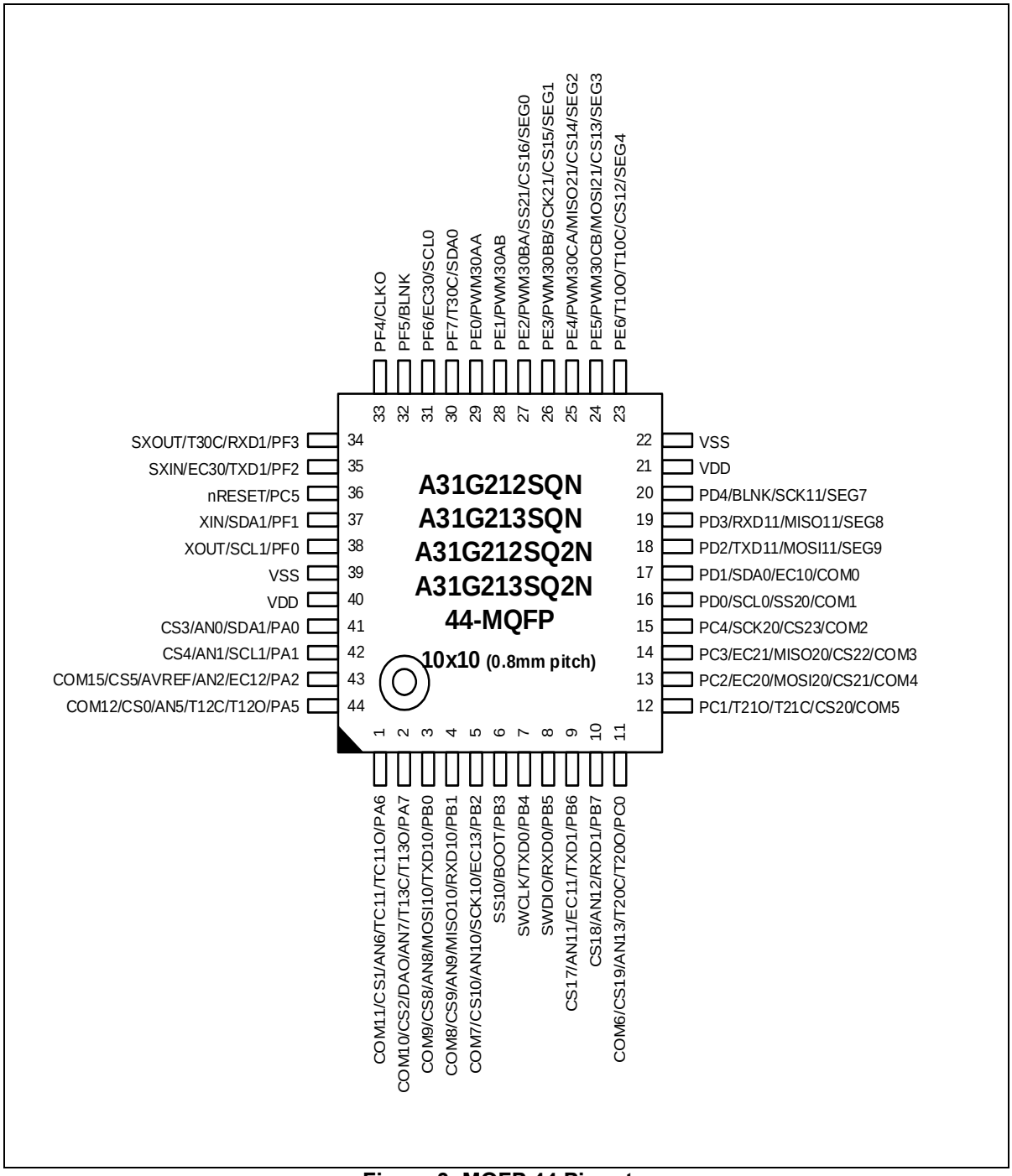


Figure 3. MQFP 44 Pinouts

A31G212KNN, A31G213KNN, A31G212KN2N, A31G213KN2N (32-LQFP)

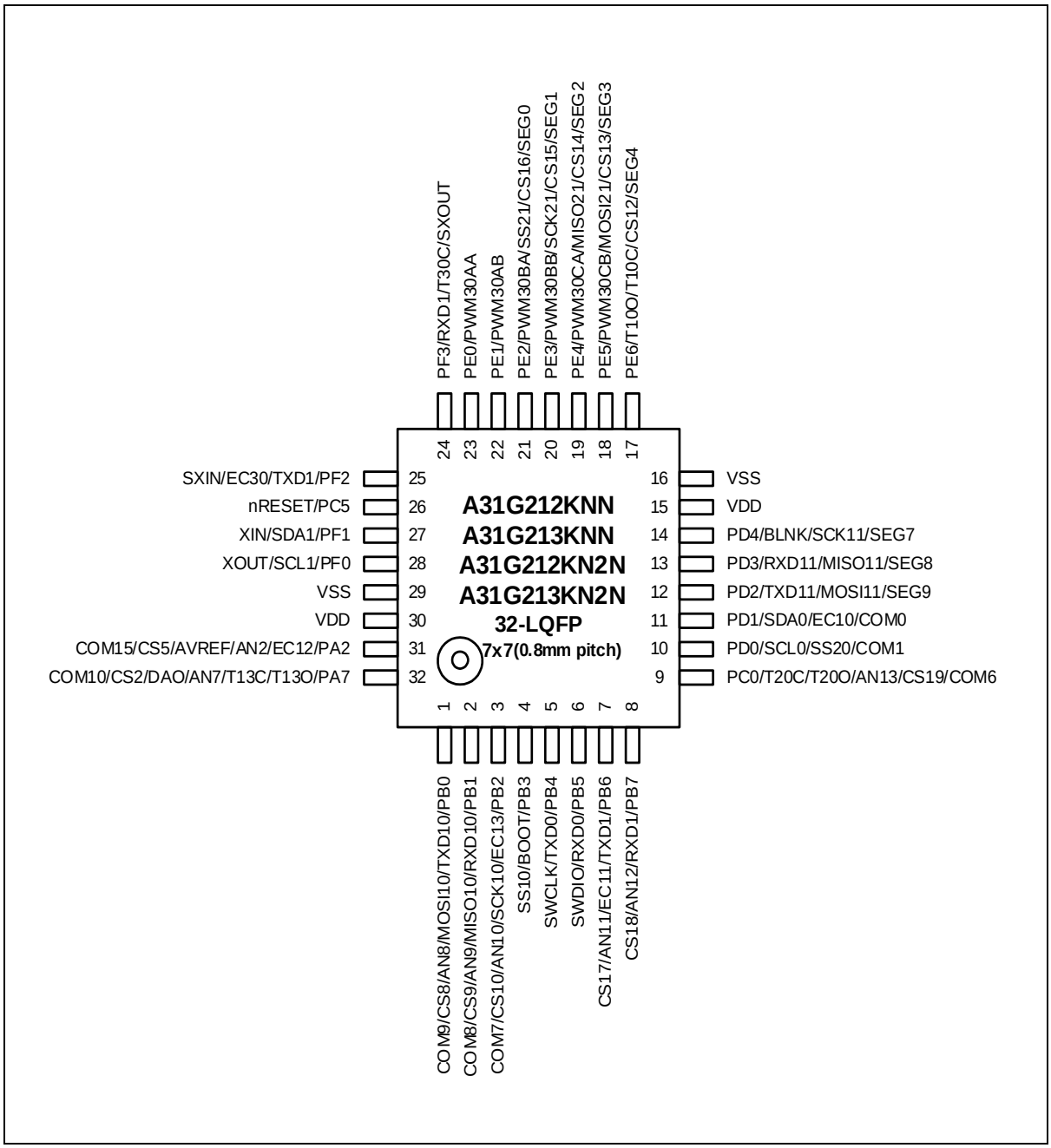


Figure 4. LQFP 32 Pinouts

2.1.3 A31G212KUN, A31G213KUN, A31G212KU2N, A31G213KU2N (32-QFN)

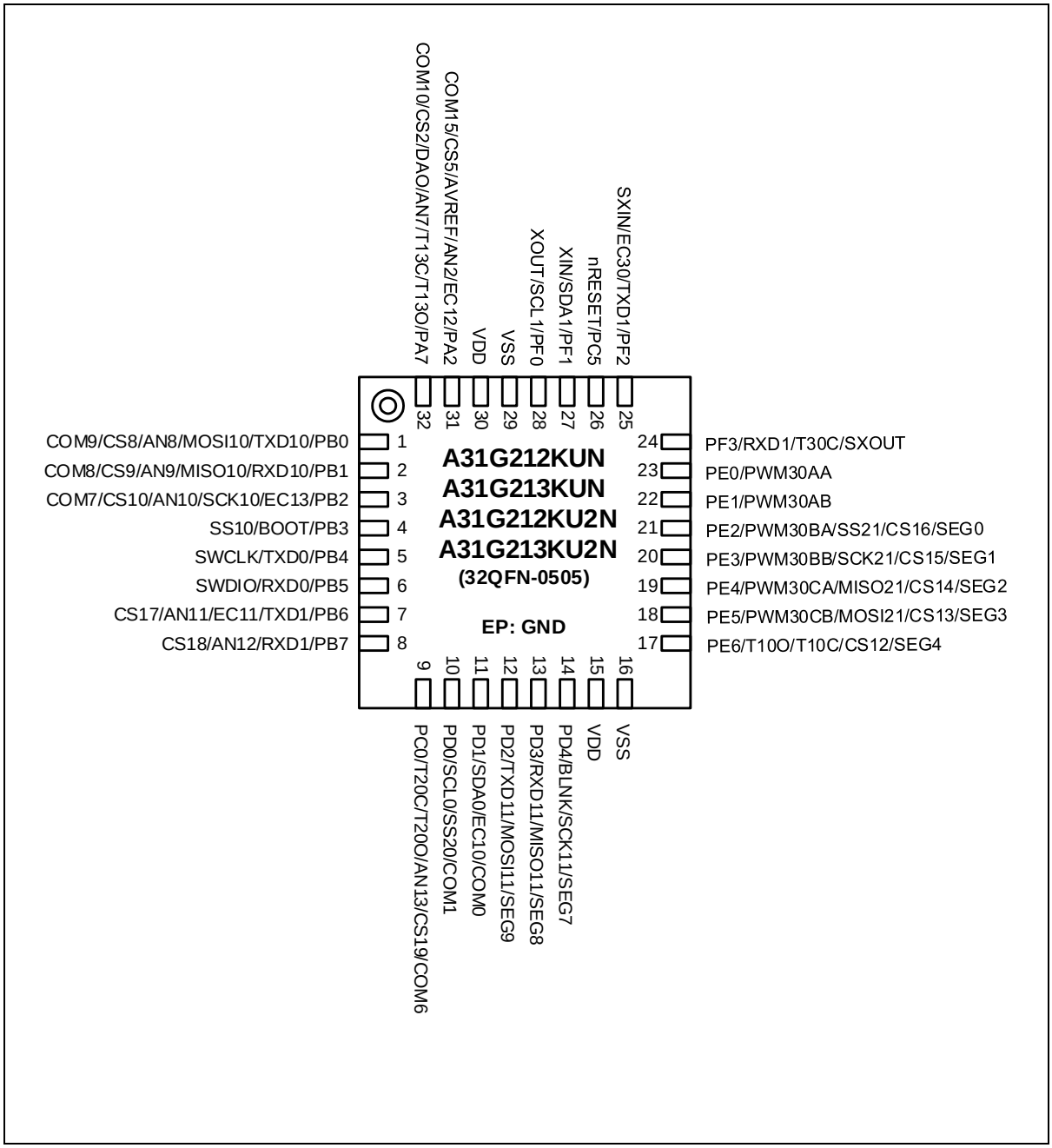


Figure 5. QFN 32 Pinouts

2.1.4 A31G212GRN, A31G213GRN, A31G212GR2N, A31G213GR2N (28-TSSOP)

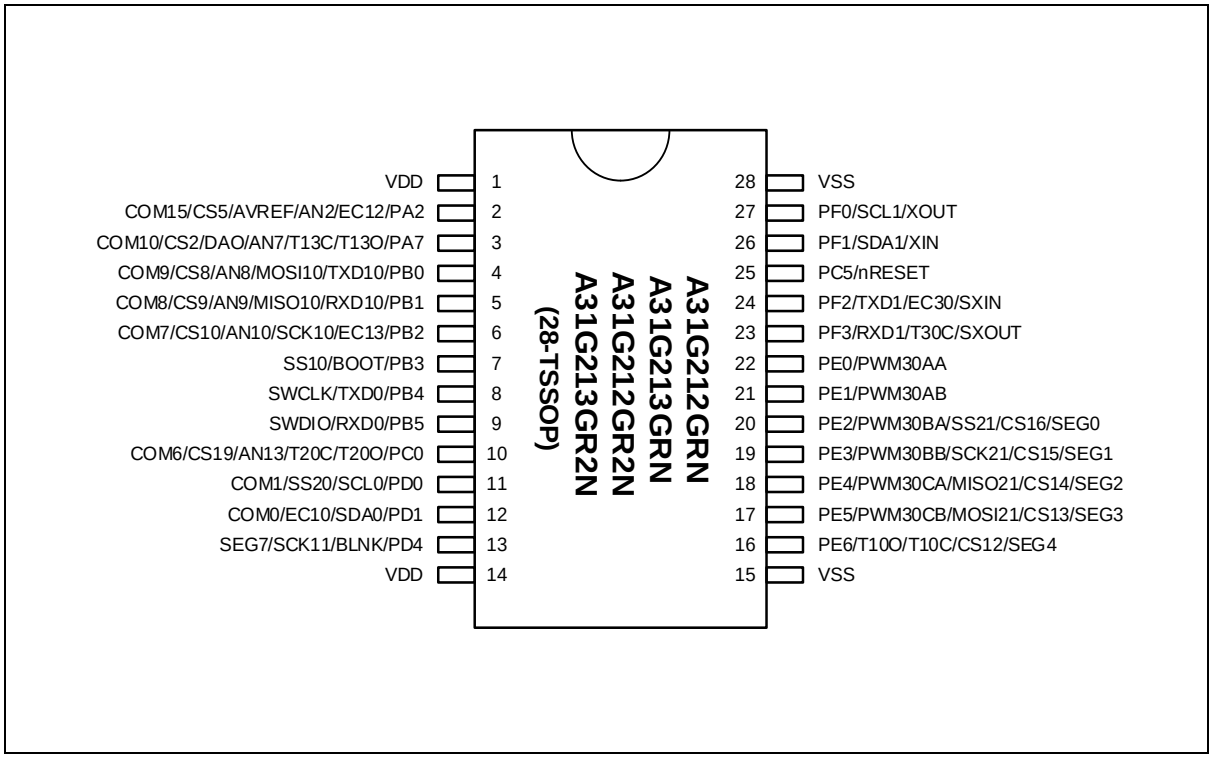


Figure 6. TSSOP 28 Pinouts

2.2 Pin description

Pin configuration information in Table 2 contains two pairs of power/ground and other dedicated pins. These multi-function pins have up to ten selections of functions including GPIO. Configuration including pin ordering can be changed without notice.

Table 2. Pin Description

Pin No				Pin Name	Type	Description	Remark
48	44	32	28				
1	44	-	-	PA5*	IOUDS	PORT A Bit 5 Input/Output	
				T12O	O	Timer 12 Output	
				T12C	I	Timer 12 Clock/Capture Input	
				AN5	IA	Analog Input 5	
				CS0	IA	Capacitive Touch switch input 0	
				COM12	O	LED Common Signal 12 Output	
2	1	-	-	PA6*	IOUDS	PORT A Bit 6 Input/Output	
				T11O	O	Timer 11 Output	
				T11C	I	Timer 11 Clock/Capture Input	
				AN6	IA	Analog Input 6	
				CS1	IA	Capacitive Touch switch input 1	
				COM11	O	LED Common Signal 11 Output	
3	2	32	3	PA7*	IOUDS	PORT A Bit 7 Input/Output	
				T13O	O	Timer 13 Output	
				T13C	I	Timer 13 Clock/Capture Input	
				AN7	IA	Analog Input 7	
				DAO	OA	Digital to analog output	
				CS2	IA	Capacitive Touch switch input 2	
				COM10	O	LED Common Signal 10 Output	
4	3	1	4	PB0	IOUDS	PORT B Bit 0 Input/Output	
				TXD10*	O	UART Channel 10 TxD Input	
				MOSI10	I/O	SPI Channel 10 Master Out / Slave In	
				AN8	IA	Analog Input 8	
				CS8	IA	Capacitive Touch switch input 8	
				COM9	O	LED Common Signal 9 Output	
5	4	2	5	PB1	IOUDS	PORT B Bit 1 Input/Output	
				RXD10*	I	UART Channel 10 RxD Input	
				MISO10	I/O	SPI10 Master-Input/Slave-Output Data signal	
				AN9	IA	Analog Input 9	
				CS9	IA	Capacitive Touch switch input 9	
				COM8	O	LED Common Signal 8 Output	
6	5	3	6	PB2*	IOUDS	PORT B Bit 2 Input/Output	
				EC13	I	Timer 13 Event Count Input	
				SCK10	I/O	SPI10 Data Clock Input/Output	
				AN10	IA	Analog Input 10	
				CS10	IA	Capacitive Touch switch input 10	
				COM7	O	LED Common Signal 7 Output	
7	6	4	7	PB3	IOUDS	PORT B Bit 3 Input/Output	
				BOOT*	I	Boot mode Selection Input	Pull-up
				SS10	I/O	SPI Channel 10 Slave Select signal	

Table 2. Pin Description (continued)

Pin number				Pin name	Type	Description	Remark
48	44	32	28				
8	7	5	8	PB4	IOUDS	PORT B Bit 4 Input/Output	
				TXD0	O	UART Channel 0 TxD Input	
				SWCLK*	I	SWD Clock Input	Pull-up
9	8	6	9	PB5	IOUDS	PORT B Bit 5 Input/Output	
				RXD0	I	UART Channel 0 RxD Input	
				SWDIO*	I/O	SWD Data Input/Output	Pull-up
10	9	7	-	PB6*	IOUDS	PORT B Bit 6 Input/Output	
				TXD1	O	UART Channel 1 TxD Input	
				EC11	I	Timer 11 Event Count Input	
				AN11	IA	Analog Input 11	
				CS17	IA	Capacitive Touch switch input 17	
11	10	8	-	PB7*	IOUDS	PORT B Bit 7 Input/Output	
				RXD1	I	UART Channel 1 RxD Input	
				AN12	IA	Analog Input 12	
				CS18	IA	Capacitive Touch switch input 18	
12	11	9	10	PC0*	IOUDS	PORT C Bit 0 Input/Output	
				T20O	O	Timer 20 Output	
				T20C	I	Timer 20 Clock/Capture Input	
				AN13	IA	Analog Input 13	
				CS19	IA	Capacitive Touch switch input 19	
				COM6	O	LED Common Signal 6 Output	
13	12	-	-	PC1*	IOUDS	PORT C Bit 1 Input/Output	
				T21O	O	Timer 21 Output	
				T21C	I	Timer 21 Clock/Capture Input	
				CS20	IA	Capacitive Touch switch input 20	
				COM5	O	LED Common Signal 5 Output	
14	13	-	-	PC2*	IOUDS	PORT C Bit 2 Input/Output	
				EC20	I	Timer 20 Event Count Input	
				MOSI20	I/O	SPI Channel 20 Master Out / Slave In	
				CS21	IA	Capacitive Touch switch input 21	
				COM4	O	LED Common Signal 4 Output	
15	14	-	-	PC3*	IOUDS	PORT C Bit 3 Input/Output	
				EC21	I	Timer 21 Event Count Input	
				MISO20	I/O	SPI20 Master-Input/Slave-Output Data signal	
				CS22	IA	Capacitive Touch switch input 22	
				COM3	O	LED Common Signal 3 Output	
16	15	-	-	PC4*	IOUDS	PORT C Bit 4 Input/Output	
				SCK20	I/O	SPI20 Data Clock Input/Output	
				CS23	IA	Capacitive Touch switch input 23	
				COM2	O	LED Common Signal 2 Output	
17	16	10	11	PD0*	IOUDS	PORT D Bit 0 Input/Output	
				SCL0	O	I2C Channel 0 SCL In/Out	
				SS20	I/O	SPI Channel 20 Slave Select signal	
				COM1	O	LED Common Signal 1 Output	

Table 2. Pin Description (continued)

Pin number				Pin name	Type	Description	Remark
48	44	32	28				
18	17	11	12	PD1*	IOUDS	PORT D Bit 1 Input/Output	
				SDA0	O	I2C Channel 0 SDA In/Out	
				EC10	I	Timer 10 Event Count Input	
				COM0	O	LED Common Signal 0 Output	
19	18	12	-	PD2*	IOUDS	PORT D Bit 2 Input/Output	
				TXD11	O	UART Channel 11 TxD Input	
				MOSI11	I/O	SPI Channel 11 Master Out / Slave In	
				SEG9	O	LED Segment Signal 9 Output	
20	19	13	-	PD3*	IOUDS	PORT D Bit 3 Input/Output	
				RXD11	I	UART Channel 11 RxD Input	
				MISO11	I/O	SPI11 Master-Input/Slave-Output Data signal	
				SEG8	O	LED Segment Signal 8 Output	
21	20	14	13	PD4*	IOUDS	PORT D Bit 4 Input/Output	
				BLNK	I	External Sync Signal Input for T30 PWM	
				SCK11	I/O	SPI11 Data Clock Input/Output	
				SEG7	O	LED Segment Signal 7 Output	
22	-	-	-	PD5*	IOUDS	PORT D Bit 5 Input/Output	
				SS11	I/O	SPI Channel 11 Slave Select signal	
				SEG6	O	LED Segment Signal 6 Output	
23	21	15	14	VDD	P	VDD	
24	22	16	15	VSS	P	VSS	
25			-	PE7*	IOUDS	PORT E Bit 7 Input/Output	
				T11O	O	Timer 11 Output	
				T11C	I	Timer 11 Clock/Capture Input	
				CS11	IA	Capacitive Touch switch input 11	
				SEG5	O	LED Segment Signal 5 Output	
26	23	17	16	PE6*	IOUDS	PORT E Bit 6 Input/Output	
				T10O	O	Timer 10 Output	
				T10C	I	Timer 10 Clock/Capture Input	
				CS12	IA	Capacitive Touch switch input 12	
				SEG4	O	LED Segment Signal 4 Output	
27	24	18	17	PE5*	IOUDS	PORT E Bit 5 Input/Output	
				PWM30CB	O	Timer 30 PWM Output	
				MOSI21	I/O	SPI Channel 21 Master Out / Slave In	
				CS13	IA	Capacitive Touch switch input 13	
				SEG3	O	LED Segment Signal 3 Output	
28	25	19	18	PE4*	IOUDS	PORT E Bit 4 Input/Output	
				PWM30CA	O	Timer 30 PWM Output	

Table 2. Pin Description (continued)

Pin number				Pin name	Type	Description	Remark
48	44	32	28				
				MISO21	I/O	SPI21 Master-Input/Slave-Output Data signal	
				CS14	IA	Capacitive Touch switch input 14	
				SEG2	O	LED Segment Signal 2 Output	
29	26	20	19	PE3*	IOUDS	PORT E Bit 3 Input/Output	
				PWM30BB	O	Timer 30 PWM Output	
				SCK21	I/O	SPI21 Data Clock Input/Output	
				CS15	IA	Capacitive Touch switch input 15	
				SEG1	O	LED Segment Signal 1 Output	
30	27	21	20	PE2*	IOUDS	PORT E Bit 2 Input/Output	
				PWM30BA	O	Timer 30 PWM Output	
				SS21	I/O	SPI Channel 21 Slave Select signal	
				CS16	IA	Capacitive Touch switch input 16	
				SEG0	O	LED Segment Signal 0 Output	
31	28	22	21	PE1*	IOUDS	PORT E Bit 1 Input/Output	
				PWM30AB	O	Timer 30 PWM Output	
32	29	23	22	PE0*	IOUDS	PORT E Bit 0 Input/Output	
				PWM30AA	O	Timer 30 PWM Output	
33	30	-	-	PF7*	IODS	PORT F Bit 7 Input/Output	Open-drain
				T30C	I	Timer 30 Clock/Capture Input	
				SDA0	O	I2C Channel 0 SDA In/Out	
34	31	-	-	PF6*	IODS	PORT F Bit 6 Input/Output	Open-drain
				EC30	I	Timer 30 Event Count Input	
				SCL0	O	I2C Channel 0 SCL In/Out	
35	32	-	-	PF5*	IODS	PORT F Bit 5 Input/Output	Open-drain
				BLNK	I	External Sync Signal Input for T30 PWM	
36	33	-	-	PF4*	IOUDS	PORT F Bit 4 Input/Output	
				CLKO	O	System Clock Output	
37	34	24	23	PF3*	IOUDS	PORT F Bit 3 Input/Output	
				RXD1	I	UART Channel 1 RxD Input	
				T30C	I	Timer 30 Clock/Capture Input	
				SXOUT	O	Sub Oscillator Output	
38	35	25	24	PF2*	IOUDS	PORT F Bit 2 Input/Output	
				TXD1	O	UART Channel 1 TxD Input	
				EC30	I	Timer 30 Event Count Input	
				SXIN	I	Sub Oscillator Input	
39	36	26	25	PC5	IOUDS	PORT C Bit 5 Input/Output	
				nRESET*	IU	External Reset Input	Pull-up
40	37	27	26	PF1*	IOUDS	PORT F Bit 1 Input/Output	

Table 2. Pin Description (continued)

Pin number				Pin name	Type	Description	Remark
48	44	32	28				
				SDA1	O	I2C Channel 1 SDA In/Out	
				XIN	I	Main Oscillator Input	
41	38	28	27	PF0*	IOUDS	PORT F Bit 0 Input/Output	
				SCL1	O	I2C Channel 1 SCL In/Out	
				XOUT	O	Main Oscillator Output	
42	39	29	28	VSS	P	GND	
43	40	30	1	VDD	P	VDD	
44	41	-	-	PA0*	IOUDS	PORT A Bit 0 Input/Output	
				SDA1	O	I2C Channel 1 SDA In/Out	
				AN0	IA	Analog Input 0	
				CS3	IA	Capacitive Touch switch input 3	
45	42	-	-	PA1*	IOUDS	PORT A Bit 1 Input/Output	
				SCL1	O	I2C Channel 1 SCL In/Out	
				AN1	IA	Analog Input 1	
				CS4	IA	Capacitive Touch switch input 4	
46	43	31	2	PA2*	IOUDS	PORT A Bit 2 Input/Output	
				EC12	I	Timer 12 Event Count Input	
				AN2	IA	Analog Input 2	
				AVREF	IA	A/D Converter Reference Input	
				CS5	IA	Capacitive Touch switch input 5	
				COM15	O	LED Common Signal 15 Output	
47	-	-	-	PA3*	IOUDS	PORT A Bit 3 Input/Output	
				AN3	IA	Analog Input 3	
				CS6	IA	Capacitive Touch switch input 6	
				COM14	O	LED Common Signal 14 Output	
48	-	-	-	PA4*	IOUDS	PORT A Bit 4 Input/Output	
				AN4	IA	Analog Input 4	
				CS7	IA	Capacitive Touch switch input 7	
				COM13	O	LED Common Signal 13 Output	

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. The * means 'selected pin function after reset condition'.
3. Pin order may be changed with revision notice.

3 System and memory overview

3.1 System architecture

Main system of A31G21x series consists of the followings:

- ARM® Cortex® -M0+ core
- General purpose DMA
- Internal SRAM
- Internal Flash memory
- Two AHB buses

3.1.1 Cortex-M0+ core

The ARM® Cortex®-M0+ processor is the most energy-efficient ARM processor available. It builds on the very successful Cortex-M0+ processor, retaining full instruction set and tool compatibility, while further reducing energy consumption and increasing performance. Document "DDI 0484C" from ARM provides detail information of Cortex-M0+.

3.1.2 Interrupt controller

Table 3. Interrupt Vector Map

Priority	Vector Address	Interrupt Source
-16	0x0000_0000	Stack Pointer
-15	0x0000_0004	Reset Address
-14	0x0000_0008	NMI Handler
-13	0x0000_000C	Hard Fault Handler
-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 Handler
-4	0x0000_0030	Reserved
-3	0x0000_0034	
-2	0x0000_0038	PenSV Handler
-1	0x0000_003C	SysTick Handler
0	0x0000_0040	LVI
1	0x0000_0044	SYSCLKFAIL
2	0x0000_0048	WDT
3	0x0000_004C	GPIOA,B
4	0x0000_0050	GPIOC,D
5	0x0000_0054	GPIOE
6	0x0000_0058	GPIOF
7	0x0000_005C	TIMER10
8	0x0000_0060	TIMER11
9	0x0000_0064	TIMER12
10	0x0000_0068	I2C0
11	0x0000_006C	USART10
12	0x0000_0070	WT
13	0x0000_0074	TIMER30
14	0x0000_0078	I2C1
15	0x0000_007C	TIMER20

Table 3. Interrupt Vector Map (continued)

Priority	Vector Address	Interrupt Source
16	0x0000_0080	TIMER21
17	0x0000_0084	USART11
18	0x0000_0088	ADC
19	0x0000_008C	UART0
20	0x0000_0090	UART1
21	0x0000_0094	TIMER13
22	0x0000_0098	Reserved
23	0x0000_009C	Reserved
24	0x0000_00A0	Reserved
25	0x0000_00A4	SPI20
26	0x0000_00A8	SPI21
27	0x0000_00AC	Reserved
28	0x0000_00B0	LED
29	0x0000_00B4	TOUCH
30	0x0000_00B8	Reserved
31	0x0000_00BC	CRC

NOTES:

1. Each interrupt has an associated priority-level register. Each of them is 2 bits wide, occupying the two MSBs of the Interrupt Priority Level Registers.
   Each Interrupt Priority Level Register occupies 1 byte (8 bits).NVIC registers in the Cortex-M0+ processor can only be accessed using word-size transfers, so for each access, four Interrupt Priority Level Registers are accessed at the same time.

   ** __NVIC_PRIO_BITS = 2

2. Figure 7 is a caution when using Peripheral Interrupts. Interrupt don't work if only peripheral interrupts are enabled. Enable the function in core to enable interrupt.

   * __enable_irq > NVCI_EnableIRQ(Peripheral) > Each Peripheral Interrupt

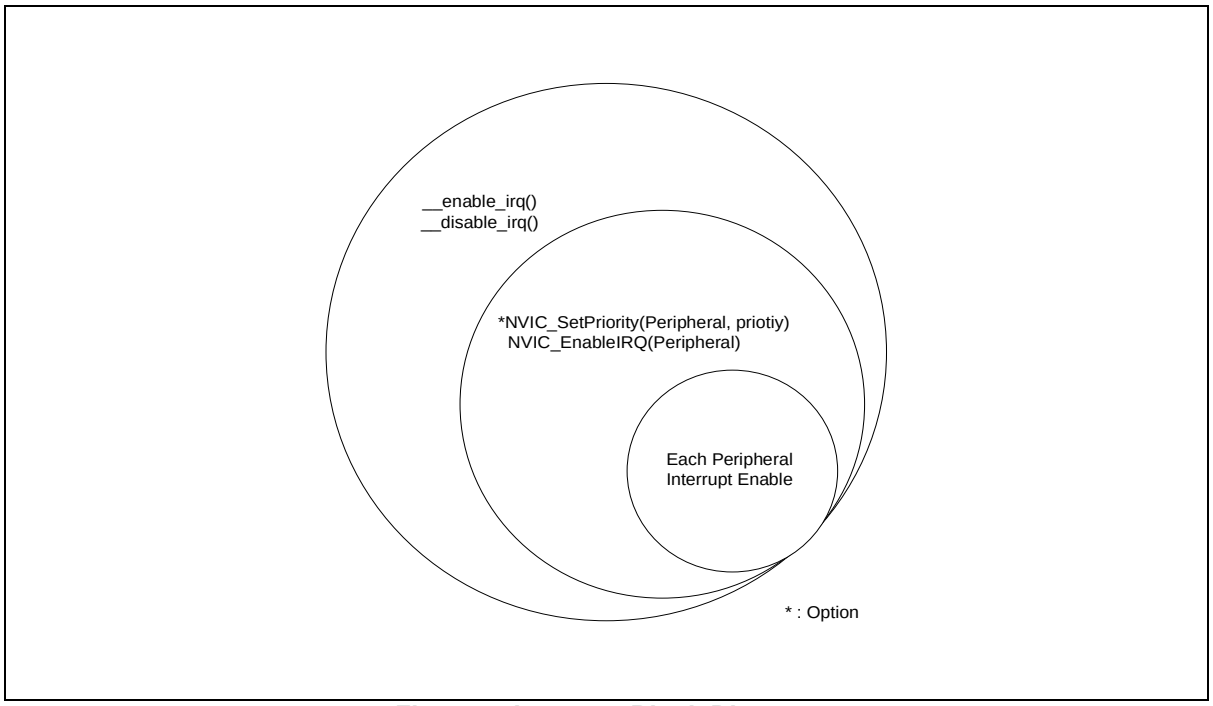


Figure 7. Interrupt Block Diagram

3.2 Memory organization

Program memory, data memory, registers and I/O ports are organized in the same address space.

3.2.1 Register boundary address

Table 4 gives the boundary addresses of peripherals in A31G21x series.

Table 4. A31G21x Memory Boundary Addresses

Boundary address	Memory area	Register description
0x4000_0000	SCU	Section 4.5
0x4000_5100	LVI/LVR	Section 4.5
0x4000_1000/1100/1200/1300/1400/1500	PCU A/B/C/D/F	Section 5.3
0x4000_0100	Flash controller	Section 6.1
0x2000_0000	Internal SRAM	Section 3.2.3
0x4000_0400/0410/0420/0430	DMACH0/1/2/3	Section 7.2
0x4000_1A00	WDT	Section 8.2
0x4000_2000	WT	Section 9.2
0x4000_2100/2200/2300/2700	Timer 10/11/12/13	Section 10.2
0x4000_2500	Timer 20	Section 11.2
0x4000_2600	Timer 21	Section 12.2
0x4000_2400	Timer 30	Section 13.2
0x4000_3800/3900	USART 10/11	Section 14.2
0x4000_4000/4100	UART0/1	Section 15.2
0x4000_4800/4900	I2C 0/1	Section 16.2
0x4000_4C00/4D00	SPI 20/21	Section 17.2
0x4000_3000	12-bit ADC	Section 18.2
0x4000_3500	5-bit DAC	Section 19.2
0x4000_3600	TOUCH	Section 20.2
0x4000_6000	LED	Section 21.2
0x4000_0300	CRC	Section 22.2

3.2.2 Memory map

Figure 8 shows addressable memory space in memory map.

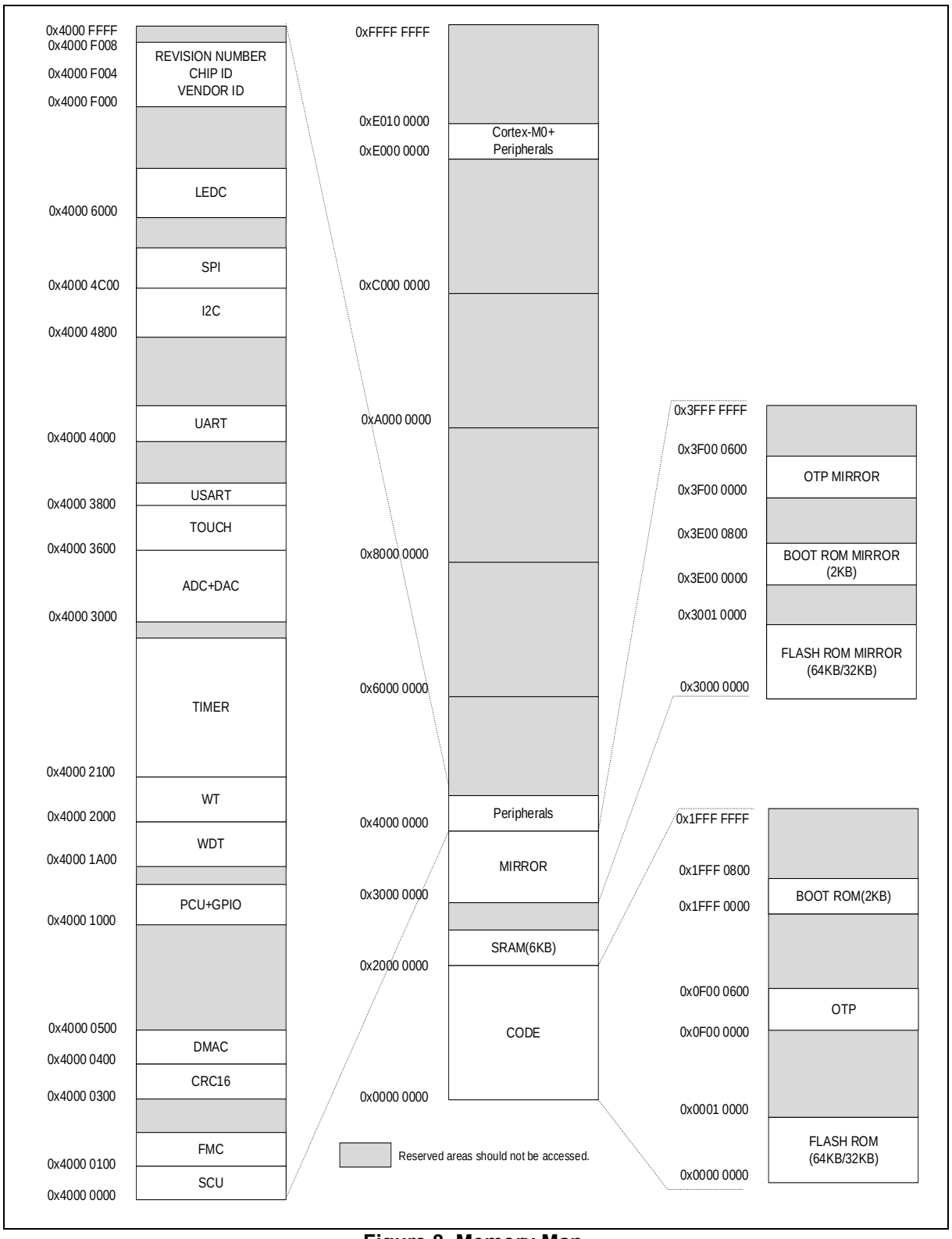


Figure 8. Memory Map

3.2.3 Embedded SRAM

A31G21x series have a block of 0-wait on-chip SRAM. The size of SRAM is 6KB and its base address is 0x2000_0000.

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.

3.2.4 Flash memory overview

A31G21x series provides internal 64KB code flash memory and its controller. This is enough to control the general system. Self-programming is available and ISP and SWD programming is also supported in boot or debugging mode.

Instruction and data cache buffer are present and overcome the low bandwidth flash memory. CPU can access flash memory with one wait state up to 48MHz bus frequency.

3.3 Boot mode

Boot mode pins

A31G21x series has boot mode option to program internal flash memory.

Boot mode can be entered by setting BOOT pin to 'L' at reset timing. (Normal state is 'H')

The boot mode supports UART boot and SPI boot.

UART boot uses TXD10/RXD10 port, and SPI boot uses MOSI10/MISO10/SCK10/SS10 port.

The pins for boot mode are listed as following: The pins for boot mode are listed in Table 5.

Table 5. Boot Mode Pin List

Block	Pin Name	Dir	Description
SYSTEM	nRESET/PC5	I	Reset Input signal
	BOOT/PB3	I	'L' to enter Boot mode
UART mode of USART10	RXD10/PB1	I	UART Boot Receive Data
	TXD10/PB0	O	UART Boot Transmit Data
SPI mode of USART10	SS10/PB3	I	SPI Boot Slave Selectable after Boot ROM
	SCK10/PB2	I	SPI Boot Clock Input
	MISO10/PB1	I	SPI Boot Data Input with function exchange
	MOSI10/PB0	O	SPI Boot Data Output with function exchange

Boot mode connections

User can design a target board using any of boot mode ports such as SPI or UART mode of USART10. Sample connection diagrams of boot mode are introduced in the following figures:

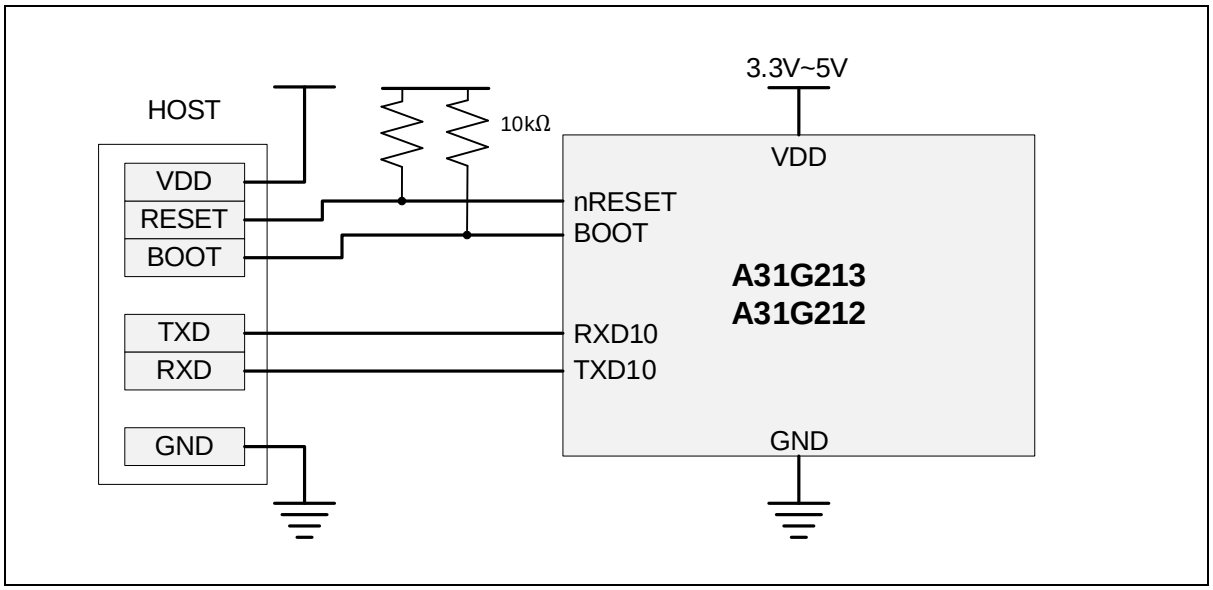


Figure 9. Connection Diagram of UART Boot

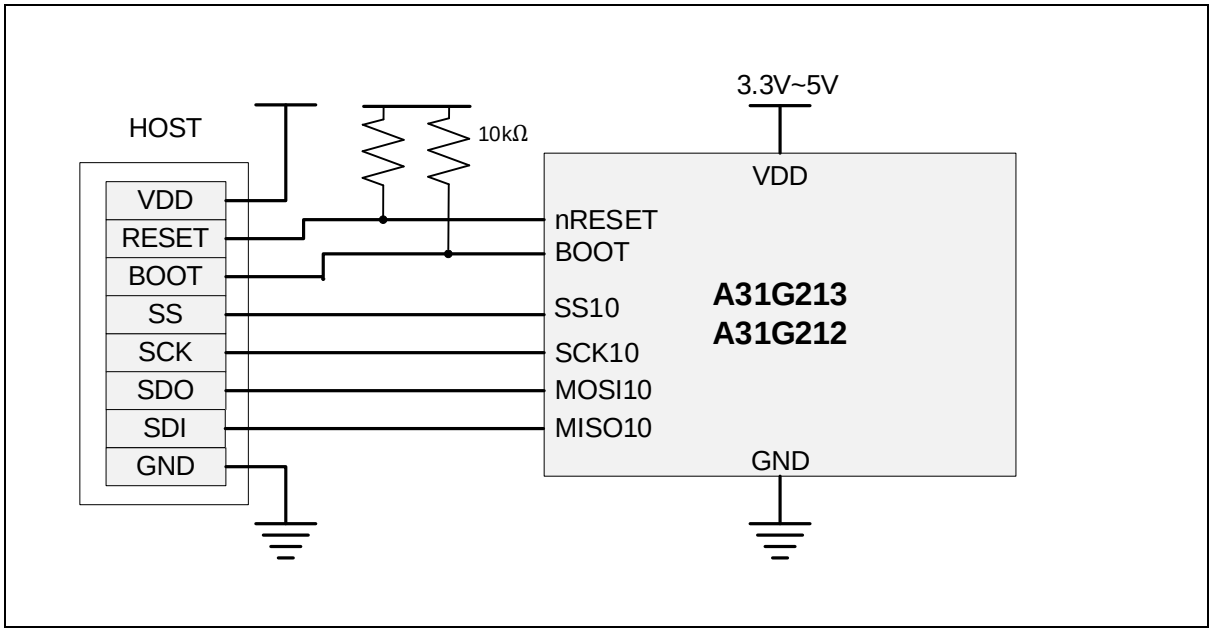
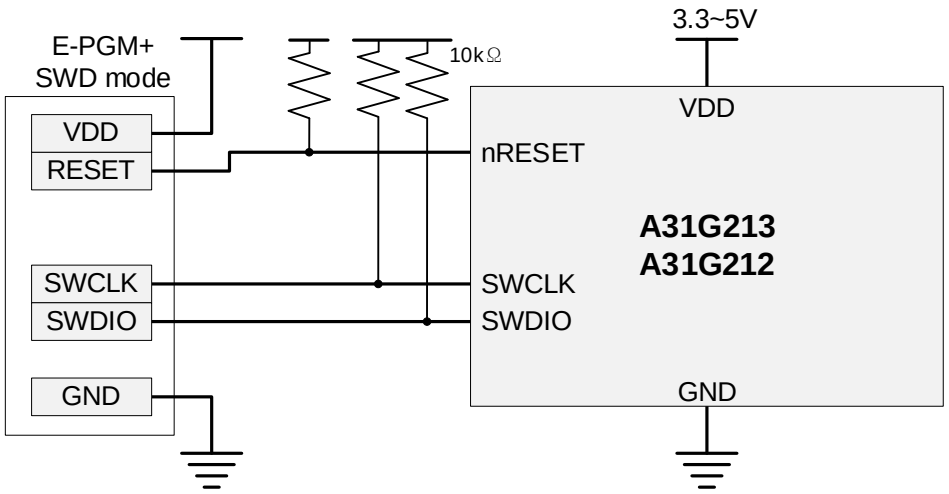


Figure 10. Connection Diagram of SPI Boot

SWD mode connections

A user can use SWD mode for writing with E-PGM+. This mode can be used for writing & debugging.



NOTES:

1. A31G21x series should not be connected to SWD mode in Boot Mode.

Figure 11. Connection Diagram of E-PGM+ and SWD Port

4 System Control Unit (SCU)

A31G21x series have a built-in intelligent power control block which manages system analog blocks and operating modes. System control unit (SCU) block controls an internal reset and clock signals to maintain optimize system performance and power dissipation.

Four pins in Table 6 are assigned for SCU block.

Table 6. SCU Pins

Pin name	Type	Description
nRESET/PC5	I	External Reset Input
XIN/XOUT	OSC	External Crystal Oscillator
SXIN/SXOUT	OSC	External sub-Crystal Oscillator
CLKO	O	Clock Output Monitoring Signal

4.1 SCU block diagram

In this subsection, SCU block diagram is introduced in Figure 12.

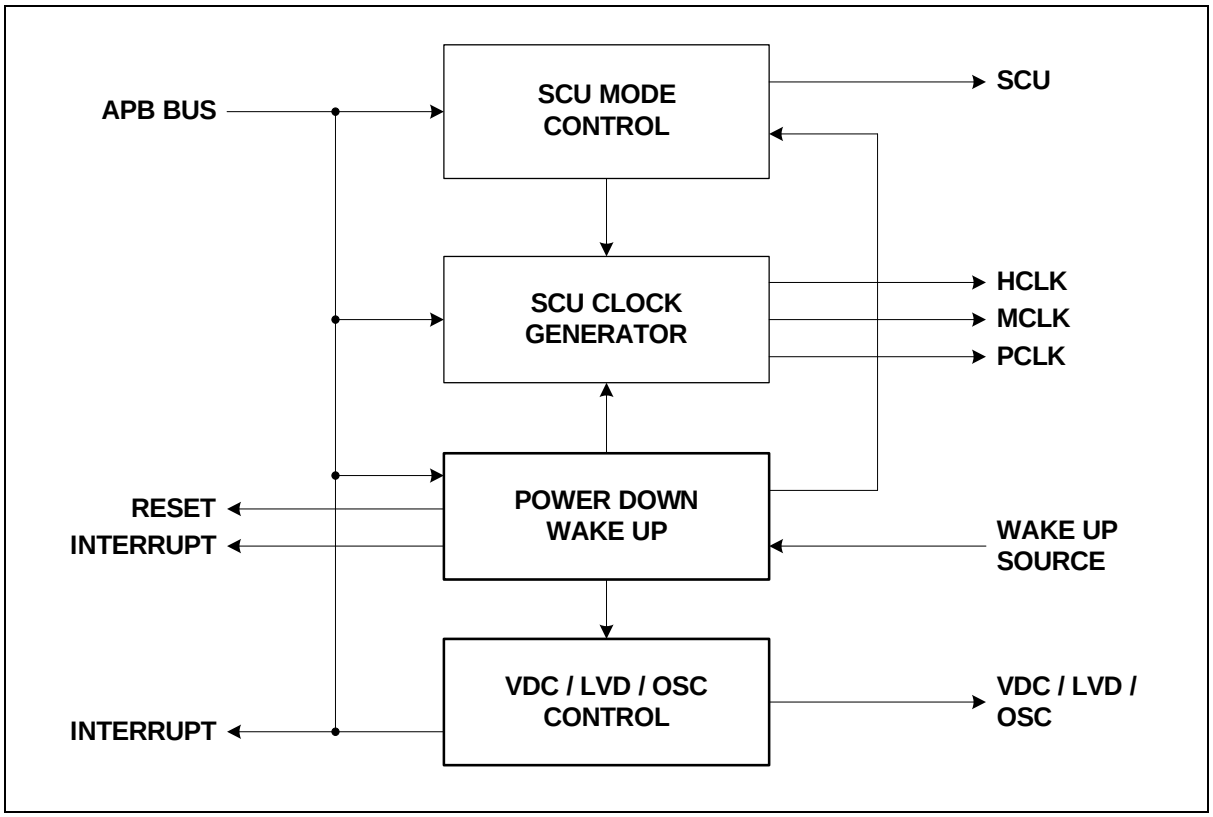


Figure 12. SCU Block Diagram

4.2 Clock system

A31G21x series have two main operating clocks. One is HCLK which produces a clock signal both for CPU and AHB bus system. The other is PCLK which produces a clock signal for peripheral systems.

A user can keep the clock system variation under software control. Through Figure 13 and Table 7, users learn about the clock system of A31G21x devices and clock sources.

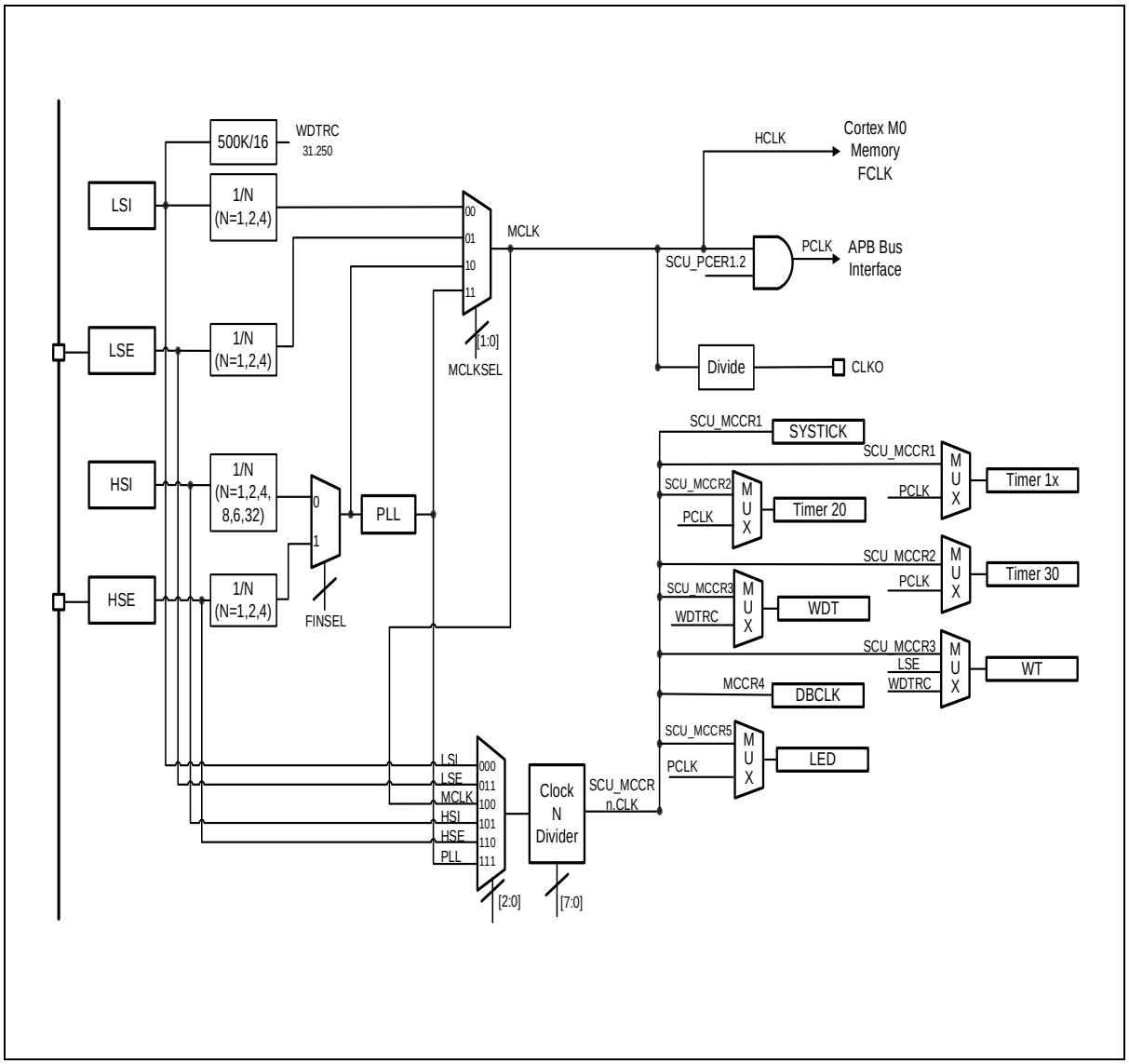


Figure 13. Clock Tree Configuration

All multiplexers switching clock sources have a glitch-free circuit in each. So a clock can be switched without glitch risks. When a user tries to change a clock mux control, both of clock sources must be alive. If one of them is not alive, clock change operation is stopped and system will be halted and not be recovered.

Table 7. Clock Sources

Clock name	Frequency	Description
HSE	2-16 MHz	High Speed External Oscillator
LSE	32.768 KHz	Low Speed External Oscillator
HSI	32 MHz	High Speed Internal OSC
LSI	500 KHz	Low Speed Internal OSC

4.2.1 HCLK clock domain

HCLK clock feeds the clock to the CPU and the AHB bus. Cortex-M0+ CPU requires 2 clocks related with FCLK and HCLK. FCLK is free running clock and it is always running except in power down mode. HCLK can be stopped in SLEEP mode and POWER-DOWN mode. BUS system and memory systems are operated by MCLK clock. Maximum bus operating clock speed is 48MHz.

4.2.2 PCLK clock domain

PCLK is the master clock of all peripherals. Each peripheral clock is enabled by SCU_PCER1, and SCU_PCER2 registers can be used by each peripheral. Before enabling the PCLK input clock of each block, it can't be accessible even reading its registers. It can be stopped in power down mode.

4.2.3 Clock configuration procedure

After power-up, the default system clock is fed by LSI (500KHz) clock. LSI is default enabled at power-up sequence. The other clock sources will be enabled by user controls with the LSI system clock.

HSI (32MHz) clock can be enabled by SCU_CSCR register.

HSE (2-16MHz) clock can be enabled by SCU_CSCR register. Before enable HSE block, the pin mux configuration should be set for XIN, XOUT function. PF1 and PF0 pins are shared with HSE's XIN and XOUT function – PF_MOD and PF_AFSR1 registers should be configured properly. After enabling the HSE block, you must wait for more than 5msec time to ensure stable operation of crystal oscillation.

LSE (32.768KHz) clock can be enabled by SCU_CSCR register. Before enable LSE block, the pin mux configuration should be set for SXIN, SXOUT function. PF2 and PF3 pins are shared with LSE's SXIN and SXOUT function – PF_MOD and PF_AFSR1 registers should be configured properly. After enabling the LSE block, you must wait for more than 10msec time to ensure stable operation of crystal oscillation.

You can change the MCLK by SCU_SCCR Register.

You can find an example flow chart to configure the system clock in below Figure. You can find an example flow chart configuring the system clock in Figure 14.

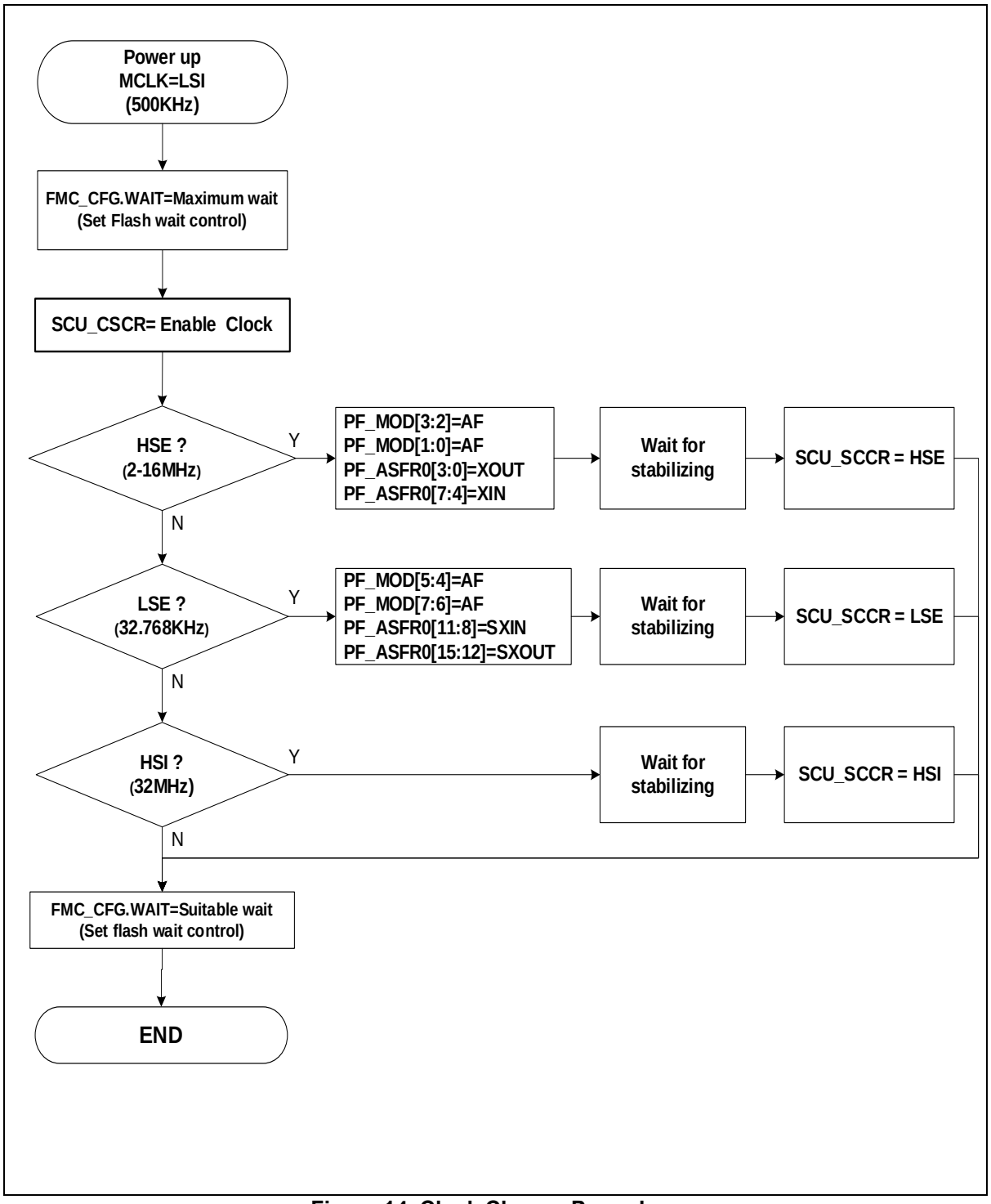


Figure 14. Clock Change Procedure

When you speed up the system clock to the maximum operating frequency, you should check flash wait control configuration. Flash read access time is one of limitation factor for performance. The wait control recommendation is suggested in Table 8.

Table 8. Flash Wait Control Recommendation

FM.CFG.WAIT	FLASH access wait	Available maximum system clock frequency
000	0 clock wait	Up to 20MHz
001	1 clock wait	Up to 40MHz
010	2 clock wait	Up to 48MHz
011	3 clock wait	Up to 48MHz
100	4 clock wait	Up to 48MHz
11x	5 clock wait	Up to 48MHz

4.3 Reset

A31G21x series have two system reset options. One is to cold reset that is effective during power up or down sequence. The other is warm reset which is generated by several reset sources. A reset event makes a chip to turn to an initial state. Reset sources of the cold reset and the warm reset are listed in Table 9.

Table 9. Reset Sources of Cold Reset and Warm Reset

Reset sources	
• nRESET Pin • WDT reset • LVD reset • MCLK Fail reset	• HSE Fail reset • S/W reset • CPU request reset

4.3.1 Cold reset

Cold reset is important feature of the chip when power is up. This characteristic will globally affect the system boot. Internal VDC is enabled when VDDEXT power is turn on. Internal POR trigger level is 1.4V of VDDEXT voltage out level. At this time, boot operation is started. The LSI clock is enabled and counts 4.25msec time for internal VDC level stabilizing. In this time, VDDEXT voltage level should be over than initial LVD level (1.65V). After 4.25msec counting, the cold reset is released and counts 0.4msec time for warm reset synchronizing. After released cold and warm reset, BOOTROM and CPU are running.

Figure 15 shows power-up sequence and internal reset waveforms.

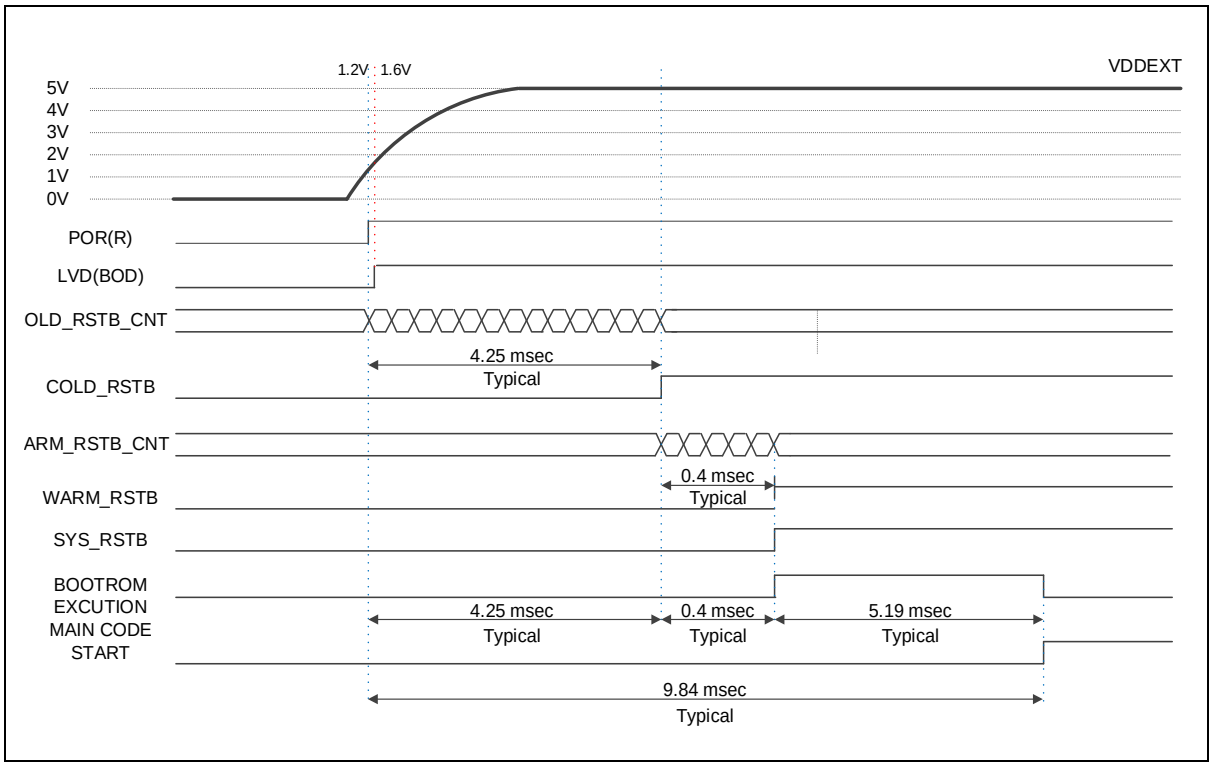


Figure 15. Power-up Procedure

4.3.2 Warm reset

Warm reset event has several reset sources and some parts of chip returns to an initial state when the warm reset condition is occurred.

The warm reset source is controlled by SCU_RSER register and the status is appeared in SCU_RSSR register. The reset for each peripheral blocks is controlled by SCU_PRER register. The reset can be masked independently.

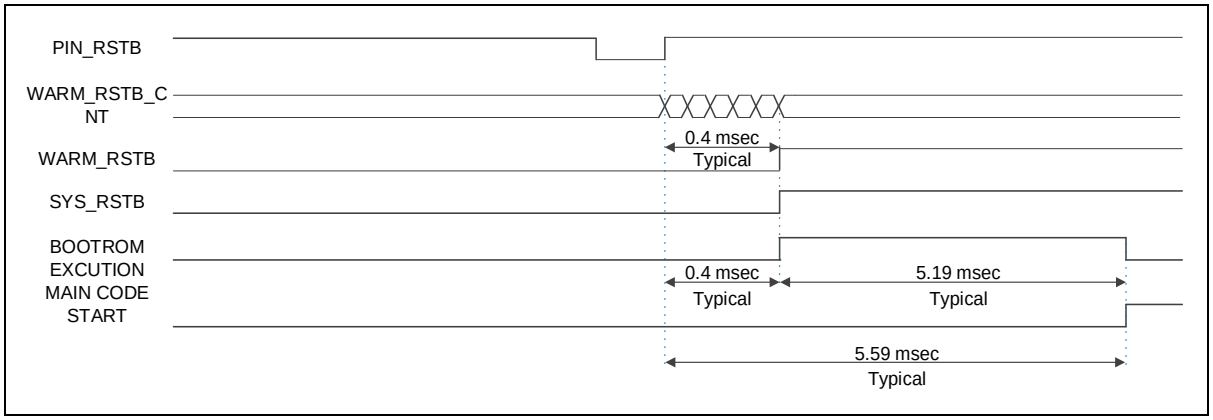


Figure 16. Warm Reset Diagram

4.3.3 LVR reset

Voltage level of LVR is set by the low voltage reset configuration register (SCULV_LVRCNFG).

The LVR reset status is appeared in RSTSSR register. The reset for LVR is controlled by SCULV_LVRCR register. The register is cleared to "0x00" on POR reset.

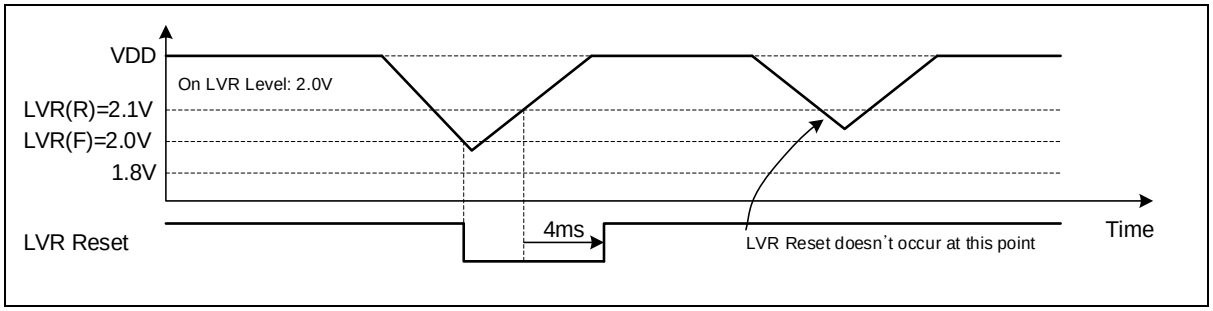
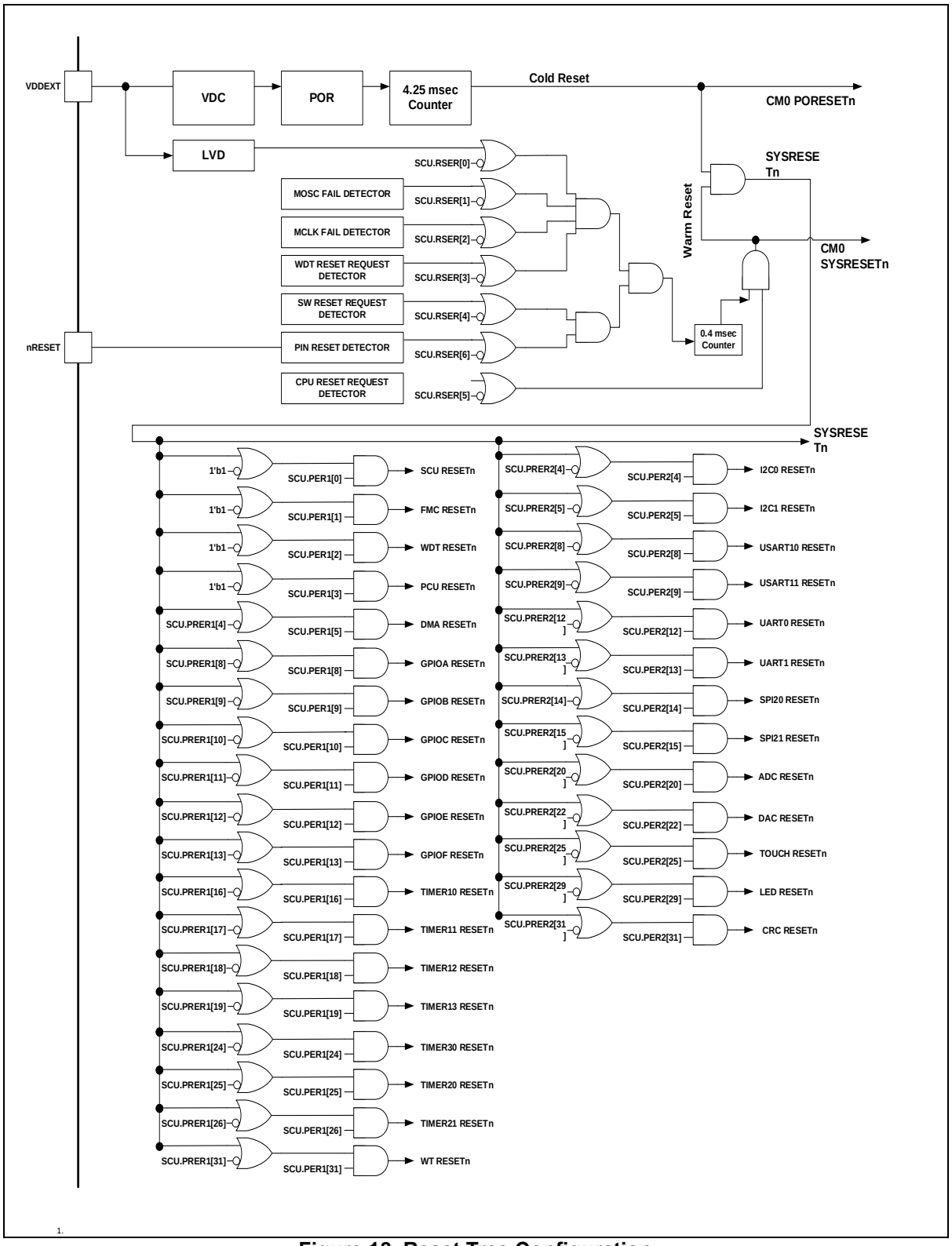


Figure 17. LVR Reset Timing Diagram

4.3.4 Reset tree

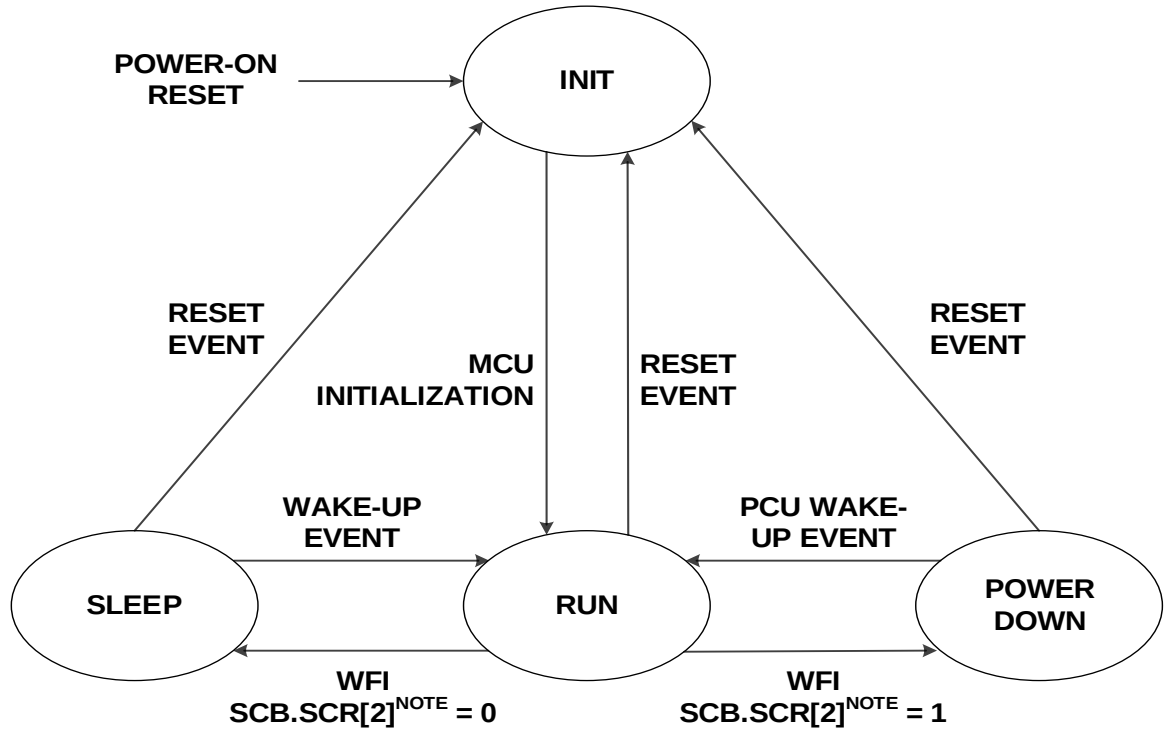


1.

Figure 18. Reset Tree Configuration

4.4 Operation mode

INIT mode is initial state of the chip when reset is asserted. The RUN mode is max performance of the CPU with high-speed clock system. And the SLEEP and the POWER-DOWN mode can be used as the low power consumption mode. The low power consumption is achieved by halting processor core and unused peripherals. Figure 19 describes transition between the operation modes.



NOTE: SCB.SCR is System Control Register in System Control Block.

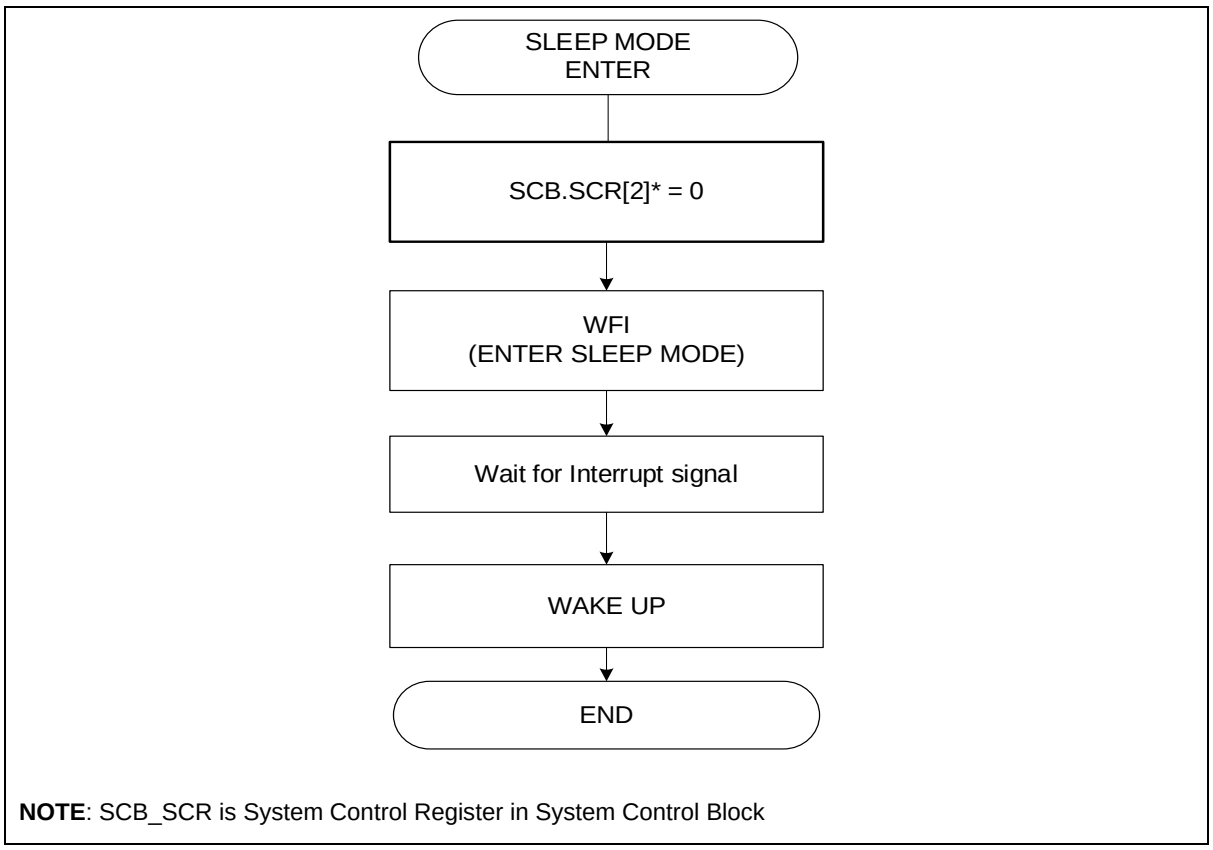
Figure 19. Transition between Operation Modes

4.4.1 RUN mode

In RUM mode, CPU and the peripheral hardware operate with a high-speed clock. After a reset followed by INIT state, the system enters in the RUN mode.

4.4.2 SLEEP mode

Only CPU is stopped in this mode. Each peripheral function can be enabled by the function enable and clock enable bit in the SCU_PER and SCU_PCER register.

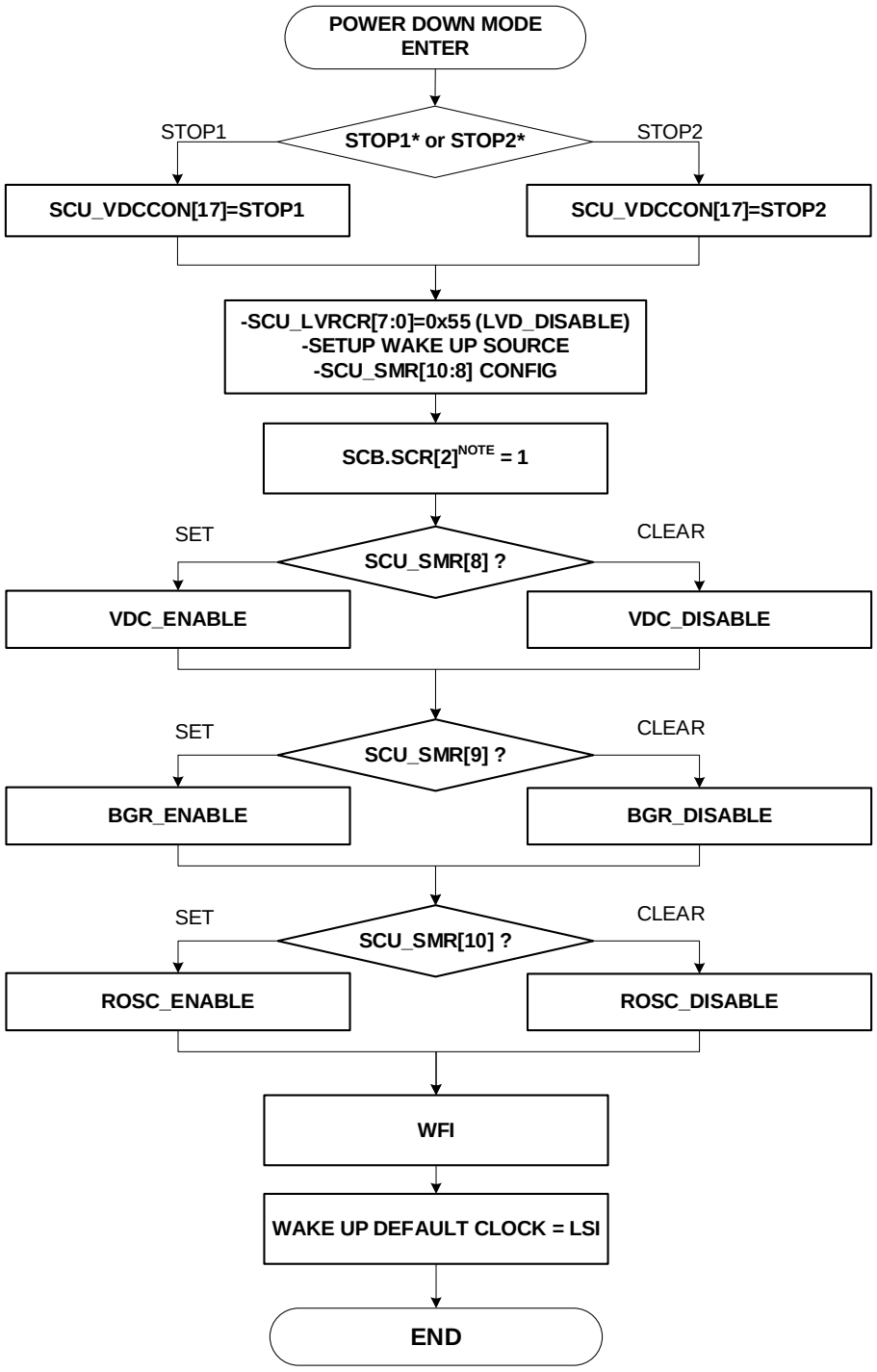


NOTE: SCB_SCR is System Control Register in System Control Block

Figure 20. SLEEP Mode Operation Sequence

4.4.3 POWER-DOWN mode

The A31G21x series have three POWER-DOWN modes:



NOTES:

1. When STOP1, VDC current is limited 500uA.
2. When STOP2, VDC current is limited 10uA.
3. SCB.SCR is System Control Register in System Control Block.

Figure 21. POWER-DOWN Mode Block Diagram

4.5 Registers

Base address of SCU (chip configuration) and register map are introduced in the followings:

Table 10. Base Address of SCU (Chip Configuration)

Name	Base address
SCU (Chip Configuration)	0x4000_F000

Table 11. SCU Register Map (Chip Configuration)

Name	Offset	Type	Description	Reset value	Ref.
SCUCC_VENDORID	0x00	RO	Vendor Identification Register	0x4142_4F56	4.5.1
SCUCC_CHIPID	0x04	RO	Chip Identification Register.	0x4D31_A00x	4.5.2
SCUCC_REVNR	0x08	RO	Revision Number Register	0x0000_00xx	4.5.3

NOTE: 'RO' means 'Read Only'.

Base address of SCU and register map are introduced in the followings:

Table 12. Base Address of SCU

Name	Base address
SCU	0x4000_0000

Table 13. SCU Register Map

Name	Offset	Type	Description	Reset value	Ref.
SCU_SMR	0x0004	RW	System Mode Register	0x0000_0000	4.5.4
SCU_SCR	0x0008	RW	System Control Register	0x0000_0000	4.5.5
SCU_WUER	0x0010	RW	Wake up source enable register	0x0000_0000	4.5.6
SCU_WUSR	0x0014	RO	Wake up source status register	0x0000_0000	4.5.7
SCU_RSER	0x0018	RW	Reset source enable register	0x0000_0069	4.5.8
SCU_RSSR	0x001C	RW	Reset source status register	0x0000_0080*	4.5.9
SCU_PRER1	0x0020	RW	Peripheral reset enable register 1	0x870F_3F1F*	4.5.10
SCU_PRER2	0x0024	RW	Peripheral reset enable register 2	0xA250_F330*	4.5.11
SCU_PER1	0x0028	RW	Peripheral enable register 1	0x0000_000F*	4.5.12
SCU_PER2	0x002C	RW	Peripheral enable register 2	0x0000_0100*	4.5.13
SCU_PCER1	0x0030	RW	Peripheral clock enable register 1	0x0000_000F*	4.5.14
SCU_PCER2	0x0034	RW	Peripheral clock enable register 2	0x0000_0100*	4.5.15
SCU_PPCLKSR	0x0038	RW	Peripheral clock selection register	0x0000_0000	4.5.16
SCU_CSCR	0x0040	RW	Clock Source Control register	0x0000_0800	4.5.17
SCU_SCCR	0x0044	RW	System Clock Control register	0x0000_0000	4.5.18
SCU_CMR	0x0048	RW	Clock Monitoring register	0x0000_0090	4.5.19
SCU_NMIR	0x004C	RW	NMI control register	0x0000_0000	4.5.20
SCU_COR	0x0050	RW	Clock Output Control register	0x0000_000F	4.5.21
SCU_PLLCON	0x0060	RW	PLL Control register	0x0000_0000	4.5.22
SCU_VDCCON	0x0064	RW	VDC Control register	0x0000_007F	4.5.23

Table 13. SCU Register Map

Name	Offset	Type	Description	Reset value	Ref.
SCU_LSICON	0x006C	RW	Internal Ring OSC Control Register	0x0000_0001	4.5.24
SCU_EOSCR	0x0080	RW	External Oscillator control register	0x0000_1014	4.5.25
SCU_EMODR	0x0084	RO	External mode pin read register	0x0000_0000	4.5.26
SCU_RSTDBCR	0x0088	RW	Pin Reset Debounce Control Register	0x0000_0000	4.5.27
SCU_MCCR1	0x0090	RW	Misc Clock Control register 1	0x0000_0000	4.5.28
SCU_MCCR2	0x0094	RW	Misc Clock Control register 2	0x0000_0000	4.5.29
SCU_MCCR3	0x0098	RW	Misc Clock Control register 3	0x0000_0000	4.5.30
SCU_MCCR4	0x009C	RW	Misc Clock Control register 4	0x0000_0000	4.5.31
SCU_MCCR5	0x00A0	RW	Misc Clock Control register 5	0x0000_0000	4.5.32

NOTES:

1. 'RO' means 'Read Only'.
2. 'RW' means 'Read and Write'.

Base address of LVI/LVR unit and register map are introduced in the followings:

Table 14. Base Address of LVI/LVR

NAME	BASE ADDRESS
SCULV(LVI/LVR)	0x4000_5100

Table 15. LVI/LVR Register Map

Name	Offset	Type	Description	Reset value	Ref.
SCULV_LVICR	0x00	RW	Low Voltage Indicator Control Register	0x0000_0000	4.6.33
SCULV_LVRCR	0x04	RW	Low Voltage Reset Control Register	0x0000_0000	4.6.34
SCULV_LVRCNFIG	0x08	RW	Configuration for Low Voltage Reset	0x0000_000F	4.6.35

NOTES:

1. 'RO' means 'Read Only' and 'RW' means 'Read and Write'.
2. 'RC' means 'Read and Write 1 Clear'.

4.5.1 SCUCC_VENDORID: vendor ID register

SCUCC_VENDORID register shows the vendor identification information. This is a 32-bit read-only SCU_VENDORIDVendor ID 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			
0x4D31A008 or 0x4D31A009			
RO			

31	CHIPID	Chip Identification bits.
0		0x4D31A008 64KB flash memory for program
		0x4D31A009 32KB flash memory for program

4.5.2 SCU_CHIPID: chip ID register

SCU_CHIPID Register shows Chip identification information. This register is 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			
0x4D31A008 or 0x4D31A009			
RO			

31	CHIPID	Chip Identification bits.
0		0x4D31A008 64KB flash memory for program
		0x4D31A009 32KB flash memory for program

4.5.3 SCU_REVNR: revision number register

Revision Number register is 32-bit read-only register. This Register is able to 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
-			XX
-			RO

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

4.5.4 SCU_SMR: system mode register

Current operating mode is shown in this SCU mode register. The previous operating mode will be saved in this register after reset event. There is VDC On/Off control bit in power-down mode.

SCU_SMR=0x4000_0004

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			ROSCAON	BGRAON	VDCAON	Reserved	PREVMODE	Reserved
-			0	0	0	-	00	-
-			RW	RW	RW	-	RO	-

10	ROSCAON	ROSC Always on select bit in POWER-DOWN mode
	0	ROSC is automatically off entering POWER-DOWN mode
	1	ROSC isn't automatically off entering POWER-DOWN mode
9	BGRAON	BGR Always on select bit in POWER-DOWN mode
	0	BGR is automatically off entering POWER-DOWN mode
	1	BGR isn't automatically off entering POWER-DOWN mode
8	VDCAON	VDC Always on select bit in POWER-DOWN mode
	0	VDC is automatically off entering POWER-DOWN mode
	1	VDC isn't automatically off entering POWER-DOWN mode
5	PREVMODE	Previous operating mode before current reset event
4	00	Previous operating mode was RUN mode
	01	Previous operating mode was SLEEP mode
	10	Previous operating mode was POWER-DOWN mode
	11	Previous operating mode was INIT mode

4.5.5 SCU_SCR: system control register

It is possible to reset MCU as SWRST bit set.

System Mode Register is 32-bit register.

SCU_SCR=0x4000_0008

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		Reserved		SWRST
0x0000		-		0
WO		-		RW

31	WTIDKY	Write Identification Key
16		On writes, write 0x9EB3 to these bits, otherwise the write is ignored.
0	SWRST	Internal soft reset activation bit (check SCU_RSER[4] for reset)
	0	Normal operation
	1	Internal soft reset generated and auto cleared

4.5.6 SCU_WUER: wakeup source enable register

Enable wakeup source when the chip is in the POWER-DOWN mode. Wakeup sources which will be used the source of chip wakeup should be enabled in each bit field. If the source is used as a wakeup source, the corresponding bit should be written with '1'. If the source is not used as a wakeup source, the bit should be written with '0'.

SCU_WUER=-0x4000_0010

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			GPIOFWUE	GPIOEWUE	GPIODWUE	GPIOCWUE	GPIOBWUE	GPIOAWUE	Reserved	WTWUE	WDTWUE	LVDWUE
-			0	0	0	0	0	0	-	0	0	0
-			RW	RW	RW	RW	RW	RW	-	RW	RW	RW

13	GPIOFWUE	Enable wakeup source of GPIOF port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
12	GPIOEWUE	Enable wakeup source of GPIOE port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
11	GPIODWUE	Enable wakeup source of GPIOD port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
10	GPIOCWUE	Enable wakeup source of GPIOC port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
9	GPIOBWUE	Enable wakeup source of GPIOB port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
8	GPIOAWUE	Enable wakeup source of GPIOA port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
2	WTWUE	Enable wakeup source of watch timer event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
1	WDTWUE	Enable wakeup source of watchdog timer event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
0	LVDWUE	Enable wakeup source of LVD event
	0	Not used for wakeup source
	1	Enable the wakeup event generation

4.5.7 SCU_WUSR: wakeup source status register

When the system is waked up by any wakeup source, the wakeup source is identified by reading this register. When the bit is set 1, the related wakeup source issues the wake-up to the SCU. The bit will be cleared when the event source is cleared by the software.

SCU_WUSR=0x4000_0014

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			GPIOFWU	GPIOEWU	GPIODWU	GPIOCWU	GPIOBWU	GPIOAWU	Reserved	WTWU	WDTWU	LVDWU
-			0	0	0	0	0	0	-	0	0	0
-			RO	RO	RO	RO	RO	RO	-	RO	RO	RO

13	GPIOFWU	Status of wakeup source of GPIOF port pin change event
		0 No wakeup event
		1 Wakeup event was generated
12	GPIOEWU	Status of wakeup source of GPIOE port pin change event
		0 No wakeup event
		1 Wakeup event was generated
11	GPIODWU	Status of wakeup source of GPIOD port pin change event
		0 No wakeup event
		1 Wakeup event was generated
10	GPIOCWU	Status of wakeup source of GPIOC port pin change event
		0 No wakeup event
		1 Wakeup event was generated
9	GPIOBWU	Status of wakeup source of GPIOB port pin change event
		0 No wakeup event
		1 Wakeup event was generated
8	GPIOAWU	Status of wakeup source of GPIOA port pin change event
		0 No wakeup event
		1 Wakeup event was generated
2	WTWU	Status of wakeup source of watch timer event
		0 No wakeup event
		1 Wakeup event was generated
1	WDTWU	Status of wakeup source of watchdog timer event
		0 No wakeup event
		1 Wakeup event was generated
0	LVDWU	Status of wakeup source of LVD event
		0 No wakeup event
		1 Wakeup event was generated

4.5.8 SCU_RSER: reset source enable register

The reset source to the CPU can be selected by RSER register. When writing '1' in the bit field of each reset source, the reset source event will be transferred to reset generator. When writing '0' in the bit field of each reset source, the reset source event will be masked and not generate the reset event.

SCU_RSER=0x4000_0018

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																									PINRST	CPURST	SWRST	WDTRST	MCKFRST	MOFRST	LVDRST
-																									1	1	0	1	0	0	1
-																									RW	RW	RW	RW	RW	RW	RW

6	PINRST	External pin reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
5	CPURST	CPU request reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
4	SWRST	Software reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
3	WDTRST	Watchdog Timer reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
2	MCKFRST	MCLK Clock fail reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
1	MOFRST	HSE Clock fail reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled
0	LVDRST	LVD reset enable bit	
		0	Reset from this event is masked
		1	Reset from this event is enabled

4.5.9 SCU_RSSR: reset source status register

The SCU_RSSR shows the reset source information when reset event is occurred. '1' shows reset event was existed and '0' shows reset event was not existed for corresponding reset source.

When reset source is founded, write '1' into the corresponding bit will clear the reset status.

SCU_RSSR=0x4000_001C

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			PORST	PINRST	CPURST	SWRST	WDTRST	MCKFRST	MOFRST	LVDRST
-			1	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

7	PORST	Power-on reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
6	PINRST	External pin reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
5	CPURST	CPU request reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
4	SWRST	Software reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
3	WDTRST	Watchdog Timer reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
2	MCKFRST	MCLK Clock fail reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
1	MOFRST	HSE Clock fail reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
0	LVDRST	LVD reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status

4.5.10 SCU_PRER1: peripheral reset enable register 1

The reset of each peripheral by event reset, can be masked by user setting. SCU_PRER1/SCU_PRER2 register will control the enable of the event reset. If the corresponding bit is '1', the peripheral corresponded with this bit, accepts the reset event. Otherwise, the peripheral is protected from reset event and maintain current operation.

SCU_PRER1=0x4000_0020

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
WT	Reserved	TIMER21	TIMER20	TIMER30	Reserved	TIMER13	TIMER12	TIMER11	TIMER10	Reserved	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	Reserved	DMA	PCU	WDT	FMC	SCU
1	-	1	1	1	-	1	1	1	1	-	1	1	1	1	1	1	-	1	1	1	1	1
RW	-	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW

31	WT	WT reset mask
26	TIMER21	TIMER21 reset mask
25	TIMER20	TIMER20 reset mask
24	TIMER30	TIMER30 reset mask
19	TIMER13	TIMER13 reset mask
18	TIMER12	TIMER12 reset mask
17	TIMER11	TIMER11 reset mask
16	TIMER10	TIMER10 reset mask
13	GPIOF	GPIOF reset mask
12	GPIOE	GPIOE reset mask
11	GPIOD	GPIOD reset mask
10	GPIOC	GPIOC reset mask
9	GPIOB	GPIOB reset mask
8	GPIOA	GPIOA reset mask
4	DMA	DMA reset mask
3	PCU	Port controller reset mask
2	WDT	Watchdog Timer reset mask
1	FMC	Flash memory controller reset mask
0	SCU	Power Management Unit reset mask

4.5.11 SCU_PRER2: peripheral reset enable register 2

Peripheral Reset Enable Register 2 is 32-bit register.

SCU_PRER2=0x4000_0024

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
CRC	Reserved	LED	Reserved	TOUCH	Reserved		DAC	Reserved	ADC	Reserved	SPI21	SPI20	UART1	UART0	Reserved	USART11	USART10	Reserved	I2C1	I2C0	Reserved
1	-	1	-	1	-		1		1	-	1	1	1	1	-	1	1	-	1	1	-
RW	-	RW	-	RW	-		RW		RW	-	RW	RW	RW	RW	-	RW	RW	-	RW	RW	-

31	CRC	CRC reset enable
29	LED	LED reset enable
25	TOUCH	TOUCH reset enable
22	DAC	DAC reset enable
20	ADC	ADC reset enable
15	SPI21	SPI21 reset enable
14	SPI20	SPI20 reset enable
13	UART1	UART1 reset enable
12	UART0	UART0 reset enable
9	USART11	USART11 reset enable
8	USART10	USART10 reset enable
5	I2C1	I2C1 reset enable
4	I2C0	I2C0 reset enable

4.5.12 SCU_PER1: peripheral enable register 1

To use peripheral unit, it should be activated by writing '1' to the correspond bit in the SCU_PER1/ SCU_PER2 register. Before the activation, the peripheral will stay in reset state.

All the peripherals enabled by default. To disable the peripheral unit, write '0' to the correspond bit in the SCU_PER1/ SCU_PER2 register, and then the peripheral enters the reset state.

SCU_PER1=0x4000_0028

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
WT	Reserved				TIMER21	TIMER20	TIMER30	Reserved				TIMER13	TIMER12	TIMER11	TIMER10	Reserved		GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	Reserved			DMA	Reserved			
0	-				0	0	0	-				0	0	0	0	-		0	0	0	0	0	0	-			0	1111			
RW	-				RW	RW	RW	-				RW	RW	RW	RW	-		RW	RW	RW	RW	RW	RW	-			RW	-			

31	WT	WT function enable
26	TIMER21	TIMER21 function enable
25	TIMER20	TIMER20 function enable
24	TIMER30	TIMER30 function enable
19	TIMER13	TIMER13 function enable
18	TIMER12	TIMER12 function enable
17	TIMER11	TIMER11 function enable
16	TIMER10	TIMER10 function enable
13	GPIOF	GPIOF function enable
12	GPIOE	GPIOE function enable
11	GPIOD	GPIOD function enable
10	GPIOC	GPIOC function enable
9	GPIOB	GPIOB function enable
8	GPIOA	GPIOA function enable
4	DMA	DMA function enable

4.5.13 SCU_PER2: peripheral enable register 2

Peripheral Enable Register 2 is 32-bit register.

SCU_PER2=0x4000_002C

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
CRC	Reserved	LED	Reserved	TOUCH	Reserved		DAC	Reserved	ADC	Reserved	SPI21	SPI20	UART1	UART0	Reserved	USART11	USART10	Reserved	I2C1	I2C0	Reserved
0	-	0	-	0	-		0		0	-	0	0	0	0	-	0	1	-	0	0	-
RW	-	RW	-	RW	-		RW		RW	-	RW	RW	RW	RW	-	RW	RW	-	RW	RW	-

31	CRC	CRC function enable
29	LED	LED function enable
25	TOUCH	TOUCH function enable
22	DAC	DAC function enable
20	ADC	ADC function enable
15	SPI21	SPI21 function enable
14	SPI20	SPI20 function enable
13	UART1	UART1 function enable
12	UART0	UART0 function enable
9	USART11	USART11 function enable
8	USART10	USART10 function enable
5	I2C1	I2C1 function enable
4	I2C0	I2C0 function enable

4.5.14 SCU_PCER1: peripheral clock enable register 1

To use peripheral unit, its clock should be activated by writing '1'to the corresponding bit in the SCU_PCER1/ SCU_PCER2 register. Before enabling its clock, the peripheral won't operate properly.

To stop the clock of the peripheral unit, write '0'to the correspond bit in the SCU_PCER1/ SCU_PCER2 register, and then the clock of the peripheral is stopped.

SCU_PCER1=0x4000_0030

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
WT	Reserved	TIMER21	TIMER20	TIMER30	Reserved	TIMER13	TIMER12	TIMER11	TIMER10	Reserved	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	Reserved	DMA	Reserved
0	-	0	0	0	-	0	0	0	0	-	0	0	0	0	0	0	-	0	1111
RW	-	RW	RW	RW	-	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	-	RW	-

31	WT	WT clock enable
26	TIMER21	TIMER21 clock enable
25	TIMER20	TIMER20 clock enable
24	TIMER30	TIMER30 clock enable
19	TIMER13	TIMER13 clock enable
18	TIMER12	TIMER12 clock enable
17	TIMER11	TIMER11 clock enable
16	TIMER10	TIMER10 clock enable
13	GPIOF	GPIOF clock enable
12	GPIOE	GPIOE clock enable
11	GPIOD	GPIOD clock enable
10	GPIOC	GPIOC clock enable
9	GPIOB	GPIOB clock enable
8	GPIOA	GPIOA clock enable
4	DMA	DMA clock enable

4.5.15 SCU_PCER2: peripheral clock enable register 2

To use peripheral unit, its clock should be activated by writing '1' to the correspond

SCU_PCER2=0x4000_0034

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
CRC	Reserved	LED	Reserved	TOUCH	Reserved		DAC	Reserved	ADC	Reserved	SPI21	SPI20	UART1	UART0	Reserved	USART11	USART10	Reserved	I2C1	I2C0	Reserved
0	-	0	-	0	-		0		0	-	0	0	0	0	-	0	1	-	0	0	-
RW	-	RW	-	RW	-		RW		RW	-	RW	RW	RW	RW	-	RW	RW	-	RW	RW	-

31	CRC	CRC clock enable
29	LED	LED clock enable
25	TOUCH	TOUCH clock enable
22	DAC	DAC clock enable
20	ADC	ADC clock enable
15	SPI21	SPI21 clock enable
14	SPI20	SPI20 clock enable
13	UART1	UART1 clock enable
12	UART0	UART0 clock enable
9	USART11	USART11 clock enable
8	USART10	USART10 clock enable
5	I2C1	I2C1 clock enable
4	I2C0	I2C0 clock enable

4.5.16 SCU_PPCLKSR: peripheral clock selection register

Peripheral Clock Selection Register is 32-bit register. This Register is able to 32/16/8-bit access.

SCU_PPCLKSR=0x4000_0038

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		T1xCLK	Reserved	T20CLK	Reserved		T30CLK		Reserved	LEDCLK	Reserved		WTCLK	Reserved		WDTCLK
-		0	0	0	-		0		-	0	-		0	-		0
-		RW	RW	RW	-		RW		-	RW	-		RW	-		RW

22	T1xCLK	Timer 1x Clock Selection bit
		0 SCU_MCCR1 Timer1x clock
		1 PCLK clock
20	T20CLK	Timer 20 Clock Selection bit
		0 SCU_MCCR2 Timer20 clock
		1 PCLK clock
17	T30CLK	Timer 30 Clock Selection bit
		0 SCU_MCCR2 Timer30 clock
		1 PCLK clock
10	LEDCLK	LED Clock Selection bit
		0 SCU_MCCR5 LED clock
		1 PCLK clock
4	WTCLK	Watch Timer Clock Selection bit
3		00 SCU_MCCR3 WT clock
		01 LSE clock
		10 WDTRC clock
		11 Reserved
		NOTE: These bits should be changed during the WTEN bit of watch timer control register (WT_CR) is "0b".
0	WDTCLK	Watch-dog Timer Clock Selection bit
		0 WDTRC clock
		1 SCU_MCCR3 WDT clock

4.5.17 SCU_CSCR: clock source control register

The A31G21x series has multiple clock sources to generate internal operating clocks. Each clock sources can be controlled by SCU_CSCR register.

SCU_CSCR=0x4000_0040

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		LSECON	LSICON	HSICON	HSECON
0x0000		0000	1000	0000	0000
WO		RW	RW	RW	RW

31 16	WTIDKY	Write Identification Key	
		On writes, write 0xA507 to these bits, otherwise the write is ignored.	
15 12	LSECON	External crystal sub oscillator control	
		0XXX	Disable external sub crystal oscillator
		1000	Enable external sub crystal oscillator
		1001	Enable external sub crystal oscillator divide by 2
		1010	Enable external sub crystal oscillator divide by 4
		Other	Reserved
11 8	LSICON	Low speed internal oscillator control	
		0XXX	Disable low speed internal oscillator
		1000	Enable low speed internal oscillator
		1001	Enable low speed internal oscillator divide by 2
		1010	Enable low speed internal oscillator divide by 4
		Other	Reserved
7 4	HSICON	High speed internal oscillator control	
		0XXX	Disable high speed internal oscillator
		1000	Enable high speed internal oscillator
		1001	Enable high speed internal oscillator divide by 2
		1010	Enable high speed internal oscillator divide by 4
		1011	Enable high speed internal oscillator divide by 8
		1100	Enable high speed internal oscillator divide by 16
		1101	Enable high speed internal oscillator divide by 32
		1111	Reserved
3 0	HSECON	External crystal main oscillator control	
		0XXX	Disable external main crystal oscillator
		1000	Enable external main crystal oscillator
		1001	Enable external main crystal oscillator divide by 2
		1010	Enable external main crystal oscillator divide by 4
		Other	Reserved

4.5.18 SCU_SCCR: system clock control register

Select the system clock source in SCU_SCCR, and selected clock source becomes MCLK. Before changing clock, clock sources have to be alive by SCU_CSCR register.

SCU_SCCR=0x4000_0044

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		Reserved		FINSEL	MCLKSEL
0x0000		-		0	00
WO		-		RW	RW

31 16	WTIDKY	Write Identification Key	
		On writes, write 0x570A to these bits, otherwise the write is ignored.	
2	FINSEL	PLL input source FIN select register	
		0	HSI clock is used as FIN clock
		1	HSE clock is used as FIN clock
1 0	MCLKSEL	System clock select register	
		00	Internal ring oscillator(500KHz)
		01	LSE XTAL (32KHz)
		10	PLL bypassed clock
		11	PLL output clock

NOTE: When change MCLKSEL, both clock sources should be alive.
Ex) Both of HSI and HSE should be alive, otherwise the chip will do malfunction

4.5.19 SCU_CMR: clock monitoring register

The clock can be monitored by LSI for security purpose.

SCU_CMR=0x4000_0048

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		MCLKREC	Reserved	LSEMNT	LSEIE	LSEFAIL	LSESTS	MCLKMNT	MCLKIE	MCLKFAIL	MCLKSTS	HSEMNT	HSEIE	HSEFAIL	HSESTS
-		0	-	0	0	0	0	1	0	0	1	0	0	0	0
-		RW	-	RW	RW	RC	RO	RW	RW	RC	RO	RW	RW	RC	RO

15	MCLKREC	MCLK fail auto recovery	
		0	MCLK is changed to LSI by default when MCLKFAIL issued
		1	MCLK auto recovery is disabled
11	LSEMNT	External sub oscillator monitoring enable	
		0	External sub oscillator monitoring disabled
		1	External sub oscillator monitoring enabled
10	LSEIE	External sub oscillator fail interrupt enable	
		0	External sub oscillator fail interrupt disabled
		1	External sub oscillator fail interrupt enabled
9	LSEFAIL	External sub oscillator fail interrupt	
		0	External sub oscillator fail interrupt not occurred
		1	Read : External sub oscillator fail interrupt is pending Write : Clear pending interrupt
8	LSESTS	External sub oscillator status	
		0	Not oscillate
		1	External sub oscillator is working normally
7	MCLKMNT	MCLK monitoring enable	
		0	MCLK monitoring disabled
		1	MCLK monitoring enabled
6	MCLKIE	MCLK fail interrupt enable	
		0	MCLK fail interrupt disabled
		1	MCLK fail interrupt enabled
5	MCLKFAIL	MCLK fail interrupt	
		0	MCLK fail interrupt not occurred
		1	Read : MCLK fail interrupt is pending Write : Clear pending interrupt
4	MCLKSTS	MCLK clock status	
		0	No clock is present on MCLK
		1	Clock is present on MCLK
3	HSEMNT	External main oscillator monitoring enable	
		0	External main oscillator monitoring disabled
		1	External main oscillator monitoring enabled
2	HSEIE	External main oscillator fail interrupt enable	
		0	External main oscillator fail interrupt disabled
		1	External main oscillator fail interrupt enabled
1	HSEFAIL	External main oscillator fail interrupt	
		0	External main oscillator fail interrupt not occurred
		1	Read : External main oscillator fail interrupt is pending Write : Clear pending interrupt
0	HSESTS	External main oscillator status	
		0	Not oscillate
		1	External main oscillator is working normally

4.5.20 SCU_NMIR: NMI control register

SCU_NMIR is the non-maskable interrupt configuration register which can be set by software. There are three kinds of interrupt sources from WDT and SCU. It will jump to NMI handler if Selected NMI event occurred and it must check event status. For clearing occurred status, it should clear the interrupt flags of that peri occurred.

Write access key is required 0xA32C on SCU_NMIR[31:16] when write register.

SCU_NMIR=0x4000_004C

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
ACCESSCODE		Reserved	WDTINTSTS	MCLKFAILSTS	LVDSTS	Reserved	WDTINTEN	MCLKFAILEN	LVDEN
-		-	0	0	0	-	0	0	0
WO		-	RO	RO	RO	-	RW	RW	RW

31 16	ACCESSCODE	This field enables writing access to this register. Writing 0xA32C is to enable writing.
10	WDTINTSTS	WDT Interrupt condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
9	MCLKFAILSTS	MCLK Fail condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
8	LVDSTS	LVI condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
2	WDTINTEN	WDT Interrupt condition enable for NMI interrupt
		0 Disable
		1 Enable
1	MCLKFAILEN	MCLK Fail condition enable for NMI interrupt
		0 Disable
		1 Enable
0	LVDEN	LVI Fail condition enable for NMI interrupt
		0 Disable
		1 Enable

4.5.21 SCU_COR: clock output register

The A31G21x series can drive the clock from internal MCLK clock with dedicated post divider. To use CLKO output function, it should be set as CLKO that has output mode in PF4 Pin Mux. Clock Output Register is 8-bit register.

SCU_COR=0x4000_0050

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	CLKODIV
-				0	1111
-				RW	RW

4	CLKOEN	Clock output enable
		0 CLKO is disabled and stay "L" output
		1 CLKO Is enabled
3 0	CLKODIV	Clock output divider value CLKO = MCLK (CLKODIV = 0) $\mathrm{CLKO} = \frac{MCLK}{2 * (\mathrm{CLKODIV} + 1)} \quad (CLKODIV > 0)$

4.5.22 SCU_PLLCON: PLL control register

Integrated PLL will synthesize high speed clock for extremely high performance of the CPU. The PLL controlled by register setting.

SCU_PLLCON=0x4000_0060

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
LOCKSTS	Reserved	PLLRSTB	PLLEN	BYPASSB	PLLMODE	Reserved	PREDIV	POSTRIV1	POSTDIV2	OUTDIV
0	-	0	0	0	0	-	000	00000000	0000	0000
RO	-	RW	RW	RW	RW	-	RW	RW	RW	RW

31	LOCK	LOCK status
		0 PLL is not locked
		1 PLL is locked
23	PLLRSTB	PLL reset
		0 PLL reset is asserted
		1 PLL reset is negated
22	PLLEN	PLL enable
		0 PLL is disabled
		1 PLL is enabled
21	BYPASSB	FIN bypass
		0 FOUT is bypassed as FIN
		1 FOUT is PLL output
20	PLLMODE	PLL VCO mode
		0 VCO is the same with FOUT
		1 VCO frequency is x2 of FOUT
18 16	PREDIV	FIN predivider I
		0~7 FIN divided by (PREDIV + 1), (FIN/1 ~ FIN/8)
15 8	POSTDIV1	Feedback control 1 (N1)
		0x00 N1 = 0 (N1 + 1)
		0xFF N1 = 255 (N1 + 1)
7 4	POSTDIV2	Feedback control 1 (N2)
		0x0 N2 = 0 (N2 + 1)
		0xF N2 = 15 (N2 + 1)
3 0	OUTDIV	output divider control (P)
		0x0 P = 0 (P+1)
		0xF P = 15 (P+1)

NOTES:

1. Bit 20 of PLLCON must be kept at 0.
2. When PLLEN is set to '1', PLLRSTB is set to '1' and PLL Divider count after at least 1us.
3. Wait more than 190us for PLL Lock Time, then check the PLL Lock flag(Bit 31).
4. At power down(Stop/Standby) mode, set PLLRSTB to '0', PLLEN to '0'
5. Bits of OUTDIV is not recommended to set to '1'

$$F_{out} = \frac{F_{IN} * (N_1 + 1)}{(R + 1) * (N_2 + 1) * (P + 1)} * (D + 1)$$

Symbol	Description
R	Pre Divider Counter Value
N_1	Post Divider1 Counter Value
N_2	Post Divider2 Counter Value
P	Output Divider Counter Value
D	Frequency Doubler

Calculating the PLL output frequency value

The PLL of the A31G21x series can accurately set the output frequency, F_{OUT}, in 1MHz increments. The formula for the F_{IN} input to the F_{VCO} input of the PLL is as follows, and the input range of the F_{IN} frequency is between 1MHz and 3MHz. However, it is recommended to set the input frequency (FIN) to 2MHz as much as possible.

$$\mathrm{FIN} = \frac{\mathrm{PLLINCLK}}{(\mathrm{R}+1)}, \quad \text{Where } 1\mathrm{MHz} \le \mathrm{FIN} \le 3\mathrm{MHz}\ (\text{Recommanded FIN} = 2\mathrm{MHz})$$

At this time, the range of F_{VCO} output frequency should be set to 200MHz or less, and the calculation formula is as follows.

$$\mathrm{F_{VCO}} = \mathrm{FIN} \times (\mathrm{N_1}+1), \quad \mathrm{VCO} \le 200\mathrm{MHz}\ if\ \mathrm{D} = 0$$

PLLCON also supports the Doubler function which can double the F_{VCO} output through the bit setting of VCOMODE. When using this doubler function, the output of F_{VCOx2} should be set to 250MHz or less.

$$V_{VCOx2} = \mathrm{VCO} \times (\mathrm{D}+1), \quad \mathrm{VCOx2} \le 250\mathrm{MHz}\ if\ \mathrm{D} = 1$$

As a result, the final frequency of PLL, F_{OUT}, can be obtained from the formula below using the formula above.

$$\mathrm{F_{OUT}} = \frac{\mathrm{PLLINCLK} \times (\mathrm{N_1}+1)}{(\mathrm{R}+1) \times (\mathrm{N_2}+1) \times (\mathrm{P}+1)} \times (\mathrm{D}+1) = \frac{\mathrm{FIN} \times (\mathrm{N_1}+1)}{(\mathrm{N_2}+1) \times (\mathrm{P}+1)} \times (\mathrm{D}+1)$$

4.5.23 SCU_VDCCON: VDC control register

On chip VDC control register. VDCTRIM is used for the trim value of VDC output. VDCWDLY value can be written with writing '1' to VDCWDLY_WEN bit simultaneously.

SCU_VDCCON=0x4000_0064

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	VDC15 WTIDKY	VDC15_PDBGR	Reserved	VDC15_STOP	VDC15_IDLE	VDC15_LOCK	Reserved	VDCWDLY_WEN	VDCWDLY
-	0000	0	-	0	0	0	-	0	0x7F
-	WO	RW	-	RW	RW	RW	-	WO	RW

23 20	VDC15 WTIDKY	VDC15 Write Identification Key On writes, write 0x5 to these bits, otherwise the write is ignored.
19	VDC15_PDBGR	VDC15 1.2V BGR / 1.0V Buffer Power-down Signal *In BGR on → off, VDCLOCK_I should be 0
		0 BGR/Buffer ON (RUN/IDLE/STOP1)
		1 BGR/Buffer OFF (STOP2)
17	VDC15_STOP	VDC15 STOP Mode Control Signal
		0 STOP1 Mode
		1 STOP2 Mode
16	VDC15_IDLE	VDC15 IDLE Mode Control Signal
		0 no IDLE Mode (RUN or STOP1 or STOP2 Mode)
		1 IDLE Mode
15	VDC15_LOCK	VDC15 VDCLOCK Control Signal for *BGR Stabilization *In BGR off → on Sequence, VDCLOCK_I should be 0 during 120usec
		0 VDC Using BMR Reference Voltage
		1 VDC Using BGR Reference Voltage
8	VDCWDLY_WEN	VDCWDLY value write enable. VDCWDLY value can be written with writing '1' to VDCWDLY_WEN bit simultaneously.
7 0	VDCWDLY	VDC warm-up delay count value. When SCU is waked up from POWER-DOWN mode, the warm-up delay is inserted for VDC output being stabilized. The amount of delay can be defined with this register value 7F : 4msec

4.5.24 SCU_LSICON: low speed internal OSC control register

SCU_LSICON controls the internal LDO and Level shifter using for LSI operation.

SCU_ROSCCON=0x4000_006C

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				EN_LDO	SKIP_LS
-				0	1
-				RW	RW

1	EN_LDO	Internal LDO On/Off	
		0	Disable
		1	Enable
0	SKIP_LS	Internal Level Shifter control signal	
		0	Enable
		1	Disable

4.5.25 SCU_EOSCR: external oscillator control register

External main crystal oscillator has two characteristics. For the noise immunity, NMOS amp type is recommended and for the low power characteristic, INV amp type is recommended. This register is 16-bit register.

SCU_EOSCR=0x4000_0080

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		ESEN	Reserved	ESISEL	Reserved	ESNCBYPS	EMEN	Reserved	ISEL	NCOPT	Reserved	NCSKIP
-		0		01	-	0	0	-	01	01	-	0
-		WO		RW	-	RW	WO	-	RW	RW	-	RW

15	ESEN	Write enable for External LSE	
		0	Write access disabled
		1	Write access enabled
13 12	ESISEL	Select current for External LSE	
		00	1.57uA
		01	1.79uA
		10	1.93uA
		11	2.04uA
8	ESNCBYPS	Noise Cancel Bypass enable for External LSE	
		0	Disable
		1	Enable (Noise Cancel bypassed)
7	EMEN	Write enable for External HSE	
		0	Write access disabled
		1	Write access enabled
5 4	ISE	Select current for External HSE	
		00	150uA
		01	300uA
		10	450uA
		11	600uA
3 2	NCOPT	Noise Cancel delay Option for External HSE	
		00	25ns (12MHz < HSE < 16MHz)
		01	20ns (8MHz < HSE < 12MHz)
		10	15ns (4MHz <HSE < 8MHz)
		11	10ns (1MHz <HSE < 4MHz)
0	NCSKIP	Noise Cancel SKIP enable for External HSE	
		0	Disable
		1	Enable (Noise Cancel skipped)

4.5.26 SCU_EMODR: external mode status register

External Mode Status Register shows external mode pin status while booting.

This register is 8-bit register.

SCU_EMODR=0x4000_0084

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																															BOOT
-																															0
-																															RO

0	BOOT	BOOT pin level	
		0	BOOT (PB3) pin is low
		1	BOOT (PB3) pin is high

4.5.27 SCU_RSTDBCR: pin reset debounce control register

Pin Reset Debounce Control Register.

SCU_RSTDBCR=0x4000_0088

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		Reserved	CLKCNT	Reserved	EN
0x0000		-	00_0000	-	0
WO		-	RW	-	RW

31	WTIDKY	Write Identification Key	
16		On writes, write 0x0514 to these bits, otherwise the write is ignored.	
13	CLKCNT	Noise Cancel delay Option for External HSE	
8		N	N clock checking for debounce by LSI (500KHz)
0	EN	Pin reset debounce enable bit	
		0	Disable
		1	Enable

NOTE: If user want to operate pin reset debounce, user must enable LSI (500KHz). Because pin reset debounce use LSI for clock source.

4.5.28 SCU_MCCR1: miscellaneous clock control register 1

The A31G21x series can drive the clock from internal MCLK clock with dedicated post divider. STCSEL bits and STCDIV bits of SCU_MCCR1 are used as SYSTICK external clock source. TEXT1CSEL bits and TEXT1DIV bits of SCU_MCCR1 are used as TIMER1n external clock source. This register is 32-bit register.

SCU_MCCR1=0x4000_0090

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	TEXT1CSEL	TEXT1DIV	Reserved	STCSEL	SYSTICKDIV
-	0x0	0x00	-	0x0	0x00
-	RW	RW	-	RW	RW

26 24	TEXT1CSEL	TIMER1n EXT Clock source select bit 0xx LSI 500KHz 011 LSE(32KHz) 100 MCLK(bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
23 16	TEXT1DIV	TIMER1n EXT Clock N divider 0x00 disabled 0xN (selected clock)/N To change the value, set 0x0 first without changing TEXT1CSEL
10 8	STCSEL	SYSTIC Clock source select bit 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
7 0	STDIV	SYSTICK Clock N divider 0x00 disabled 0xN (selected clock)/N To change the value, set 0x0 first without changing STCSEL

4.5.29 SCU_MCCR2: miscellaneous clock control register 2

The A31G21x series can drive the clock from internal MCLK clock with dedicated post divider. TEXT2CSEL bits and TEXT2DIV bits of SCU_MCCR2 are used as TIMER20 external clock source. TEXT3CSEL bits and TEXT3DIV bits of SCU_MCCR2 are used as TIMER30 external clock source. This register is 32-bit register.

SCU_MCCR2=0x4000_0094

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	TEXT3CSEL	TEXT3DIV	Reserved	TEXT2CSEL	TEXT2DIV
-	0x0	0x00	-	0x0	0x00
-	RW	RW	-	RW	RW

26 24	TEXT3CSEL	TIMER 30 EXT Clock source select bit 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
23 16	TEXT3DIV	TIMER 30 EXT Clock N divider 0x00 disabled 0xN (selected clock) / N To change the value, set 0x0 first without changing TEXT3DIV
10 8	TEXT2CSEL	TIMER 20 EXT Clock source select bit 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
7 0	TEXT2DIV	TIMER 20 EXT Clock N divider 0x00 disabled 0xN (selected clock) / N To change the value, set 0x0 first without changing TEXT2CSEL

4.5.30 SCU_MCCR3: miscellaneous clock control register 3

The A31G21x series can drive the clock from internal MCLK clock with dedicated post divider. WDTCSEL bits and WDTDIV bits of SCU_MCCR3 are used as WDT external clock source. WTEXTCSEL bits and WTEXTCDIV bits of SCU_MCCR3 are used as WT external clock source. This register is 32-bit register.

SCU_MCCR3=0x4000_0098

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	WTEXTCSEL	WTEXTCDIV	Reserved	WDTCSEL	WDTDIV
-	0x0	0x00	-	0x0	0x00
-	RW	RW	-	RW	RW

26 24	WTEXTCSEL	WT External Clock source select bit 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
23 16	WTEXTCDIV	WT External Clock N divider 0x00 disabled 0xN (selected clock) / N To change the value, set 0x0 first without changing WTEXTCSEL
10 8	WDTCSEL	WDT Clock source select bit 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
7 0	WDTDIV	WDT Clock N divider 0x00 disabled 0xN (selected clock) / N To change the value, set 0x0 first without changing WDTCSEL

4.5.31 SCU_MCCR4: miscellaneous clock control register 4

The A31G21x series can drive the clock from internal MCLK clock with dedicated post divider. PD0CSEL bits and PD0DIV bits of SCU_MCCR4 are used as PA-/ PB-/ PC-Debounce Clock source. PD1CSEL bits and PD1DIV bits of SCU_MCCR4 are used as PD-/ PE-/ PF-Debounce Clock source. This register is 32-bit register.

SCU_MCCR4=0x4000_009C

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	PD1CSEL	PD1DIV	Reserved	PD0CSEL	PD0DIV
-	0x0	0x00	-	0x0	0x00
-	RW	RW	-	RW	RW

26 24	PD1CSEL	Debounce Clock for PORT source select bit (PD,PE,PF) 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
23 16	PD1DIV	PORT Debounce Clock N divider (PD,PE,PF) 0x00 Disabled 0xN (selected clock) / N To change the value, set 0x0 first without changing PD1CSEL Note)Clockisnotactivatedduring PD1DIVbit is disabled.
10 8	PD0CSEL	Debounce Clock for PORT source select bit (PA,PB,PC) 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
7 0	PD0DIV	PORT Debounce Clock N divider (PA,PB,PC) 0x00 disabled 0xN (selected clock) / N To change the value, set 0x0 first without changing PD0CSEL Note)Clockisnotactivatedduring PD0DIVbit is disabled.

4.5.32 SCU_MCCR5: miscellaneous clock control register 5

The A31G21x series can drive the clock from internal MCLK clock with dedicated post divider. LEDCSEL bits and LEDDIV bits of SCU_MCCR5 are used as LED Clock source.

SCU_MCCR5=0x4000_00A0

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			LEDCSEL	LEDDIV
-			0x0	0x00
-			RW	RW

10 8	LEDCSEL	LED Clock source select bit 0xx LSI 500KHz 011 LSE (32KHz) 100 MCLK (bus clock) 101 HSI 32MHz 110 HSE 111 PLL Clock
7 0	LEDDIV	LED Clock N divider 0x00 Disabled 0xN (selected clock) / N To change the value, set 0x0 first without changing LEDCSEL

4.5.33 SCULV_LVICR: low voltage indicator control register

Low Voltage Indicator Control Register is 32-bit register. This register is able to 32/16/8-bit access.

SCULV_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
-			0	0	0	0	0000
-			RW	RW	RW	RW	RW

7	LVIEN	LVI Enable bit.	
		0	Disable low voltage indicator.
		1	Enable low voltage indicator.
5	LVINTEN	LVI Interrupt Enable bit.	
		0	Disable low voltage indicator interrupt.
		1	Enable low voltage indicator interrupt.
4	LVIFLAG	LVI Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
3 0	LVIVS	LVI Voltage Selection bits.	
		0000	1.60V
		0001	1.69V
		0010	1.78V
		0011	1.90V
		0100	1.99V
		0101	2.12V
		0110	2.30V
		0111	2.47V
		1000	2.67V
		1001	3.04V
		1010	3.18V
		1011	3.59V
		1100	3.72V
		1101	4.03V
		1110	4.20V
		1111	4.48V

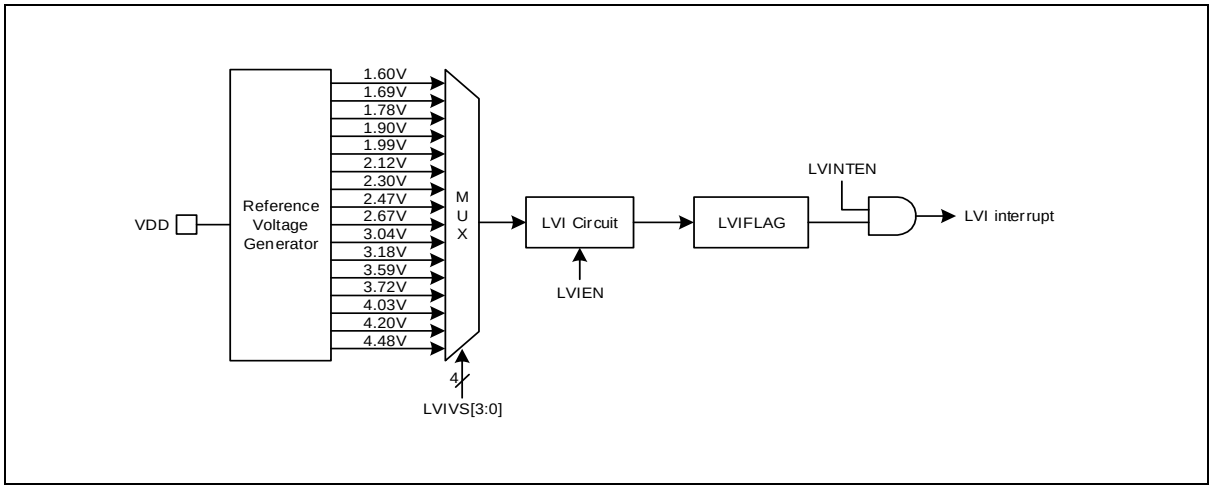


Figure 22. LVI Block Diagram

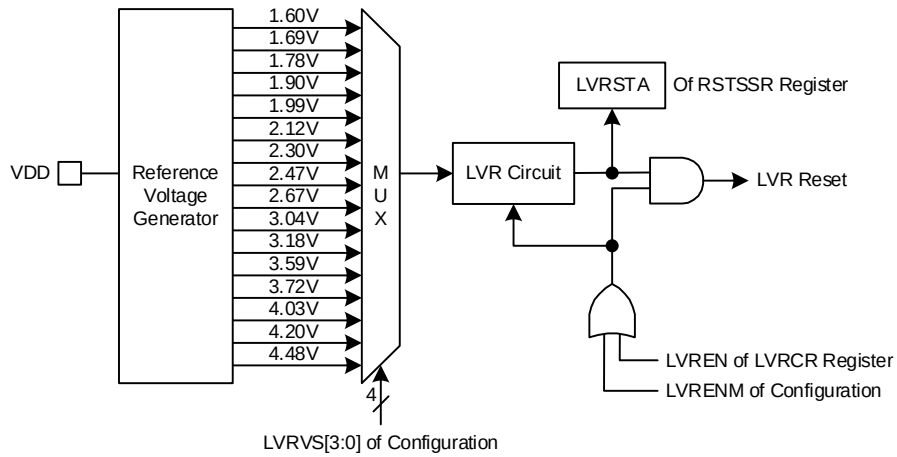
4.5.34 SCULV_LVRCR: low voltage reset control register

Low Voltage Reset Control Register is 32-bit register. This register is able to 32/16/8-bit access.

SCULV_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
-			0x00
-			RW

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



NOTE: The voltage level of LVR is set by the low voltage reset configuration register (SCULV_LVRCNFG).

Figure 23. LVR Block Diagram

4.5.35 SCULV_LVRCNFIG: configuration for low voltage reset

Low Voltage Indicator Control Register is 32-bit register. This register is able to 32/16/8-bit access.

SCULV_LVICR=0x4000_5108

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		LVRENM	Reserved	LVRVS
0x0000		0x0000	-	0xF
WO		RW	-	RW

31 24	WTIDKY	Write Identification Key On writes, write 0x72A5 to these bits, otherwise the write is ignored.	
15 8	LVRENM	LVR Reset Operation Control Master Configuration	
		0xAA	LVR operation is decided by the LVREN of SCULV_LVRCR register
		Others	Master enable LVR operation
3 0	LVRVS	LVR Voltage Selection bits.	
		1111	1.60V
		1110	1.69V
		1101	1.78V
		1100	1.90V
		1011	1.99V
		1010	2.12V
		1001	2.30V
		1000	2.47V
		0111	2.67V
		0110	3.04V
		0101	3.18V
		0100	3.59V
		0011	3.72V
		0010	4.03V
		0001	4.20V
		0000	4.48V

5 PCU and GPIO

Port Control Unit (PCU) configures and controls external I/Os as listed in the followings:

- External signal directions of each pins
- Interrupt trigger mode for each pins
- Internal pull-up/down register control and open drain control

General Purpose Input/Output (GPIO) is corresponding to the most pins except dedicated-function pins. GPIO ports are controlled by GPIO block as listed in the followings:

- Output signal level (H/L) select
- External interrupt interface
- Pull up/down enable or disable

Four pins in Table 16 are assigned for PCU and GPIO blocks.

Table 16. PCU and GPIO Pins

Pin name	Type	Description
PA	IO	PA0 to PA7
PB	IO	PB0 to PB7
PC	IO	PC0 to PC5
PD	IO	PD0 to PD5
PF	IO	PF0 to PF7

5.1 PCU and GPIO block diagram

Figure 24 describes PCU in block diagram.

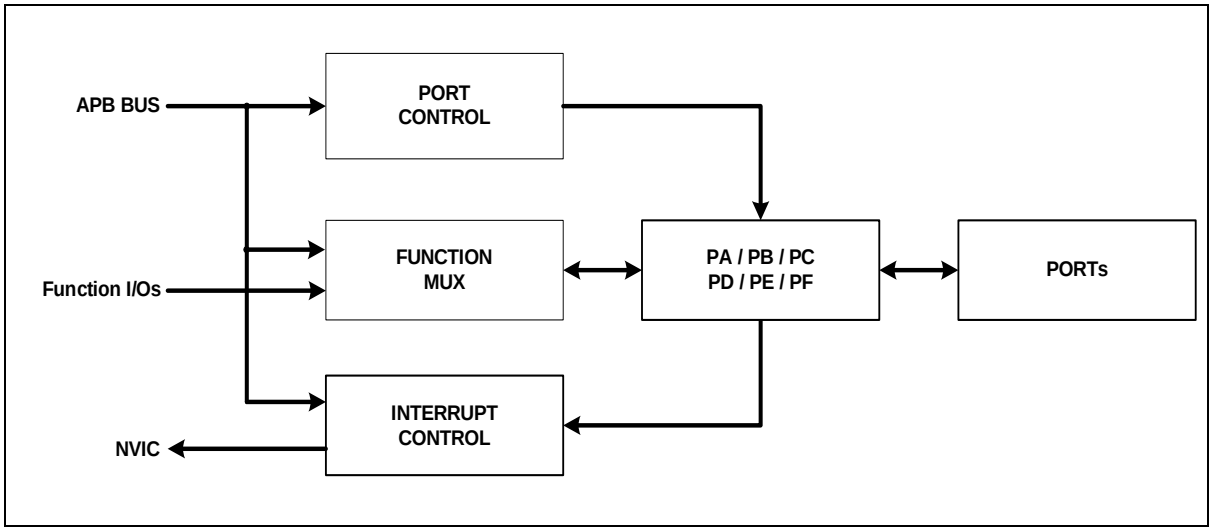


Figure 24. PCU Block Diagram

Figure 25 describes GPIO in block diagram.

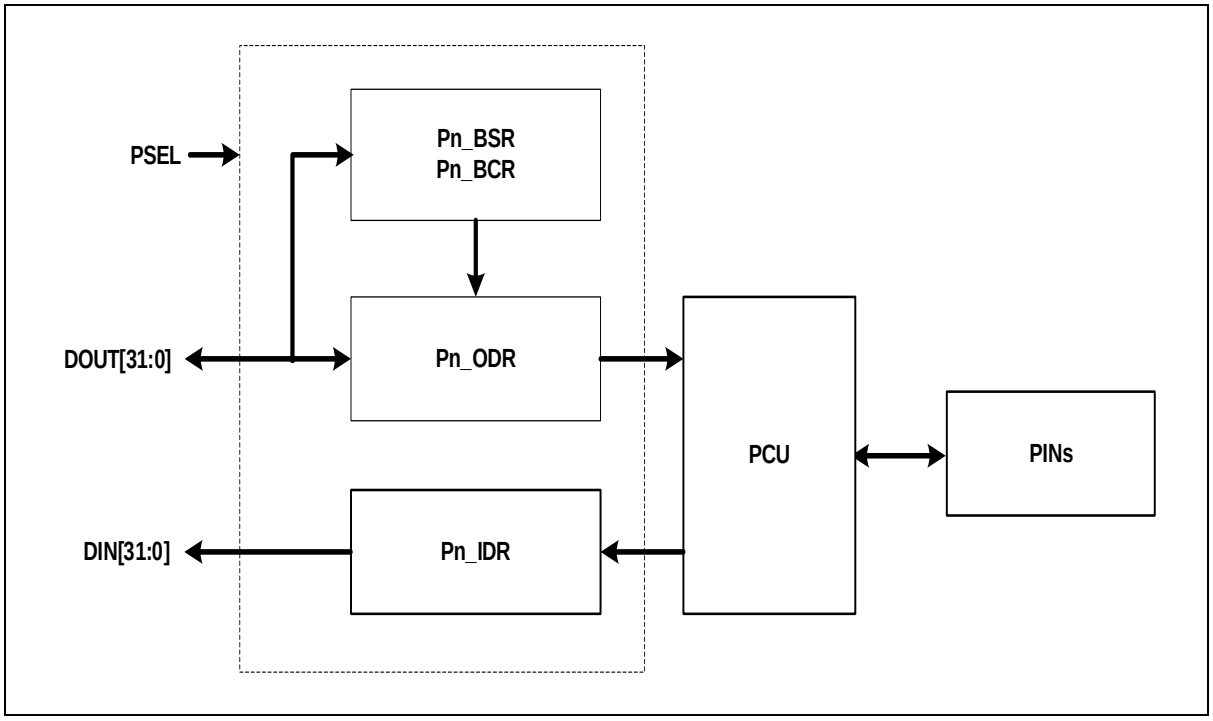


Figure 25. GPIO Block Diagram

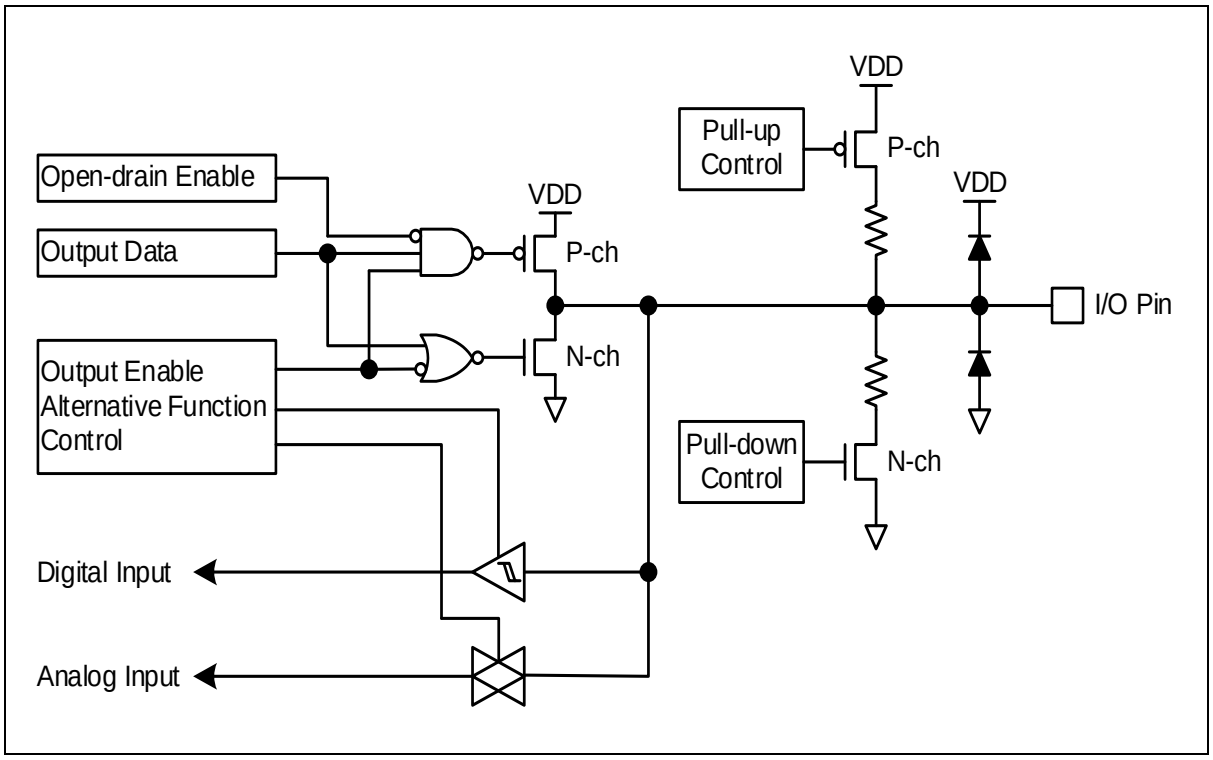


Figure 26. I/O Port Block Diagram (General I/O Pins)

Figure 27 introduces external interrupt I/O pins.

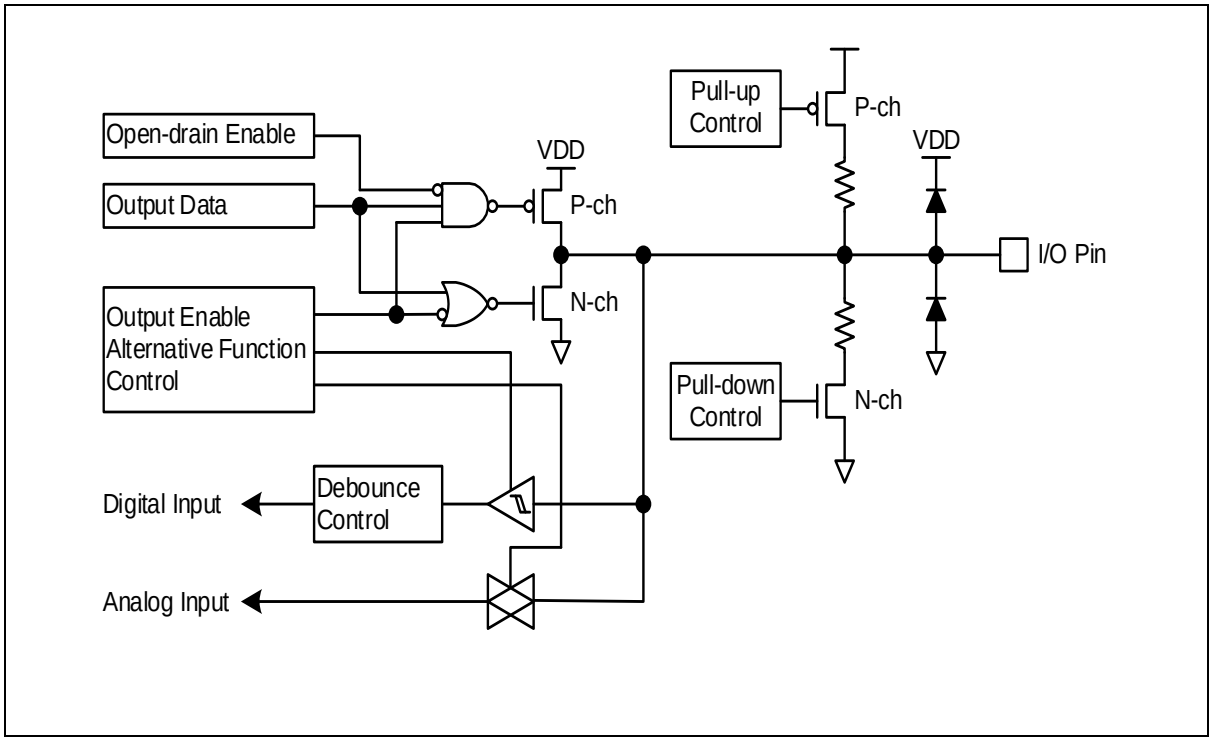


Figure 27. I/O Port Block Diagram (External Interrupt I/O Pins)

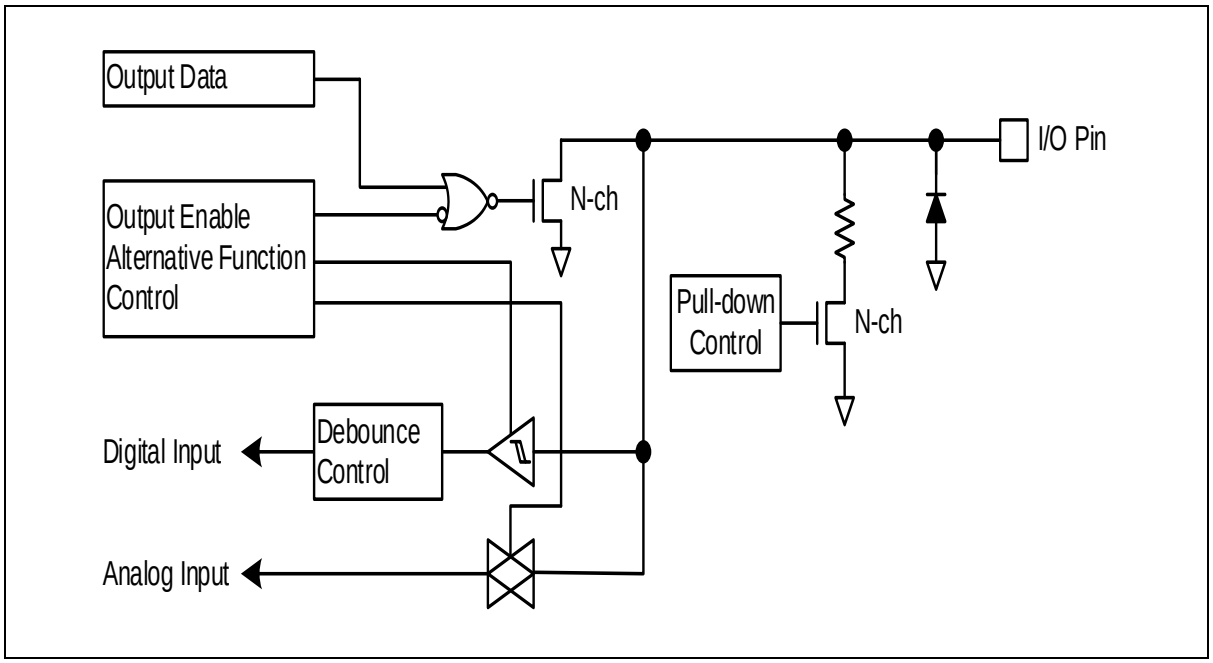


Figure 28. I/O Port Block Diagram (PF5, PF6, PF7 I/O pins)

5.2 Pin multiplexing

GPIO pins have alternative function pins. Table 17 shows pin multiplexing information.

Table 17. GPIO Alternative Function

Port	Pin	Functions				
		AF0	**AF1**	**AF2**	**AF3**	**AF4**
PA	0		SDA1		AN0	CS3
	1		SCL1		AN1	CS4
	2		EC12		AN2/AVREF	CS5/COM15
	3				AN3	CS6/COM14
	4				AN4	CS7/COM13
	5		T12O	T12C	AN5	CS0/COM12
	6		T11O	T11C	AN6	CS1/COM11
	7		T13O	T13C	AN7/ DAO	CS2/COM10
PB	0		TXD10	MOSI10	AN8	CS8/COM9
	1		RXD10	MISO10	AN9	CS9/COM8
	2		EC13	SCK10	AN10	CS10/COM7
	3		BOOT	SS10		
	4		TXD0	SWCLK		
	5		RXD0	SWDIO		
	6		TXD1	EC11	AN11	CS17
	7		RXD1		AN12	CS18
PC	0		T20O	T20C	AN13	CS19/COM6
	1		T21O	T21C		CS20/COM5
	2		EC20	MOSI20		CS21/COM4
	3		EC21	MISO20		CS22/COM3
	4			SCK20		CS23/COM2
	5	nRESET				
PD	0		SCL0	SS20		COM1
	1		SDA0	EC10		COM0
	2		TXD11	MOSI11		SEG9
	3		RXD11	MISO11		SEG8
	4		BLNK	SCK11		SEG7
	5			SS11		SEG6

Table 17. GPIO Alternative Function (continued)

Port	Pin	Functions				
		AF0	AF1	AF2	AF3	AF4
PE	0		PWM30AA			
	1		PWM30AB			
	2		PWM30BA	SS21		CS16/SEG0
	3		PWM30BB	SCK21		CS15/SEG1
	4		PWM30CA	MISO21		CS14/SEG2
	5		PWM30CB	MOSI21		CS13/SEG3
	6		T10O	T10C		CS12/SEG4
	7		T11O	T11C		CS11/SEG5
PF	0		SCL1		XOUT	
	1		SDA1		XIN	
	2		TXD1	EC30	SXIN	
	3		RXD1	T30C	SXOUT	
	4		CLKO			
	5		BLNK			
	6		EC30	SCL0		
	7		T30C	SDA0		

NOTE: On connection with debugger host, The SWCLK and SWDIO pins are always for SW-DP pins. So, the corresponding bits of PB_MOD/PB_TYP/PB_AFSR/PB_PUPD registers may not be written by software.

5.3 Registers

Base address of PCU is introduced in the followings:

Table 18. Base Address of PCU

Name	Base address	Description
PA	0x4000_1000	General Port A
PB	0x4000_1100	General Port B
PC	0x4000_1200	General Port C
PD	0x4000_1300	General Port D
PE	0x4000_1400	General Port E
PF	0x4000_1500	General Port F

Table 19. PCU and GPIO Register Map

Name	Offset	Type	Description	Reset value	Ref.
Pn_MOD	0x00	RW	Port n Mode Register	0x0000_XXXX	5.3.1
Pn_TYP	0x04	RW	Port n Output Type Selection Register	0x0000_0000	5.3.2
Pn_AFSR1	0x08	RW	Port n Alternative Function Selection Register	0xXXXX_XXXX	5.3.3
Pn_PUPD	0x10	RW	Port n Pull-up/down Resistor Selection Register	0x0000_XXXX	5.3.4
Pn_INDR	0x14	RO	Port n Input Data Register	0x0000_XXXX	5.3.5
Pn_OUTDR	0x18	RW	Port n Output Data Register	0x0000_0000	5.3.6
Pn_BSR	0x1C	WO	Port n Output Bit Set Register	0x0000_0000	5.3.7
Pn_BCR	0x20	WO	Port n Output Bit Clear Register	0x0000_0000	5.3.8
Pn_OUTDMSK	0x24	RW	Port n Output Data Mask Register	0x0000_0000	5.3.9
Pn_DBCR	0x28	RW	Port n Debounce Control Register	0x0000_0000	5.3.10
Pn_IER	0x2C	RW	Port n interrupt enable register	0x0000_0000	5.3.11

Table 19. PCU and GPIO Register Map (continued)

Name	Offset	Type	Description	Reset value	Ref.
Pn_ISR	0x30	RC	Port n interrupt status register	0x0000_0000	5.3.12
Pn_ICR	0x34	RW	Port n interrupt control register	0x0000_0000	5.3.13
PF_PLSR	0x38	RW	Port F level select register	0x0000_0000	5.314
PCU_SPI2PCON	0x4000_1540	RW	SPI20, 21 Port control register	0x0000_0000	5.3.15
PCU_KEYR	0x4000_1F00	RW	Port LED KEY Register	0x0000_0000	5.3.16
PCU_SEGR	0x4000_1F04	RW	Port LED SEG Register	0x0000_0000	5.3.17
PCU_COM	0x4000_1F08	RW	Port LED COM Register	0x0001_0000	5.3.18
PCU_PORTEN	0x4000_1FF0	WO	Port Access Enable	0x0000_0000	5.3.19
PCU_SPI2PCON	0x4000_1540	RW	SPI20, 21 Port control register	0x0000_0000	5.3.15
PCU_KEYR	0x4000_1F00	RW	Port LED KEY Register	0x0000_0000	5.3.16
PCU_SEGR	0x4000_1F04	RW	Port LED SEG Register	0x0000_0000	5.3.17
PCU_COM	0x4000_1F08	RW	Port LED COM Register	0x0001_0000	5.3.18
PCU_PORTEN	0x4000_1FF0	WO	Port Access Enable	0x0000_0000	5.3.19

NOTES:

1. Where n = A, B, C, D, E, and F.
2. For exception, the reset value of PB_MOD, PB_AFSR1, PB_PUPD register is 0x0000_FAB3, 0x0022_1000, 0x0000_0540 respectively.
3. For exception, the reset value of PC_MOD, PC_PUPD register is 0x0000_0BFF, 0x0000_0400 respectively.
4. For exception, the reset value of PD_MOD, PD_AFSR1, PC_PUPD register is 0x0000_0FF0, 0x0000_0005 respectively.

5.3.1 Pn_MOD: port n mode register

Input or output control of each port pin. Each pin can be configured as an input pin, an output pin or an alternative function pin. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A to F)

PA_MOD=0x4000_1000, PB_MOD=0x4000_1100, PC_MOD=0x4000_1200
PD_MOD=0x4000_1300, PE_MOD=0x4000_1400, PF_MOD=0x4000_1500

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		MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
-		11	11	11	11	11	11	11	11
-		RW	RW	RW	RW	RW	RW	RW	RW

2x+1	MODEx	Port n Mode Selection bits, x:0 to 7	
2x		00	Input mode
		01	Output mode
		10	Alternative function mode
		11	Analog input mode

NOTE: For exception, the reset value of PB_MOD, PC_MOD, PD_MOD register is 0xFAB3, 0x0BFF, 0x0FF0 respectively.

5.3.2 Pn_TYP: port n output type selection register

Pn_TYP selects control option from a Push-pull output and Open-drain output for each port pin. Size of this register is 32-bit, and it is able to do 32/16/8-bit access (n = A to F)

PA_TYP=0x4000_1004, PB_TYP=0x4000_1104, PC_TYP=0x4000_1204
PD_TYP=0x4000_1304, PE_TYP=0x4000_1404, PF_TYP=0x4000_1504

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			TYP7	TYP6	TYP5	TYP4	TYP3	TYP2	TYP1	TYP0
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

x	TYPx	Port n Output Type Selection bits, x:0 to 7	
		0	Push-pull output
		1	Open-drain output

5.3.3 Pn_AFSR1: port n alternative function selection register 0

Pn port Alternative Function select register. This register must be set properly before use the port. Otherwise the port can't guarantee its functionality.

Port n Alternative Function Selection Register 0 is 32-bit register. This Register is able to 32/16/8-bit access.

PA_AFSR1=0x4000_1008, PB_AFSR1=0x4000_1108, PC_AFSR1=0x4000_1208
PD_AFSR1=0x4000_1308, PE_AFSR1=0x4000_1408, PF_AFSR1=0x4000_1508

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
AFSB7	AFSB6	AFSB5	AFSB4	AFSB3	AFSB2	AFSB1	AFSB0
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4x+3	AFSBx	Port n Alternative Function Selection bits, x:0 to 7	
4x		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)
		Others	Reserved

NOTES:

1. The AFSR1 bits for PF0 – PF3 won't be changed during the corresponding clock is selected as the system clock (MCLK).
2. For exception, the reset value of PB_AFSR1 register is 0x00221000 respectively.

5.3.4 Pn_PUPD: port n pull-up/down resistor selection register

Every pin in the port has on-chip pull-up/down resistors which can be configured by Pn_PUPD registers.

Port n Pull-up/down Resistor Selection Register is 32-bit register. This Register is able to 32/16/8-bit access. (n = A to F)

PA_PUPD=0x4000_1010, PB_PUPD=0x4000_1110, PC_PUPD=0x4000_1210
PD_PUPD=0x4000_1310, PE_PUPD=0x4000_1410, PF_PUPD=0x4000_1510

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		PUPD7	PUPD6	PUPD5	PUPD4	PUPD3	PUPD2	PUPD1	PUPD0
-		00	00	00	00	00	00	00	00
-		RW	RW	RW	RW	RW	RW	RW	RW

2x+1	PUPDx	Port n Pull-up/down Resistor Selection bits, x:0 to 7
2x		00 Disable pull-up/down resistor
		01 Enable pull-up resistor
		10 Enable pull-down resistor
		11 Analog input(ADC/OSC)

NOTES:

1. The pull-up/down resistor of PF0 – PF3 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. For exception, the reset value of PB_PUPD, PC_ PUPD, PD_ PUPD register is 0x0540, 0x0400, 0x0005 respectively.

5.3.5 Pn_INDR: port n input data register

Each pin level status can be read in the Pn_INDR register. Even if the pin is alternative mode except analog mode and output in alternative mode, the pin level can be detected in the Pn_INDR register.

Port n Input Data Register is 32-bit register. This Register is able to 32/16/8-bit access. (n = A to F).

PA_INDR=0x4000_1014, PB_INDR=0x4000_1114, PC_INDR=0x4000_1214
PD_INDR=0x4000_1314, PE_INDR=0x4000_1414, PF_INDR=0x4000_1514

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			INDR7	INDR6	INDR5	INDR4	INDR3	INDR2	INDR1	INDR0
-			x	x	x	x	x	x	x	x
-			RO	RO	RO	RO	RO	RO	RO	RO

x	INDRx	Port n Input Data bit, x:0 to 7

5.3.6 Pn_OUTDR: port n output data register

When the pin is set as output and GPIO mode, the pin output level is defined by Pn_OUTDR registers.

Port n Output Data Register is 32-bit register. This Register is able to 32/16/8-bit access. (n = A to F)

PA_OUTDR=0x4000_1018, PB_OUTDR=0x4000_1118, PC_OUTDR=0x4000_1218
PD_OUTDR=0x4000_1318, PE_OUTDR=0x4000_1418, PF_OUTDR=0x4000_1518

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																								OUTDR7	OUTDR6	OUTDR5	OUTDR4	OUTDR3	OUTDR2	OUTDR1	OUTDR0
-																								0	0	0	0	0	0	0	0
-																								RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDRx	Port n Output Data bit, x:0 to 7 The OUTDR bits can be individually set/cleared by writing to the PnBSR/PnBCR register

5.3.7 Pn_BSR: port n output bit set register

Pn_BSR is a register for control each bit of Pn_OUTDR register. Writing a '1' into the specific bit will set a corresponding bit of Pn_OUTDR to '1'. Writing '0' in this register has no effect.

Port n Output Bit Set Register is 32-bit register. This Register is able to 32/16/8-bit access. (n = A to F)

PA_BSR=0x4000_101C, PB_BSR=0x4000_111C, PC_BSR=0x4000_121C
PD_BSR=0x4000_131C, PE_BSR=0x4000_141C, PF_BSR=0x4000_151C

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																								BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0
-																								0	0	0	0	0	0	0	0
-																								WO	WO	WO	WO	WO	WO	WO	WO

x	BSRx	Port n Output Set bit, x: 0 to 7. These bits are always read to 0x00	
		0	No effect
		1	Set the corresponding OUTDRx bit (automatically cleared to 0)

5.3.8 Pn_BCR: port n output bit clear register

Pn_BCR is a register for control each bit of Pn_OUTDR register. Writing a '1' into the specific bit will set a corresponding bit of PnOUTDR to '0'. Writing '0' in this register has no effect.

Port n Output Bit Clear Register is 32-bit register. This Register is able to 32/16/8-bit access.

(n = A to F).

PA_BCR=0x4000_1020, PB_BCR=0x4000_1120, PC_BCR=0x4000_1220
PD_BCR=0x4000_1320, PE_BCR=0x4000_1420, PF_BCR=0x4000_1520

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																								BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
-																								0	0	0	0	0	0	0	0
-																								WO	WO	WO	WO	WO	WO	WO	WO

x	BCRx	Port n Output Clear bit, x: 0 to 7. These bits are always read to 0x00	
		0	No effect
		1	Clear the corresponding OUTDRx bit (automatically cleared to 0)

5.3.9 Pn_OUTDMSK: port n output data mask register

PnOUTDMSK is a register for protection each bit of PnOUTDR register. Writing a '1' into the specific bit will protect a corresponding bit of PnOUTDR. Writing '0' in this register is unmasked.

Port n Output Data Mask Register is 32-bit register. This Register is able to 32/16/8-bit access.

(n = A to F)

PA_OUTDMSK=0x4000_1024, PB_OUTDMSK=0x4000_1124, PC_OUTDMSK=0x4000_1224
PD_OUTDMSK=0x4000_1324, PE_OUTDMSK=0x4000_1424, PF_OUTDMSK=0x4000_1524

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																								OUTDMSK7	OUTDMSK6	OUTDMSK5	OUTDMSK4	OUTDMSK3	OUTDMSK2	OUTDMSK1	OUTDMSK0
-																								0	0	0	0	0	0	0	0
-																								RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDMSKx	Port n Output Data Mask bit, x: 0 to 7.	
		0	Unmask. The corresponding OUTDR bit can be changed.
		1	Mask. The corresponding OUTDRx bit is protected.

5.3.10 Pn_DBCR: port n debounce control register

Port n Debounce Control Register is 32-bit register. This Register is able to 32/16/8-bit access.

(n = A to F)

PA_DBCR=0x4000_1028, PB_DBCR=0x4000_1128, PC_DBCR=0x4000_1228
PD_DBCR=0x4000_1328, PE_DBCR=0x4000_1428, PF_DBCR=0x4000_1528

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			DBEN7	DBEN6	DBEN5	DBEN4	DBEN3	DBEN2	DBEN1	DBEN0
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

x	DBENx	Port n Debounce Enable bit, x:0 to 7
	0	Disable debounce filter
	1	Enable debounce filter

NOTES:

1. It needs to check clock source when using the Port n Debounce Function.The clock source is selected by SCU_MCCR4 register. (The selected clock should be enabled) LSI, LSE, MCLK, HSI, MOSC and PLL clock are selectable as clock source through this SCU_MCCR4 register.
2. 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.
3. The port debounce should be disabled before deep sleep mode. If it is not disable, wake-up is not occurred

5.3.11 Pn_IER: port n interrupt enable register

The entire pin can be an external interrupt source. Both of edge trigger interrupt and level trigger interrupt are supported. The interrupt mode can be configured by setting Pn_IER registers

PA_IER=0x4000_102C, PB_IER=0x4000_112C, PC_IER=0x4000_122C
PD_IER=0x4000_132C, PE_IER=0x4000_142C, PF_IER=0x4000_152C

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		PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0
-		00	00	00	00	00	00	00	00
-		RW	RW	RW	RW	RW	RW	RW	RW

2x+1	PIEx	Port n Pin interrupt Enable Selection bits, x:0 to 7
2x		
	00	Disable Interrupt
	01	Enable interrupt as level trigger mode
	10	Reserved
	11	Enable interrupt as edge trigger mode

5.3.12 Pn_ISR: port n interrupt status register

When an interrupt is delivered to the CPU, the interrupt status can be detected by reading Pn_ISR register. PnISR register will report a source pin of interrupt and a type of interrupt.

To clear an interrupt occurrence flag, write a 1 to the bit.

PA_ISR=0x4000_1030, PB_ISR=0x4000_1130, PC_ISR=0x4000_1230
PD_ISR=0x4000_1330, PE_ISR=0x4000_1430,PF_ISR=0x4000_1530

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		PIS7	PIS6	PIS5	PIS4	PIS3	PIS2	PIS1	PIS0
-		00	00	00	00	00	00	00	00
-		RC	RC	RC	RC	RC	RC	RC	RC

2x+1	PISx	Port n Pin interrupt Status bits, x:0 to 7
2x		00 No interrupt event
		01 Falling edge interrupt event is present.
		10 Rising edge interrupt event is present.
		11 Both of rising and falling interrupt event is present in edge trigger interrupt mode. Low or High level interrupt event is present in level trigger interrupt mode.

NOTE : RC(read/clear) - Software can read as well as clear this bit by writing 1. Writing '0' has no effect on the

5.3.13 Pn_ICR: port n interrupt control register

Port Pin Interrupt mode control register.

PA_ICR=0x4000_1034, PB_ICR=0x4000_1134, PC_ICR=0x4000_1234
PD_ICR=0x4000_1334, PE_ICR=0x4000_1434, PF_ICR=0x4000_1534

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		PIC7	PIC6	PIC5	PIC4	PIC3	PIC2	PIC1	PIC0
-		00	00	00	00	00	00	00	00
-		RW	RW	RW	RW	RW	RW	RW	RW

2x+1	PICx	Port n Pin interrupt Control bits, x:0 to 7
2x		00 Prohibit external interrupt
		01 Low level interrupt or Falling edge interrupt mode
		10 High level interrupt or Rising edge interrupt mode
		11 Both of rising and falling edge interrupt mode Not support for level trigger interrupt mode.

5.3.14 PF_PLSR: port F level select register

PORT F level select register.

PF_PLSR=0x4000_1538

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				PF7LSB	PF6LSB	PF5LSB
-				00	00	00
-				RW	RW	RW

2	PF7LSB	PORTF 7 Level Select bit.	
		0	VDD level (default)
		1	1.8V level
1	PF6LSB	PORTF 6 Level Select bit.	
		0	VDD level (default)
		1	1.8V level
0	PF5LSB	PORTF 5 Level Select bit.	
		0	VDD level (default)
		1	1.8V level

5.3.15 PCU_SPI2PCON: SPI2n pin re-map register

This register redirects the pin map to use the SPI interface on the SPI2n or SPI21 pins instead of the SPI10 or SPI11 channels on USARTn. If SPI2n pin map redirection of the SPI20 or SPI21 is enable, the SPI interface will operate on the SPI20 or SPI21 channels and will not operate on the USART10 or USART11 channel. On the contrary, if SPI2n pin re-map of the SPI20 or SPI21 is disable, SPI interface will operate on USART10 or USART11 channel.

PCU_SPI2PCON =0x4000_1540

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				SPI21_PCON	SPI20_PCON
-				0	0
-				RW	RW

1	SPI21_PCON	SPI21 Port Select bit	
		0	SPI21< PE[5:2]> PORT Select
		1	SPI11 < PD[5:2]> PORT Select
0	SPI20_PCON	SPI20 Port Select bit	
		0	SPI20 <PC[4:2], PD[0]> PORT Select
		1	SPI10 <PB[3:0]> PORT Select

5.3.16 PCU_KEYR: port LED KEY register

PCU_KEYR is Port LED KEY register

PCU_KEY=0x4000_1F00

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
COM EN KEY		SEG EN KEY	
0x0000		0x0000	
RW		RW	

31	COM EN KEY	Writing 0xA087 in Operation Key register
16		LED COM signal controlled by COMR register.
15	SEG EN KEY	Writing 0x0702 in Operation Key register
0		LED SEG signal controlled by SEGR register.

5.3.17 PCU_SEGR: port LED SEG register

PCU_SEGR is Port LED SEG register

PCU_SEGR=0x4000_1F04

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			SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
-			0	0	0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	SEGx	LED SEGx Signal output control, x:0 to 9	
		0	Disable
		1	Enable

5.3.18 PCU_COMR: port LED COM register

PCU_COMR is Port LED COM register

PCU_COMR=0x4000_1F08

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															Led_endf	COM15	COM14	COM13	COM12	COM11	COM10	COM9	COM8	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
-															1	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

16	Led_endf	Led signal output control	
		0	Enable
		1	Disable
x	COM x	LED COMx Signal output control, x:0 to 15	
		0	Disable
		1	Enable

5.3.19 PCU_PORTEN: port access enable

PCU_PORTEN enables register writing permission of all PCU registers.

PCU_PORTEN=0x4000_1FF0

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																								PORTEN							
-																								-							
-																								WO							

7
0
	PORTEN	Writing the sequence of 0x15 and 0x51 in this register enables writing to PCU registers, and writing other values protects all PCU registers from writing.
		NOTE: How to use PCU_PORTEN
		PCU_PORTEN=0x15;
		PCU_PORTEN=0x51;
		
		// enable PORTEN
		…
		PCU_PORTEN=0; //
		disable PORTEN

5.4 Functional description

If an input function of a certain I/O port is used by Pin Control Register, an output function of the I/O port is disabled. Function of each port can be different in accordance with an Alternative Function Selection Register.

Input Data Register captures current data of the I/O pin or debounced input data at every GPIO clock cycle.

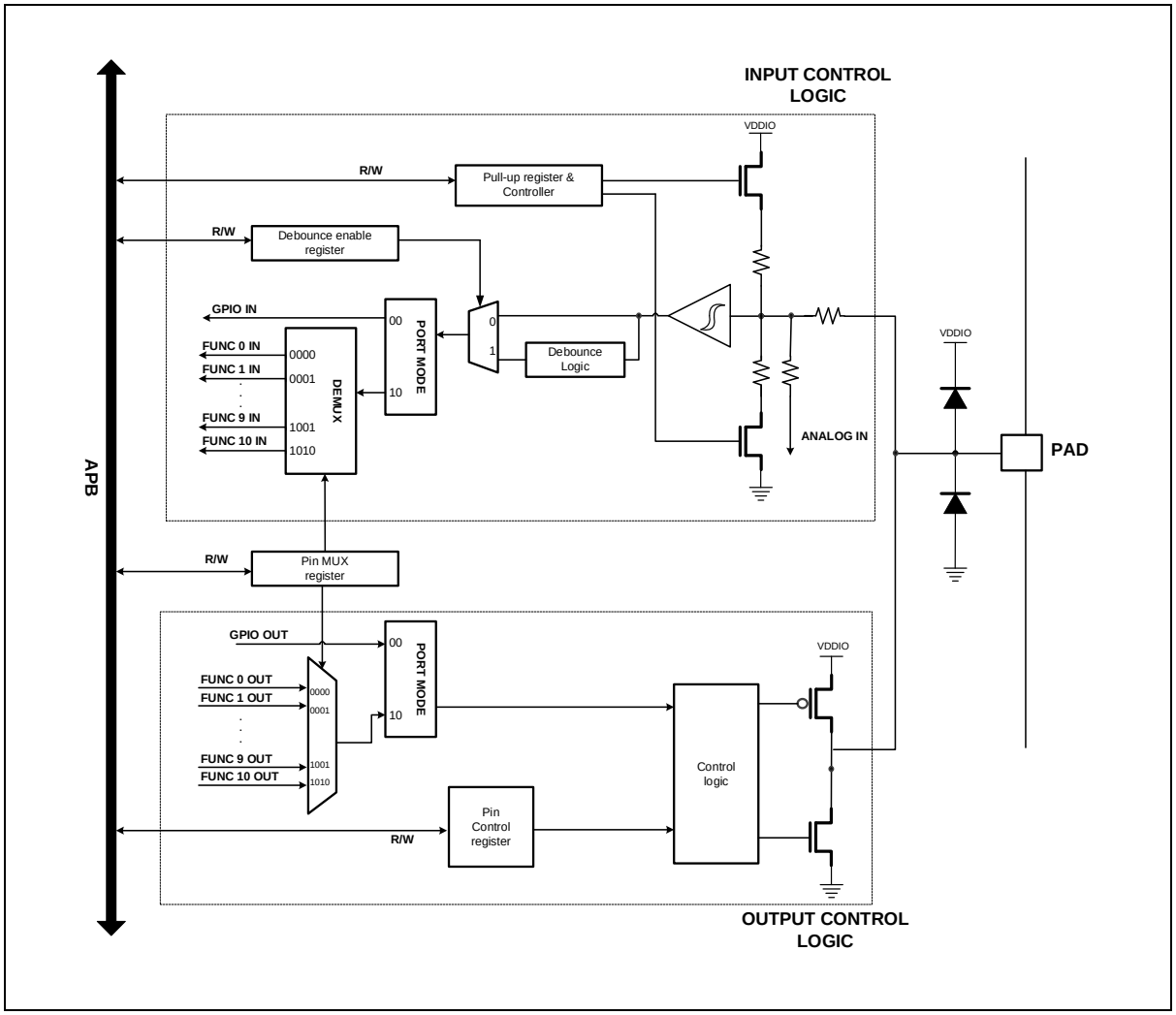


Figure 29. Port Diagram

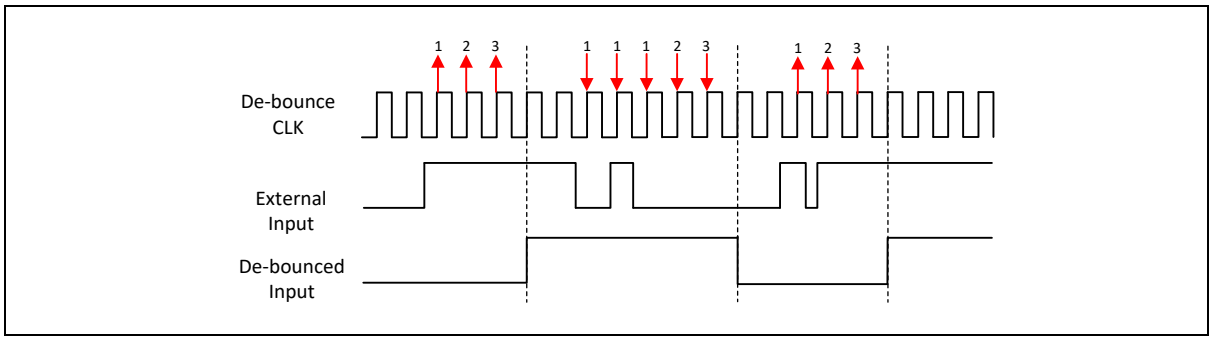


Figure 30. Debounce Function Timing Diagram

- When configured as output, the value written to the GPIO Output Data Register is output on the I/O Pin.
- When setting the Bit Set Register, GPIO Output Data Register set the high.
- When setting the Bit Clr Register, GPIO Output Data Register set the Low.
- The Input Data Register captures the data present on the I/O pin or Debounced input data at every GPIO Clock cycle.

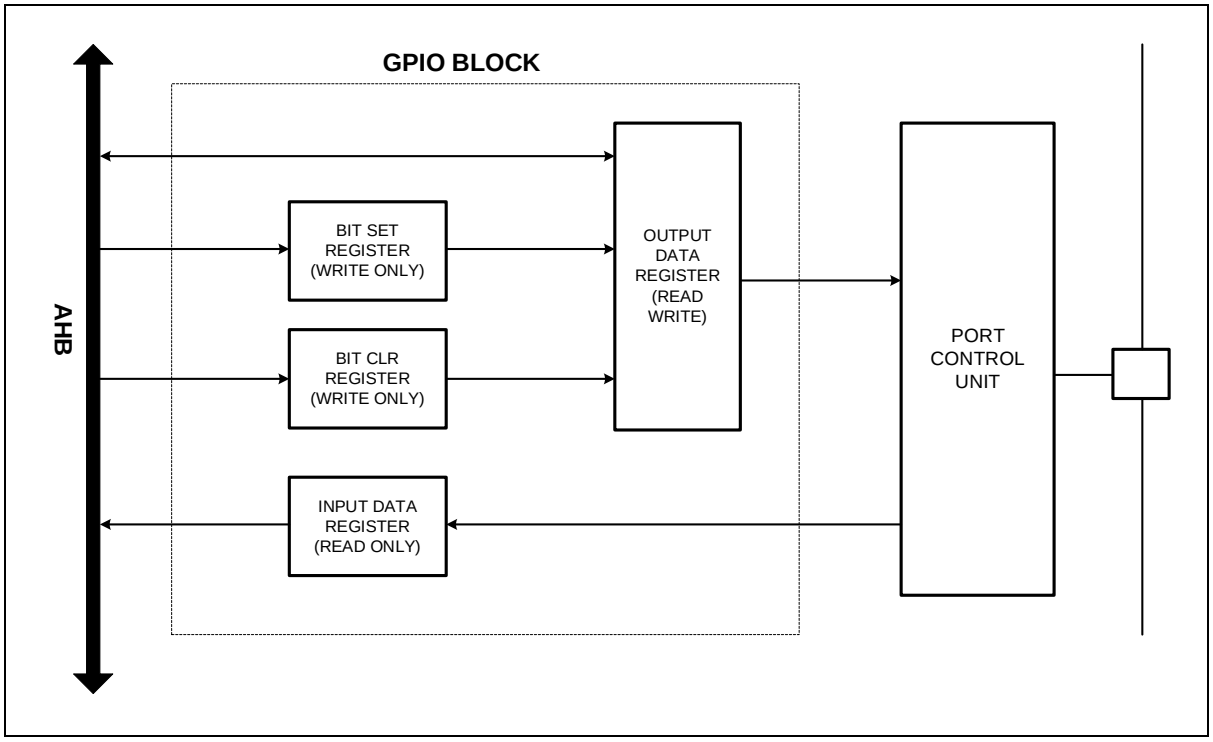


Figure 31. GPIO Diagram

6 Flash memory controller

Flash Memory Controller is an internal flash memory interface controller, and includes following features as shown below:

- 64 or 32KB Flash code memory
- Programmable wait control (0 to 4)
- Read protection support
- Self-Program support
- User option area
- 3-page (each 512 Bytes)
- Erase, Program in user mode

Table 20. Flash Memory Controller Features

Item	Description	
Size	64KB	32KB
Start Address	0x0000_0000	0x0000_0000
End Address	0x0001_0000	0x0000_7FFF
Page Size	512-byte	512-byte
Total Page Count	128 pages	64 pages
PGM Unit	512-byte	512-byte
Erase Unit	512-byte / 1KB / 4KB / bulk	512-byte/ 1KB/ 4KB/ bulk

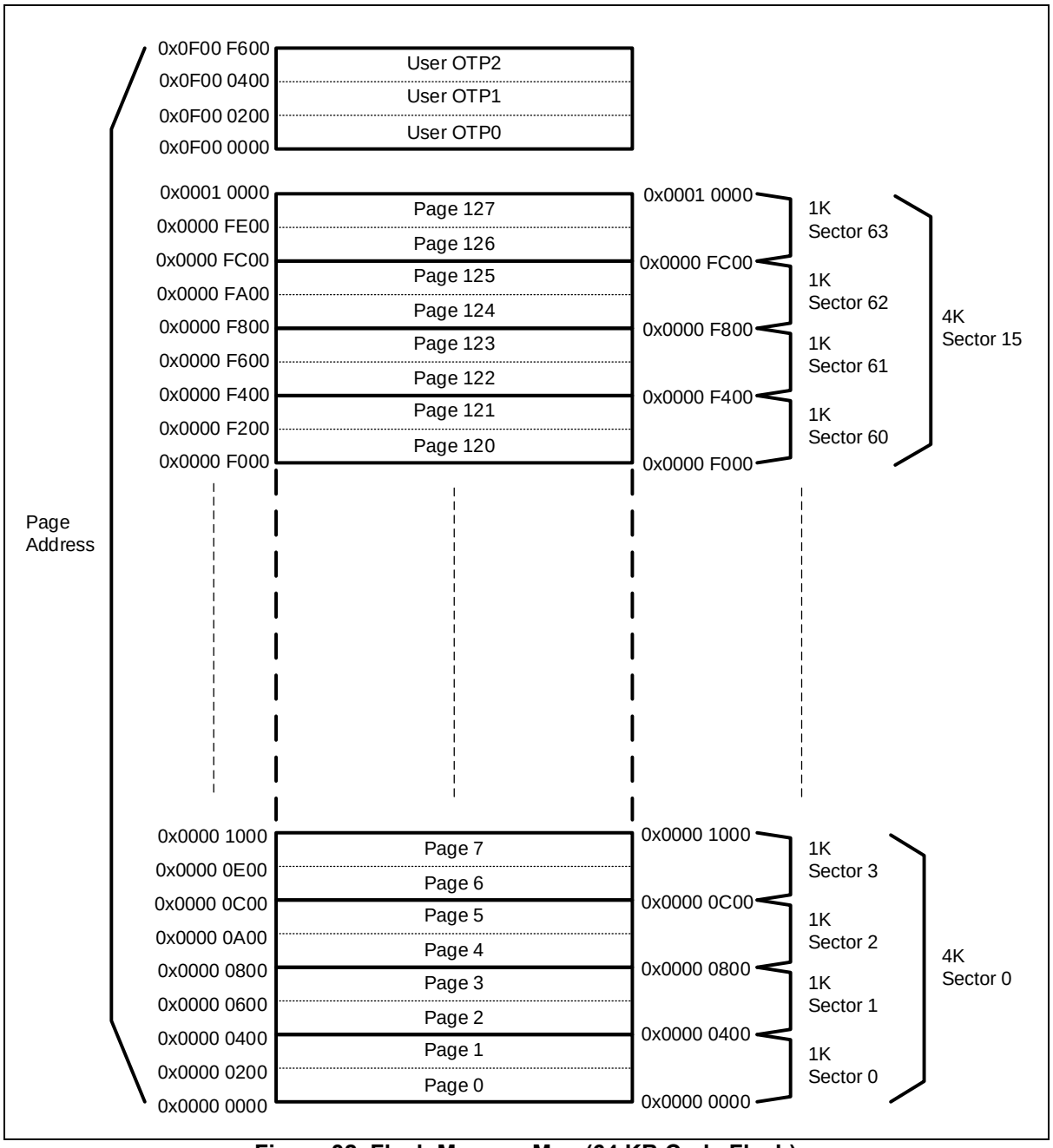


Figure 32. Flash Memory Map (64 KB Code Flash)

6.1 Registers

Base address of flash memory controller is introduced in the followings:

Table 21. Base Address of Flash Memory Controller

Name	Base address
Flash controller	0x4000_0100

Table 22. FMC Register Map

Name	Offset	Type	Description	Reset value	Ref.
FMC_MR	0x04	RW	Flash Memory Mode Select register	0x0100_0000	6.1.1
FMC_CR	0x08	RW	Flash Memory Control register	0x0000_0000	6.1.2
FMC_AR	0x0C	RW	Flash Memory Address register	0x0000_0000	6.1.3
FMC_DR	0x10	RW	Flash Memory Data register	0x0000_0000	6.1.4
FMC_BUSY	0x18	RO	Flash Write Busy Status Register	0x0000_0000	6.1.5
FMC_CRC	0x20	RO	Flash CRC16 check value	0x0000_FFFF	6.1.6
FMC_CFG	0x30	RW	Flash Memory Config Register	0x0000_8200	6.1.7
FMC_WPROT	0x34	RW	Write Protection Register	0xFFFF_FFFF	6.1.8
FMC_LOCK	0x3C	RW	Flash LOCK register	0x0000_00FF	6.1.9

6.1.1 FMC_MR: flash memory mode register

FMC_MR is an internal flash memory mode register. Size of this register is 32-bit.

FMC_MR=0x4000_0104

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			ACODE
0x010000			0x00
-			RW

7 0	ACODE	5A → A5	Flash mode entry
		A5 → 5A	Trim mode entry
		81 → 28	AMBA mode entry
		66 → 99	PROT mode entry

6.1.2 FMC_CR: flash memory control register

Internal flash memory control register.

FMC_CR=0x4000_0108

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	SELFPGM	Reserved		BBLOCK	MAS	SECT4K	SECT1K	PMODE	WADCK	PGM	ERS	HVEN
-	0	-		0	0	0	0	0	0	0	0	0
-	RW	-		RW	RW	RW	RW	RW	RW	RW	RW	RW

23	SELFPGM		When this bit is set ("1"), PGM/ERS/HVEN will be cleared automatically after WRBUSY falling edge. It also enable CPU wait control when HVEN bit is set(1) (start of program or erase operation) It also affects to PMODE bit operation.
8	BBLOCK	0	Boot Block(1st 4KB) not protected from Mass(Bulk) Erase
		1	Boot Block(1st 4KB) protection enable from Mass(bulk) erase
7	MAS	0	Mass (bulk) erase disable
		1	Mass (bulk) erase enable.
6	SECT4K	0	Sector 4K erase disable
		1	Sector 4K erase enable
5	SECT1K	0	Sector 1K erase disable
		1	Sector 1K erase enable
4	PMODE	0	Normal mode
		1	PMODE enable(Flash Address path is connected with FMC_AR) PMODE only valid when SELFPGM bit was not set(when SELFPGM = 0)
3	WADCK	0	Program/Erase address data latch clock disable
		1	Program/Erase address data latch clock enable, this bit assert for one system clock period so user cannot read
2	PGM	0	Program mode disable
		1	Program mode enable
1	ERS	0	Erase mode disable
		1	Erase mode enable
0	HVEN	0	High Voltage cycle disable
		1	High Voltage cycle enable (start program or erase cycle) User must set and clear in PMODE. In SELFPGM mode, user must set HVEN then HVEN will be cleared automatically after WRBUSY goes low.

6.1.3 FMC_AR: flash memory address register

Internal flash memory program, erase address register

FMC_AR=0x4000_010C

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			FADDR
-			0x0000
-			RW

13 0	FADDR	Word (32-bit) base address: 64KB Flash. Auto Incremental after WADCK trigger (after latching of target address).

6.1.4 FMC_DR: flash data input register

Internal flash memory Data Input register

FMC_DR=0x4000_0110

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
FDATA			
0x00000000			
RW			

31 0	FDATA	Data

6.1.5 FMC_BUSY: flash write busy status register

Flash write (program/erase) busy status monitor register. This register is 1-bit read only register.

FMC_BUSY=0x4000_0118

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			WRBUSY
-			0
-			RO

0	WRBUSY	Indicate the Status at Program or Erase cycle

6.1.6 FMC_CRC: flash CRC check register

The built-in CRC calculator calculates the flash data automatically.

FMC_CRC=0x4000_0120

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		CRC16	
		0xFFFF	
-		RO	

15 0	CRC16	CRC check value read register Polynomial : CRC ($G_1(x) = x^{16} + x^{12} + x^5 + 1$) data width: 32 (the first serial bit is D[31])

6.1.7 FMC_CFG: flash memory config register

Internal flash memory Config register. This register has the same address with FMTRIM0 register

FMC_CFG=0x4000_0130

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		Reserved	WAIT	CRCINIT	CRCEN	Reserved
0x0000		10000	010	0	0	-
RW		-	RW	RW	RW	-

31 16	WTIDKY		Write Identification Key. On writes, write 0x7858 to these bits, otherwise the write is ignored.
10 8	WAIT		This bits only be written in AMBA mode and MSB 16-bit (bit [31:16]) must be 0x7858
		000	flash access in 1 cycle (0-wait)
		001	flash access in 2 cycles (1-wait)
		010	flash access in 3 cycles (2-wait) - default
		011	flash access in 4 cycles (3-wait)
		1xx	flash access in 5 cycles (4-wait)
7	CRCINIT	0	When this bit is set('1'), CRC register will be initialized It should be reset again before read flash to generate CRC16 calculation (Initial value of FMC_CRC is 0xFFFF)
6	CRCEN	1	CRC16 enable CRC value will be calculated at every flash read timing

6.1.8 FMC_WPROT: write protection register

Internal flash memory write protection register.

FMC_WPROT=0x4000_0134

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
WPROT (A31G213)			
0xFFFF_FFFF			
RW			
Reserved		WPROT (A31G212)	
-		0xFFFF	
-		RW	

31 0	WPROT	Write protection Each 2 KB segments for whole memory address **NOTE**: The FM_WPROT register can only be modified in PROT (FM_MR = 66-> 99) mode.

6.1.9 FMC_LOCK: flash lock register

Internal flash memory read protection register.

FMC_LOCK=0x4000_013C

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			RPROT
-			0x00FF
-			RW

7 0	RPROT	Read protection 0x0000_00FF is Default. Any other value will lock flash(enable read protection) In user mode, 0xFF cannot be written To unlock, user must erase LOCK area of Flash next to MAS(bulk) erase - 1st MAS(bult) erase and then erase LOCK area - To unlock the flash, pin reset or power-on reset required **NOTE**: The FMC_WPROT register can only be modified in PROT (FMC_MR = 66-> 99) mode.

6.2 Functional description

6.2.1 Flash erase and program examples

Basic steps of flash memory programming consist of the followings. Minimum Program or Erase unit is a Page, and 32-word (128-byte) becomes a page.

- Sector 4KB Erase
- Page program
- Self page erase
- Self page program

For all of erase operations, pre-program operation is required to prevent over erase of flash memory cells. In addition, a user must enable 48MHz internal oscillator first to erase or program flash.

Bulk Erase example

1. Clear all write protection bits.
   Write 0x66 and then 0x99 into FMC_MR register to access FMC_WPROT register.
   Write 0x00000000 into FMC_WPROT register, and clear FMC_MR register as 0x00.
2. Set FMC_CR register access mode.
   Write 0x5A and then 0xA5 into FMC_MR register.
3. Set PMODE, MAS (bulk), ERS (erase) bit of FMC_CR register.
4. Set WADCK bit of FMC_CR register (write 1) to latch control bits.
   Don't set WADCK bit with step 3 bits(PMODE, MAS, ERS) at the same time.
   WADCK will be cleared automatically, so cannot read as "1".
5. Set HVEN (high voltage enable) of FMC_CR register to start Erase operation.
6. Set until WRBUSY bit of FWBUSY register is cleared.
   No IRQ for Flash PMODE operation, S/W must poll this bit.
7. Clear HVEN bit.
8. Clear MAS, PMODE, ERS bits of FMC_CR register.
9. To exit FMC_CR access mode, write 0x00 input FMC_MR register.

Sector (4KB) Erase example

1. Clear write protection bit of target address.
   Write 0x66 and then 0x99 into FMC_MR register to access FMC_WPROT register.
   Write 0x0 into corresponding FMC_WPROT bit, and clear FMC_MR register as 0x00.
2. Set FMC_CR register access mode.
   Write 0x5A and then 0xA5 into FMC_MR register.
3. Set FADDR for target address as byte address map (but least 2 bits will be ignored).
   Target address must be aligned by 4KB space.
4. Set PMODE, SECT4K, ERS (erase) bit of FMC_CR register.
5. Set WADCK bit of FMC_CR register (write 1) to latch control bits.
   Don't set WADCK bit with step 3 bits(PMODE, SECT4K, ERS) at the same time.
   WADCK will be cleared automatically, so cannot read as "1".
6. Set HVEN (high voltage enable) of FMC_CR register to start Erase operation.
7. Wait until WRBUSY bit of FWBUSY register is cleared.
   No IRQ for Flash PMODE operation, S/W must poll this bit.
8. Clear HVEN bit.
9. Clear SECT4K, PMODE, ERS bits of FMC_CR register.
10. To exit FMC_CR access mode, write 0x00 input FMC_MR register.

Program example

1. Clear write protection bit of target address.
   Write 0x66 and then 0x99 into FMC_MR register to access FMC_WPROT register.
   Write 0x0 into corresponding FMC_WPROT bit, and clear FMC_MR register as 0x00.
2. Set FMC_CR register access mode.
   Write 0x5A and then 0xA5 into FMC_MR register.
3. Set FADDR for target address as byte address map (but least 2 bits will be ignored).
   Target address must be aligned by word (32-bit) address space.
4. Write a word (32-bit) Data into FDATA register.
5. Set PMODE, PGM (program) bit of FMC_CR register.
6. Set WADCK bit of FMC_CR register (write 1) to latch control/address/data bits.
   Don't set WADCK bit with step 3 bits(PMODE, PGM) at the same time.
   WADCK will be cleared automatically, so cannot read as "1".
7. Set HVEN (high voltage enable) of FMC_CR register to start Erase operation.
8. Wait until WRBUSY bit of FWBUSY register is cleared.
   No IRQ for Flash PMODE operation, S/W must poll this bit.
9. Clear HVEN bit,
   to write at the next address, go to step 4.
   to write at the new address, go to step 3.
10. To finish program, clear PMODE, PGM bits of FMC_CR register.
11. To exit FMC_CR access mode, write 0x00 input FMC_MR register.

Self Erase example (512byte)

1. Clear write protection bit of target address.
   Write 0x66 and then 0x99 into FMC_MR register to access FMC_WPROT register.
   Write 0x0 into corresponding FMC_WPROT bit, and clear FMC_MR register as 0x00.
2. Set FMC_CR register access mode.
   Write 0x5A and then 0xA5 into FMC_MR register.
3. Set SELFPGM, ERS (erase) bit of FMC_CR register.
4. Write 0xFFFFFFFF into target address.
   Address must be aligned by 512 Bytes address space.
5. Flash will be stop until Erase complete.
   CPU will stop when Erase code runs in flash area.
   Erase code runs in SRAM area, flash cannot be accessed until erase complete.
6. To erase another page, go to step 4.
7. Clear SELFPGM, ERS bits of FMC_CR register.
8. To exit FMC_CR access mode, write 0x00 input FMC_MR register.

Self Program example

1. Clear write protection bit of target address.
   Write 0x66 and then 0x99 into FMC_MR register to access FMC_WPROT register.
   Write 0x0 into corresponding FMC_WPROT bit, and clear FMC_MR register as 0x00.
2. Set FMC_CR register access mode.
   Write 0x5A and then 0xA5 into FMC_MR register.
3. Set SELFPGM, PGM (program) bit of FMC_CR register.
4. Write 32-bit Data into target address.
   Address must be aligned by WORD (32-bit) address space.
5. Flash will be stop until Program complete.
6. CPU will stop when Program code runs in flash area.
   Program code runs in SRAM area, flash cannot be accessed until program complete.
   To Program another word, go to step 4.
7. Clear SELFPGM, PGM bits of FMC_CR register.
8. To exit FMC_CR access mode, write 0x00 input FMC_MR register.
9. Wait 5 clocks.
10. Clear Flash mode (write 0x00 and then write 0x00 into FMC_MR).

7 Direct Memory Access Controller (DMAC)

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 memory copying or moving of data within memory.

- 4 channels
- Single transfer only
- Support 8/16/32-bit data size
- Support multiple buffer with same size

Interrupt condition is transferred through a peripheral interrupt

7.1 Block diagram

In this section, DMAC block diagram is introduced in Figure 33.

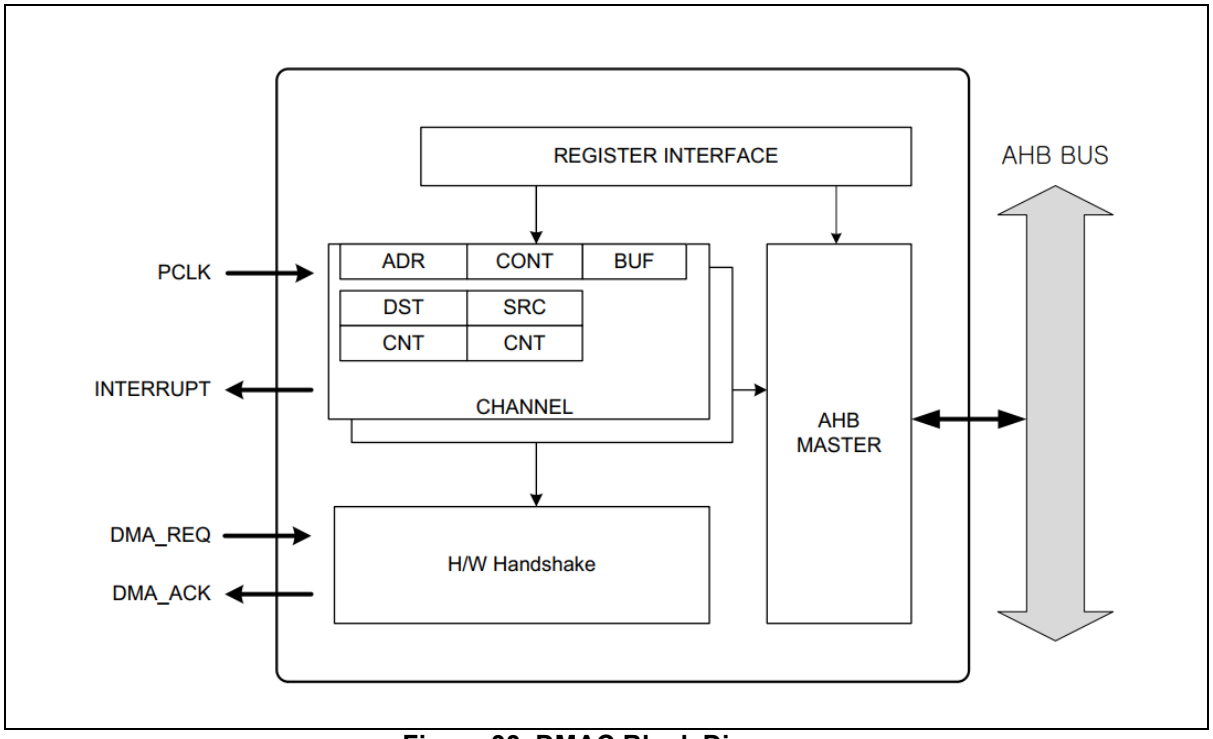


Figure 33. DMAC Block Diagram

7.2 Registers

Base address of DMAC is introduced in the followings:

Table 23. Base Address of DMAC

Name	Base address
DMA0	0x4000_0400
DMA1	0x4000_0410
DMA2	0x4000_0420
DMA3	0x4000_0430

Table 24. DMAC Register Map

Name	Offset	Type	Description	Reset value	Ref.
DMAn_CR	0x00	RW	DMA Channel n Control Register	0x0000_0000	7.2.1
DMAn_SR	0x04	RW	DMA Channel n Status Register	0x0000_0080	7.2.2
DMAn_PAR	0x08	RW	DMA Channel n Peripheral Address	0x4000_0000	7.2.3
DMAn_MAR	0x0C	RW	DMA Channel n Memory Address	0x2000_0000	7.2.4

7.2.1 DMAn.CR: DMA controller configuration register

DMAn_CR registers are DMA operation control registers, and the register size is 32-bit.

DMA0_CR=0x4000_0400, DMA1_CR=0x4000_0410
DMA2_CR=0x4000_0420, DMA3_CR=0x4000_0430

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	Reserved	PERSEL	Reserved	SIZE	DIR	Reserved
-	0x000	-	0000	-	00	0	-
-	RW	-	RW	-	RW	RW	-

27 16	TRANSCNT	Number of DMA transfer remained Required transfer number should be written before enable DMA transfer.
	0	DMA transfer is done.
	N	N transfers are remained
11 8	PERISEL	Peripheral selection
	N	Associated peripheral selection. Refer to DMA Peripheral connection table
3 2	SIZE	Bus transfer size.
	00	DMA transfer is byte size transfer
	01	DMA transfer is half word size transfer
	10	DMA transfer is word size transfer
	11	Reserved
1	DIR	Select transfer direction.
	0	Transfer direction is from memory to peripheral. (TX)
	1	Transfer direction is from peripheral to memory (RX)

NOTE: A DMA channel will be connected with selected peripheral. Below table shows peripheral selection numbers. This PERISEL field should be set with proper number of peripheral which will be connected with DMA interface.

Table 25. DMAC PERISEL Selection

PERISEL[3:0]	Associate peripheral	PERISEL[3:0]	Associate peripheral
0	CHANNEL IDLE	8	N/A
1	UART0 RX	9	SPI20 RX
2	UART0 TX	10	SPI20 TX
3	UART1 RX	11	SPI21 RX
4	UART1 TX	12	SPI21 TX
5	CRC	13	N/A
6	N/A	14	N/A
7	N/A	15	N/A

NOTE: PERISEL cannot have the same value in different channels. If the same PERISEL value is written in more than one channel, proper operation cannot be guaranteed. Unused channel should have CHANNEL IDLE value in PERISEL bit positions.

7.2.2 DMAn_SR: DMA controller status register

DMACn_SR registers represent current status of DMA Controller, and enable DMA function. The register size is 8-bit.

DMA0_SR=0x4000_0404, DMA1_SR=0x4000_0414
DMA2_SR=0x4000_0424, DMA3_SR=0x4000_0434

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			EOT	Reserved		DMAEN
-			1	-		0
-			RO	-		RW

7	EOT	End of transfer.	
		0	Data to be transferred is existing. TRANSCNT shows non zero value
		1	All data is transferred. TRANSCNT shows now 0
0	DMAEN	DMA Enable	
		0	DMA is in stop or hold state
		1	DMA is running or enabled

7.2.3 DMAn_PAR: DMA controller peripheral address register

DMAn_PAR registers represent peripheral addresses.

DMA0_PAR=0x4000_0408, DMA1_PAR=0x4000_0418
DMA2_PAR=0x4000_0428, DMA3_PAR=0x4000_0438

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
Peripheral BASE OFFSET		PAR	
0x4000		0x0000	
RO		RW	

15 0	PAR	Target Peripheral address of transmit buffer or receive buffer. Address is fixed address when each transfer is done. If DMAn_CR (DIR) is "0" this address is destination address of data transfer. If DMAn_CR (DIR) is "1", this address is source address of data transfer.

7.2.4 DMAn_MAR: DMA controller memory address register

DMAn_MAR registers represent the memory addresses.

DMA0_MAR=0x4000_040C , DMA1_MAR=0x4000_041C
DMA2_MAR=0x4000_042C , DMA3_MAR=0x4000_043C

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
Memory BASE OFFSET		MAR	
0x2000		0x0000	
RO		RW	

15 0	MAR	Target memory address of data transfer. Address is automatically incremented according to SIZE bits when each transfer is done.

7.3 Functional description

A DMA controller performs direct memory transfer by sharing the system bus with CPU core. The system bus is shared by two AHB (Advanced High-performance Bus) masters following the round-robin priority strategy. So the DMA controller can share the half of system bandwidth.

The DMA controller can be triggered only by a peripheral request. When a peripheral requests a transfer to the DMA controller, a corresponding channel is activated and the bus is accessed to transfer the requested data from memory to peripheral data buffer or vice versa.

Basic steps to trigger DMAC data transfer consist of following 10 steps:

1. Set both of peripheral address and memory address.
2. Configure DMA operation mode and transfer count.
3. Enable a DMA channel.
4. DMA request is occurred from the peripheral.
5. DMA activates the channel which was requested.
6. DMA reads data from source address and saves in internal buffer.
7. DMA writes the buffered data to destination address.
8. Transfer count number is decreased by '1'.
9. When the transfer count is '0', EOT flag is set and noticed to peripheral to issue the interrupt
10. DMA does not have interrupt sources, and the interrupt related DMA status can be shown from assigned peripheral interrupt.

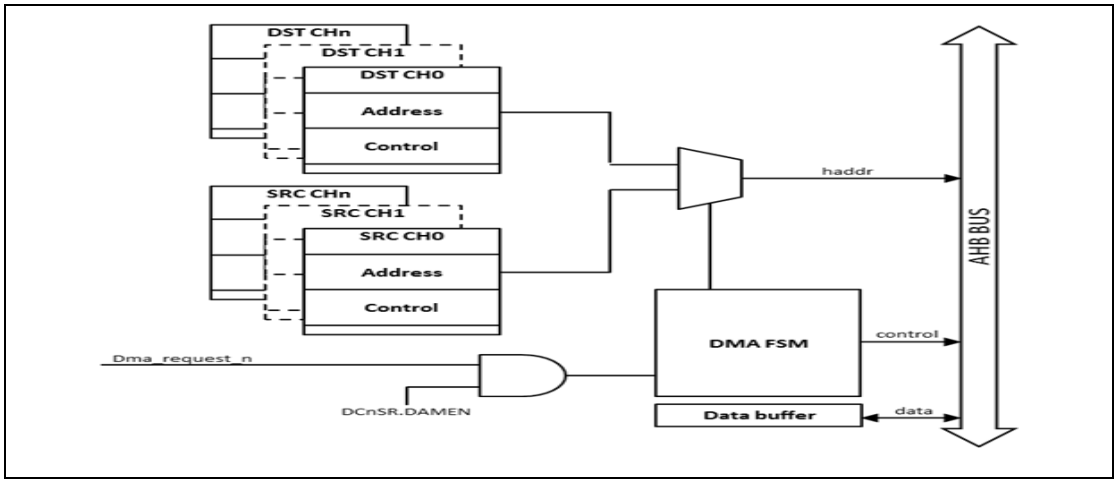


Figure 34. DMAC Functional Block Diagram

8 Watchdog Timer (WDT)

Watchdog timer (WDT) rapidly detects CPU's malfunction such as endless looping caused by noise, and resumes the CPU to the normal state. WDT signal for malfunction detection can be used as either a CPU reset or an interrupt request. When WDT is not being used for malfunction detection, it can be used as a timer generating an interrupt at fixed intervals.

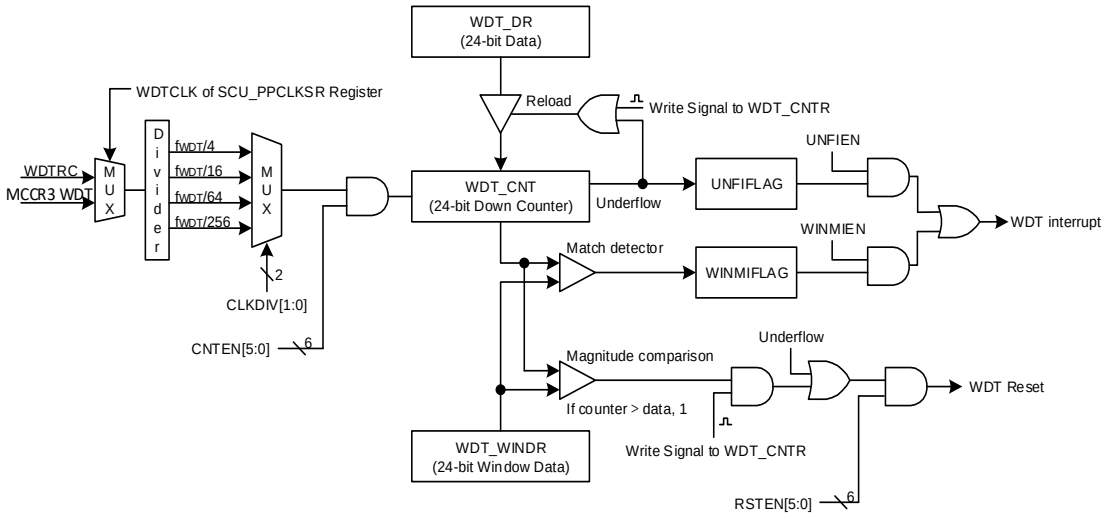
When WDT_CNT value is reached to WDT_WINDR value, a watchdog interrupt can be generated. The underflow time of WDT can be set by WDT_DR. If the underflow occurs, an internal reset is generated. WDT operates on the WDTRC embedded RC oscillator clock.

WDT of A31G21x series features followings:

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

8.1 WDT block diagram

In this section, WDT block diagram is introduced in Figure 35.



NOTE: If a reload signal is generated, the WDT_DR value is reloaded to WDT_CNT register after 1 clock delay with the clock which is selected by the CLKDIV bits.

Figure 35. WDT Block Diagram

8.2 Registers

Initial watchdog time-out period is set to 2,000-milisecond. Base address of WDT is introduced in the followings:

Table 26. Base Address of WDT

Name	Base address
WDT	0x4000_1A00

Table 27. WDT Register Map

Name	Offset	Type	Description	Reset value	Ref.
WDT_CR	0x00	RW	Watch-dog Timer Control Register	0x0000_0000	8.2.1
WDT_SR	0x04	RW	Watch-dog Timer Status Register	0x0000_0080	8.2.2
WDT_DR	0x08	RW	Watch-dog Timer Data Register	0x0000_3D09	8.2.3
WDT_CNT	0x0C	RO	Watch-dog Timer Counter Register	0x0000_0FFF	8.2.4
WDT_WINDR	0x10	RW	Watch-dog Timer Window Data Register	0x0000_FFFF	8.2.5
WDT_CNTR	0x14	WO	Watch-dog Timer Counter Reload Register	0x0000_0000	8.2.6

8.2.1 WDT_CR: watchdog timer control register

WDT module should be configured properly before running. WDT module can make reset event or assert interrupt signal to system. Size of this register is 32-bit.

WDT_CR=0x4000_1A00

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		RSTEN		CNTEN	WINMIEN	UNFIEN	CLKDIV
0x0000		000000		000000	0	0	00
WO		RW		RW	RW	RW	RW

31 16	WTIDKY	Write Identification Key.	
		On writes, write 0x5A69 to these bits, otherwise the write is ignored.	
15 10	RSTEN	Watch-dog Timer Reset Enable bits.	
		0x25	Disable watch-dog timer reset.
		Others	Enable watch-dog timer reset.
9 4	CNTEN	Watch-dog Timer Counter Enable bits.	
		0x1A	Disable watch-dog timer counter.
		Others	Enable window data match interrupt.
3	WINMIEN	Watch-dog Timer Window Match Interrupt Enable bit.	
		0	Disable window data match interrupt.
		1	Enable window data match interrupt.
2	UNFIEN	Watch-dog Timer Underflow Interrupt Enable bit.	
		0	Disable watch-dog timer underflow interrupt.
		1	Enable watch-dog timer underflow interrupt.
1 0	CLKDIV	Watch-dog Timer Clock Divider bits, The clock which is selected by SCU_PPCLKSR[0].	
		00	fWDT/4
		01	fWDT/16
		10	fWDT/64
		11	fWDT/256

8.2.2 WDT_SR: watchdog timer status register

WDT_SR register is 32-bit size, and able to do 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	Watch-dog Timer Counter Enable bit When the core is halted in the debug mode.
		0 The watch-dog timer counter continues even if the core is halted.
		1 The watch-dog timer counter is stopped when the core is halted.
		NOTE: This bit is set to "1b" by POR reset.
1	WINMIFLAG	Watch-dog Timer Window Match Interrupt Flag bit.
		0 No request occurred.
		1 Request occurred, This bit is cleared to '0' when write '1'.
0	UNFIFLAG	Watch-dog Timer Underflow Interrupt Flag bit.
		0 No request occurred.
		1 Request occurred, This bit is cleared to '0' when write '1.

8.2.3 WDT_DR: watchdog timer data register

WDT_DR register is used to update WDT_CNT register. Size of this register is 32-bit.

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		
-	0x003D09		
-	RW		

23 0	DATA	Watch-dog Timer Data bits. The range is 0x000000 to 0xFFFFFF.
		NOTE: Once any value is written to this data register, the register can't be changed until a system reset.

8.2.4 WDT_CNT: watchdog timer counter register

WDT_CNT register represents the current count value of 32-bit down counter. When the counter value reach to 0, the interrupt or reset will be asserted. Size of this register is 32-bit.

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		
-	0x000FFF		
-	RO		

23 0	CNT	Watch-dog Timer Counter bits.

8.2.5 WDT_WINDR: watchdog timer window data register

WDT_WINDR register is used to compare with WDT_CNT for WINDOW function. Size of this register is 32-bit.

If the WDT_CNT value equals the WDT_WINDER value, WDT window match interrupt is occurred.

If WDT_CNT is updated by WDT_CNTR when WDT_CNT value is bigger than WDT_WINDR Value, WDT reset is occurred.

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		
-	0x00FFFF		
-	RW		

23 0	WDATA	Watch-dog Timer Data bits. The range is 0x000000 to 0xFFFFFF.
NOTE:	Once any value is written to this data register, the register can't be changed until a system reset.	

8.2.6 WDT_CNTR: watchdog timer counter reload register

WDT_CNTR register is used to make reload signal. If reload signal is I, WDT_DR Value is reloaded to WDT_CNT. Size of this register is 32-bit.

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
-			0x00
-			WO

7 0	CNTR	Watch-dog Timer Counter Reload bits.	
		0x6A	Reload the WDT_DR value to watch-dog timer counter and re-start. (Automatically cleared to "0x00" after operation)
		Others	No effect

8.3 Functional description

WDT counter can be enabled by CNTEN (WDT_CR[9:4]). Corresponding bit field of the register CNTEN is set with any value other than 0x1A. As WDT activates, a down counter will start counting from loaded value.

If RSTEN (WDT_CR[15:10]) is set with any value other than 0x25, WDT reset would be asserted when the WDT counter value reaches to '0' (underflow event) from WDT_DR value. Before WDT counter reaches to '0', software can write 0x6A to WDT_CNTR register in order 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 is greater than window data.

8.3.1 Timing diagram

In this section, WDT interrupt and reset timing diagram is introduced in Figure 36.

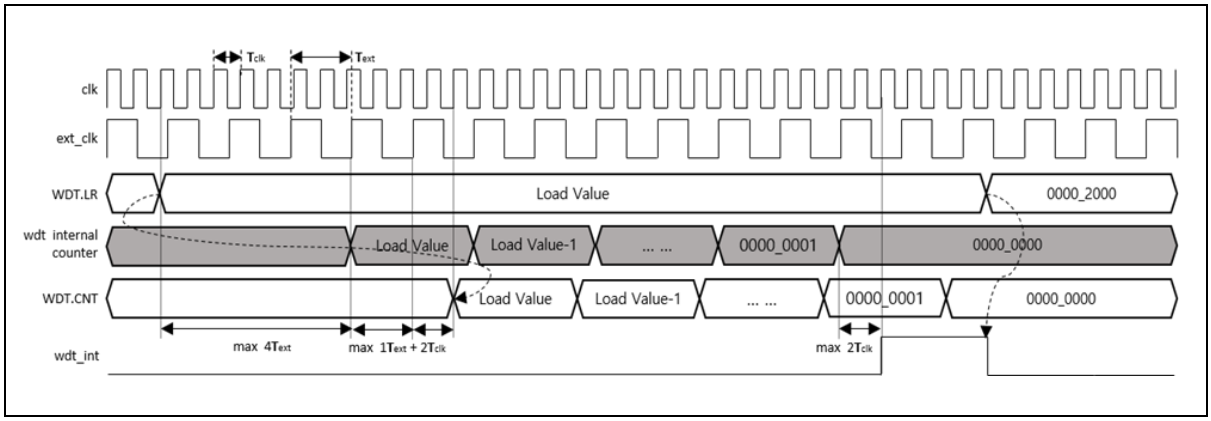


Figure 36. WDT Interrupt and WDT Reset Timing Diagram

8.3.2 Prescale table

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

Clock sources of watchdog timer can be WDTRC or SCR_MCCR3. PCLK can be selected by WDTCLK (SCU_PPCLKSR[0]) set to '1'.

To make WDT counter base clock, user can control 2-bit pre-scaler CLKDIV [1:0] in WDT_CR register and the maximum pre-scaled value is "clock source frequency/256". The pre-scaled WDT counter clock frequency values are listed in following table.

Selectable clock source (31.250 KHz ~ 48 MHz) and the time out interval when 1 count

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

*Time out period (time out period from load Value to interrupt set '1')

Table 28. Pre-scaled WDT Counter Clock Frequency

	WDTRC	SCU_MCCR3 WDT
WDTCLKIN	31.250KHz	SCU_MCCR3 WDT
WDTCLKIN/4	7.8125KHz	SCU_MCCR3 WDT/4
WDTCLKIN/16	1.953125KHz	SCU_MCCR3 WDT/16
WDTCLKIN/64	488.28125Hz	SCU_MCCR3 WDT/64
WDTCLKIN/256	122.0703125Hz	SCU_MCCR3 WDT/256

9 Watch Timer (WT)

Watch timer (WT) has functions for RTC (Real Time Clock) operation. It is generally used for RTC design. WT consists of a clock source select circuit, a timer counter circuit, an output select circuit and watch timer control registers.

Prior to operate WT, a user needs to determine an input clock source and output interval, and to set WTEN as '1' in watch timer control register (WT_CR). It is able to execute simultaneously or individually. To stop or reset WT, clear the WTEN bit in WT_CR register.

Watch timer counter circuit corporates a 26-bit counter. Low 14 bits of the counter form a binary counter and high 12 bits form an auto reload counter in order to raise resolution. In WTR, it can control WT clear and set interval value at write time, and it can read 12-bit WT counter value at read time.

9.1 WT block diagram

As shown in Figure 37, WT of A31G21x series have the following blocks:

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

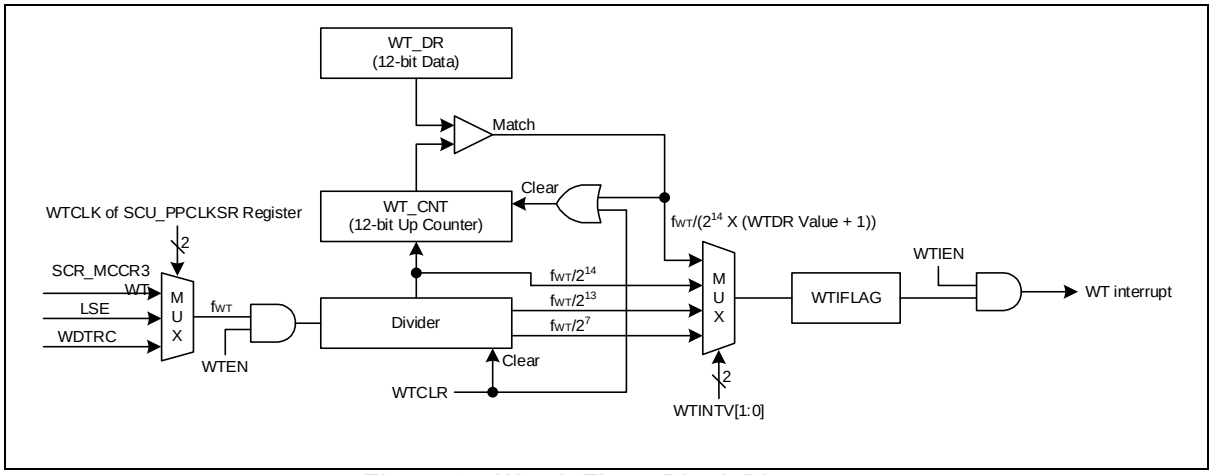


Figure 37. Watch Timer Block Diagram

9.2 Registers

Base address of WT is introduced in the followings:

Table 29. Base Address of WT

Name	Base address
WT	0x4000_2000

Table 30. WT Register Map

Name	Offset	Type	Description	Reset value	Ref.
WT_CR	0x00	RW	Watch Timer Control Register	0x0000_0000	9.2.1
WT_DR	0x04	RW	Watch Timer Data Register	0x0000_0FFF	9.2.2
WT_CNT	0x08	RO	Watch Timer Counter Register	0x0000_0000	9.2.3

9.2.1 WT_CR: watch timer control register

WT_CR is a 32-bit register, and able to do 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	WTIEN	Reserved	WTIFLAG	WTCLR
-			0	-	00	0	-	0	0
-			RW	-	RW	RW	-	RW	RW

7	WTEN	Watch Timer Operation Enable bit.	
		0	Disable watch timer operation.
		1	Enable watch timer operation.
5 4	WTINTV	Watch Timer Interval Selection bits.	
		00	$fWT/2^7$
		01	$fWT/2^{13}$
		10	$fWT/2^{14}$
		11	$fWT/(2^{14}x(WT\_DR\ value + 1))$
		NOTE: These bits should be changed during WTEN bit is "0".	
3	WTIEN	Watch Timer Interrupt Enable bit.	
		0	Disable watch timer interrupt.
		1	Enable watch timer interrupt.
1	WTIFLAG	Watch Timer Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
0	WTCLR	Watch Timer Counter and Divider Clear bit.	
		0	No effect.
		1	Clear the counter and divider (Automatically cleared to "0b" after operation)

9.2.2 WT_DR: watch timer data register

WT_DR is a 32-bit register.

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
-			0xFFF
-			RW

11 0	WTDATA	Watch Timer Data bits. The range is 0x001 to 0xFFF.

9.2.3 WT_CNT: watch timer counter register

Watch Timer Counter Register is 32-bit register.

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
-			0x000
-			RO

11 0	CNT	Watch Timer Counter bits.

10 16-bit timer10/11/12/13

16-bit timer block comprises 4 channels of 16-bit general purpose timers. Each channel has an independent 16-bit counter and a dedicated prescaler that feeds a counting clock. 16-bit timer supports periodic timer, PWM pulse, one-shot and capture mode. In addition, one more optional free-run timer is provided. Main purpose of 16-bit timer is a periodical tick timer.

16-bit timer of A31G21x series features the followings:

- 16-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- Capture mode
- 12-bit prescaler
- Synchronous start and clear function

Table 31 introduces pins assigned for 16-bit timer.

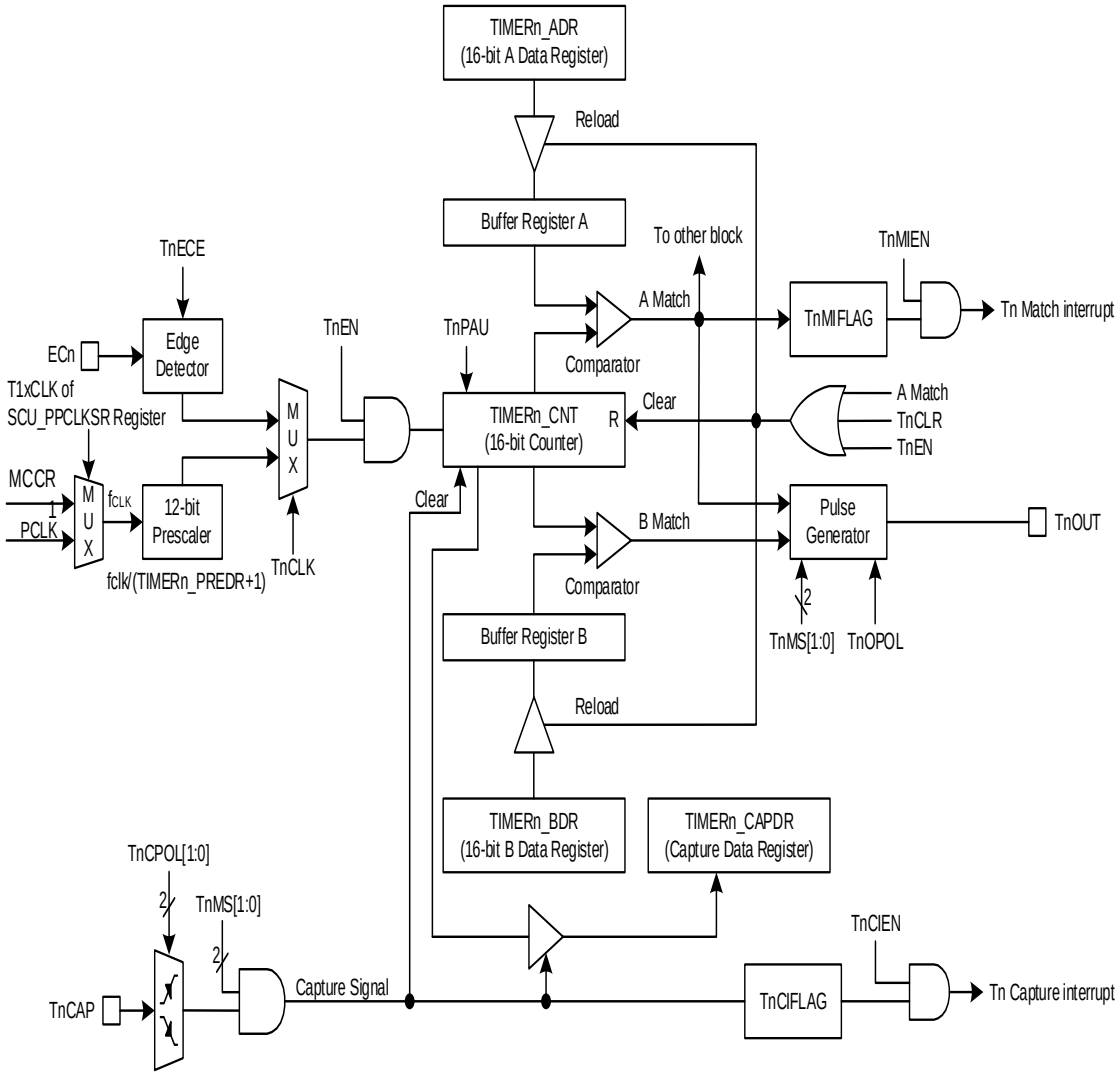
Table 31. Pin Assignment of 16-bit Timer: External Pins

Pin name	Type	Description
ECn	I	Timer 1n External Clock input
TnC	I	Timer 1n Capture input
TnO	O	Timer 1n Output

NOTE: n = 10, 11, 12, and 13

10.1 16-bit timer block diagram

In this section, 16-bit timer is described in a block diagram in Figure 38.



NOTES:

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

Figure 38. 16-bit Timer Block Diagram

10.2 Registers

Base address of 16-bit timer is introduced in the followings:

Table 32. Base Address of 16-bit Timer

Name	Base address
TIMER10	0x4000_2100
TIMER11	0x4000_2200
TIMER12	0x4000_2300
TIMER13	0x4000_2700

Table 33. TIMER 1n Register Map

Name	Offset	Type	Description	Reset value	Ref.
TIMERn_CR	0x00	RW	Timer/Counter n Control Register	0x0000_0000	11.2.1
TIMERn_ADR	0x04	RW	Timer/Counter n A Data Register	0x0000_FFFF	11.2.2
TIMERn_BDR	0x08	RW	Timer/Counter n B Data Register	0x0000_FFFF	11.2.3
TIMERn_CAPDR	0x0C	RO	Timer/Counter n Capture Data Register	0x0000_0000	11.2.4
TIMERn_PREDR	0x10	RW	Timer/Counter n Prescaler Data Register	0x0000_0FFF	11.2.5
TIMERn_CNT	0x14	RO	Timer/Counter n Counter Register	0x0000_0000	11.2.6

NOTE: n = 10, 11, 12, and 13

10.2.1 TIMERn_CR: timer/counter 1n control register

Timer module should be configured properly before running. When a target purpose is defined, the timer can be configured in the TIMERn_CR register. After configuring TIMER1n_CR, a user can start or stop the timer function by using TIMERn_CR.

TIMERn_CR is a 32-bit register, and able to do 32/16/8-bit access. (n = 10, 11, 12, and 13)

TIMER10_CR=0x4000_2100, TIMER11_CR=0x4000_2200
TIMER12_CR=0x4000_2300, TIMER13_CR=0x4000_2700

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	TnECE	Reserved	TnOPOL	TnCPOL	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 bit.
		0 Disable timer n operation.
		1 Enable timer n operation. (Counter clear and start)
14	TnCLK	Timer n Clock Selection bit.
		0 Select an internal prescaler clock.
		1 Select an external clock.
		NOTE: This bit should be changed during TnEN bit is "0b".
13 12	TnMS	Timer n Operation Mode Selection bits.
		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: This bit should be changed during TnEN bit is "0b".
11	TnECE	Timer n External Clock Edge Selection bit.
		0 Select falling edge of external clock.
		1 Select rising edge of external clock.
8	TnOPOL	TnOUT Polarity Selection bit.
		0 Start high. (TnOUT is low level at disable)
		1 Start low. (TnOUT is high level at disable)
7 6	TnCPOL	Timer n Capture Polarity Selection bits.
		00 Capture on falling edge.
		01 Capture on rising edge.
		10 Capture on both of falling and rising edge.
		11 Reserved.
5	TnMIEN	Timer n Match Interrupt Enable bit.
		0 Disable timer n match interrupt.
		1 Enable timer n match interrupt.
4	TnCIEN	Timer n Capture Interrupt Enable bit.
		0 Disable timer n capture interrupt.
		1 Enable timer n capture interrupt.
3	TnMIFLAG	Timer n Match Interrupt Flag bit.
		0 No request occurred.
		1 Request occurred, This bit is cleared to '0' when write '1'.
2	TnCIFLAG	Timer n Capture Interrupt Flag bit.
		0 No request occurred.
		1 Request occurred, This bit is cleared to '0' when write '1'.
1	TnPAU	Timer n Counter Temporary Pause Control bit.
		0 Continue counting.
		1 Temporary pause.
0	TnCLR	Timer n Counter and Prescaler Clear bit.
		0 No effect.
		1 Clear timer n counter and prescaler. (Automatically cleared to "0b" after operation)

10.2.2 TIMERn_ADR: timer/counter 1n A data register

TIMERn_ADR is a 32-bit register, and able to do 32/16/8-bit access. (n = 10, 11, 12, and 13)

TIMER10_ADR =0x4000_2104, TIMER11_ADR =0x4000_2204
TIMER12_ADR =0x4000_2304, TIMER13_ADR =0x4000_2704

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	
-		0xFFFF	
-		RW	

15 0	ADATA	Timer/Counter n A Data bits. The range is 0x0002 to 0xFFFF. **NOTE**: Do not write "0000H" in the TIMERn_ADR register when PPG mode.

10.2.3 TIMERn_BDR: timer/counter n B data register

TIMERn_BDR is a 32-bit register, and able to do 32/16/8-bit access. (n = 10, 11, 12, and 13)

TIMER10_BDR =0x4000_2108, TIMER11_BDR =0x4000_2208
TIMER12_BDR =0x4000_2308, TIMER13_BDR =0x4000_2708

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	
-		0xFFFF	
-		RW	

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

10.2.4 TIMERn_CAPDR: timer/counter n capture data register

TIMERn_ CAPDR is a 32-bit register, and able to do 32/16/8-bit access. (n = 10, 11, 12, and 13)

TIMER10_CAPDR =0x4000_210C, TIMER11_CAPDR =0x4000_220C
TIMER12_CAPDR =0x4000_230C, TIMER13_CAPDR =0x4000_270C

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		CAPD	
-		0x0000	
-		RO	

15 0	CAPD	Timer/Counter n Capture Data bits.

10.2.5 TIMERn_PREDR: timer/counter n prescaler data register

TIMERn_PREDR is a 32-bit register, and able to do 32/16/8-bit access. (n = 10, 11, 12, and 13)

TIMER10_PREDR =0x4000_2110, TIMER11_PREDR =0x4000_2210
TIMER12_PREDR =0x4000_2310, TIMER13_PREDR =0x4000_2710

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															
-																0xFFF															
-																RW															

11 0	PRED	Timer/Counter n Prescaler Data bits.

10.2.6 TIMERn_CNT: timer/counter n counter register

TIMERn_CNT is a 32-bit register, and able to do 32/16/8-bit access. (n = 10, 11, 12, and 13)

TIMER10_CNT =0x4000_2114, TIMER11_CNT =0x4000_2214
TIMER12_CNT =0x4000_2314, TIMER13_CNT =0x4000_2714

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															
-																0x0000															
-																RO															

15 0	CNT	Timer/Counter n Counter bits.

10.3 Functional description

10.3.1 Timer counter 10/11/12/13

Timer/counter n can be clocked by an internal or an external clock source (ECn). Its clock source is selected by a clock selection logic which is controlled by clock selection bits (TnCLK). (n = 10, 11, 12, and 13)

- TIMER n clock source: {fCLK/(TIMERn_PREDR +1), ECn

In capture mode, by TnCAP, data is captured into input capture data register (TIMERn_CAPDR). TIMER 1n results the comparison between a counter and the data register through TnO port in timer/counter mode. In addition, TIMER 1n outputs PWM wave form through TnO port in the PPG mode.

Table 34 introduces various operating modes of TIMERn according to the value of timer/counter register.

Table 34. TIMER 1n Operating Modes

TnEN	Alternative mode	TnMS[1:0]	TIMER1n_PREDR	Timer 1n
1	PxAFSR1 = AF1	00	0xXXX	16-bit Timer/Counter Mode
1	PxAFSR1 = AF2	01	0xXXX	16-bit Capture Mode
1	PxAFSR1 = AF1	10	0xXXX	16-bit PPG Mode (one-shot mode)
1	PxAFSR1 = AF1	11	0xXXX	16-bit PPG Mode(repeat mode)

10.3.2 16-bit timer/counter mode

16-bit timer/counter mode is selected by control register as shown in Figure 39. The 16-bit timer has a counter and data register. The counter register is increased by internal or external clock input. Timer n can use the clock input with 12-bit prescaler division rates (TIMERn_PREDR) and external clock (ECn). When each value of TIMERn_CNT and TIMERn_ADR are identical in timer 1n, a match signal is generated and the interrupt of Timer 1n occurs. The TIMERn_CNT values are automatically cleared by the match signal. It can be also cleared by software (TnCLR).

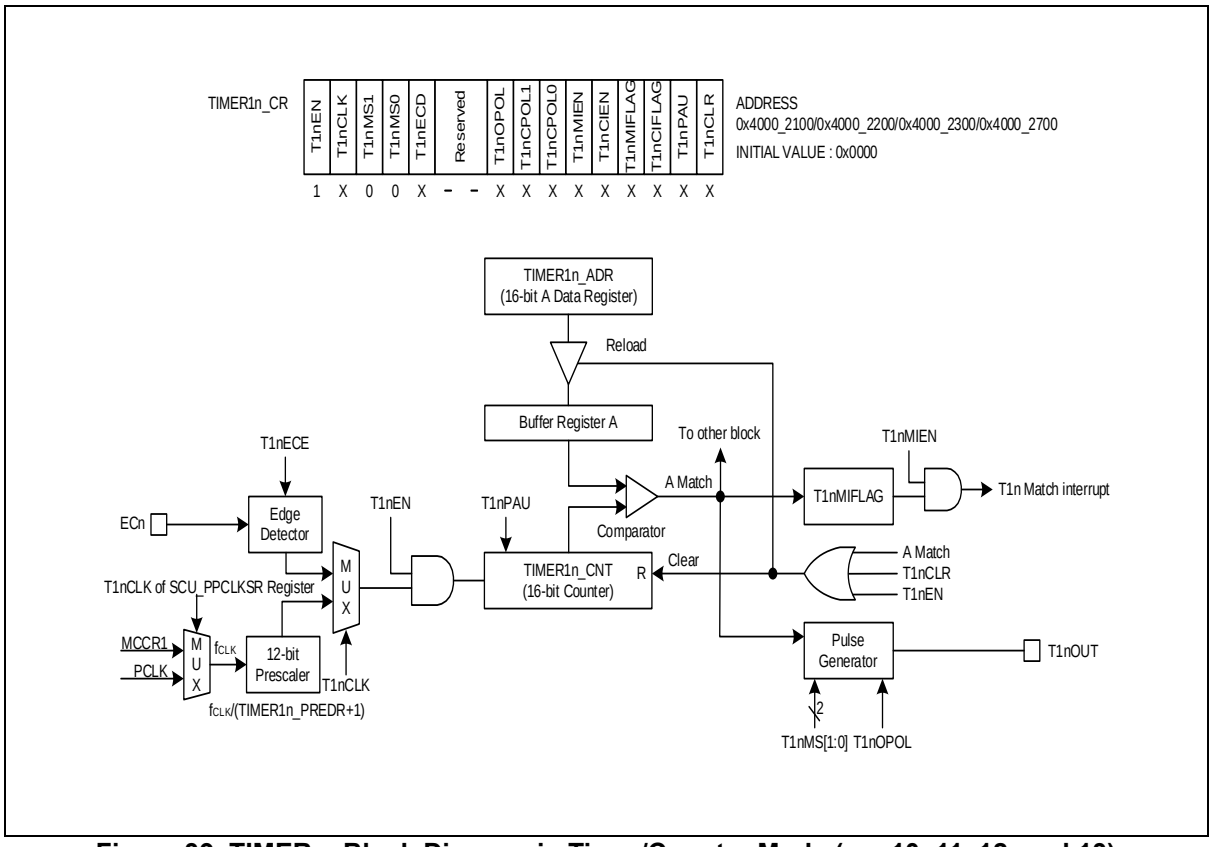


Figure 39. TIMER n Block Diagram in Timer/Counter Mode (n = 10, 11, 12, and 13)

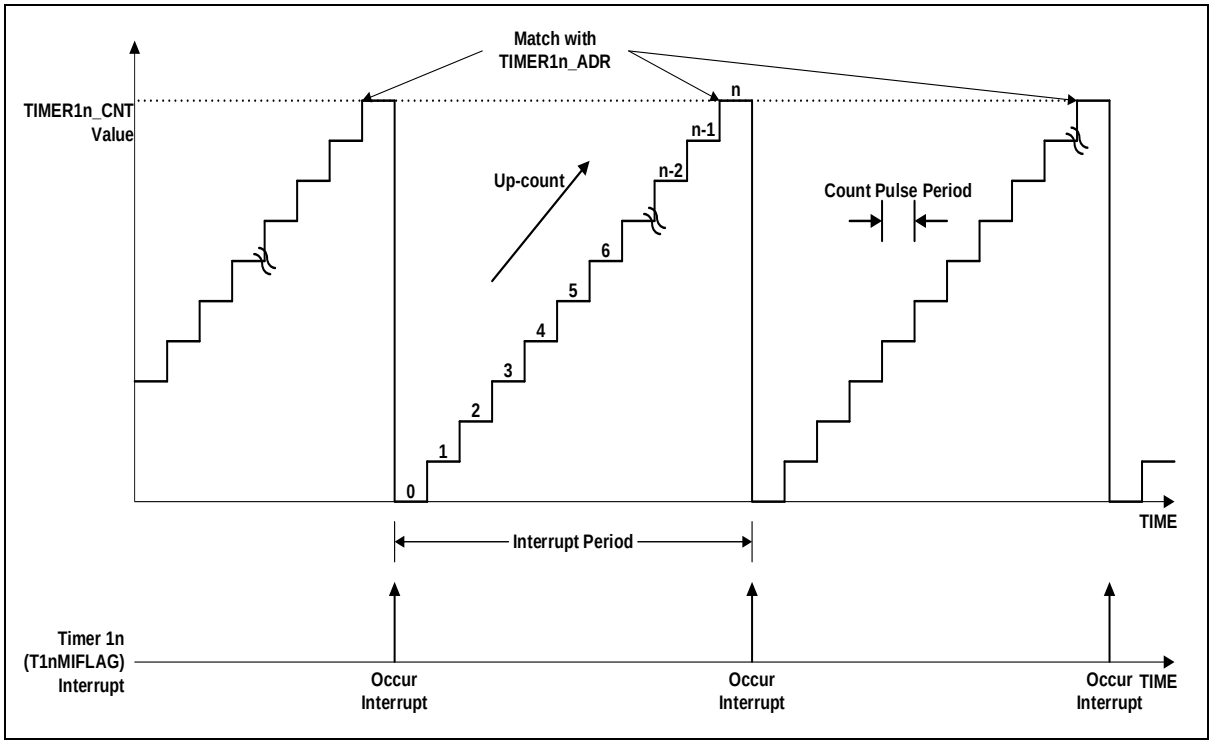


Figure 40. Timer/Counter Mode Timing Example of TIMER n (n = 10, 11, 12, and 13)

10.3.3 16-bit capture mode

Timer n capture mode is evoked by setting TnMS (TIMERn_CR [13:12]) as '01'. It can use an internal clock as a clock source. Basically, it has the same function as 16-bit timer/counter mode, and the interrupt occurs when TIMERn_CNT is equal to TIMERn_ADR. TIMERn_CNT value can be automatically cleared by a match signal. It can be cleared by software too (TnCLR).

A timer interrupt in capture mode is very useful when pulse width of captured signal is wider than the maximum period of the timer. The capture result is loaded into TIMERn_CAPDR. In the timer n capture mode, timer 1n output (TnO) waveform is not available.

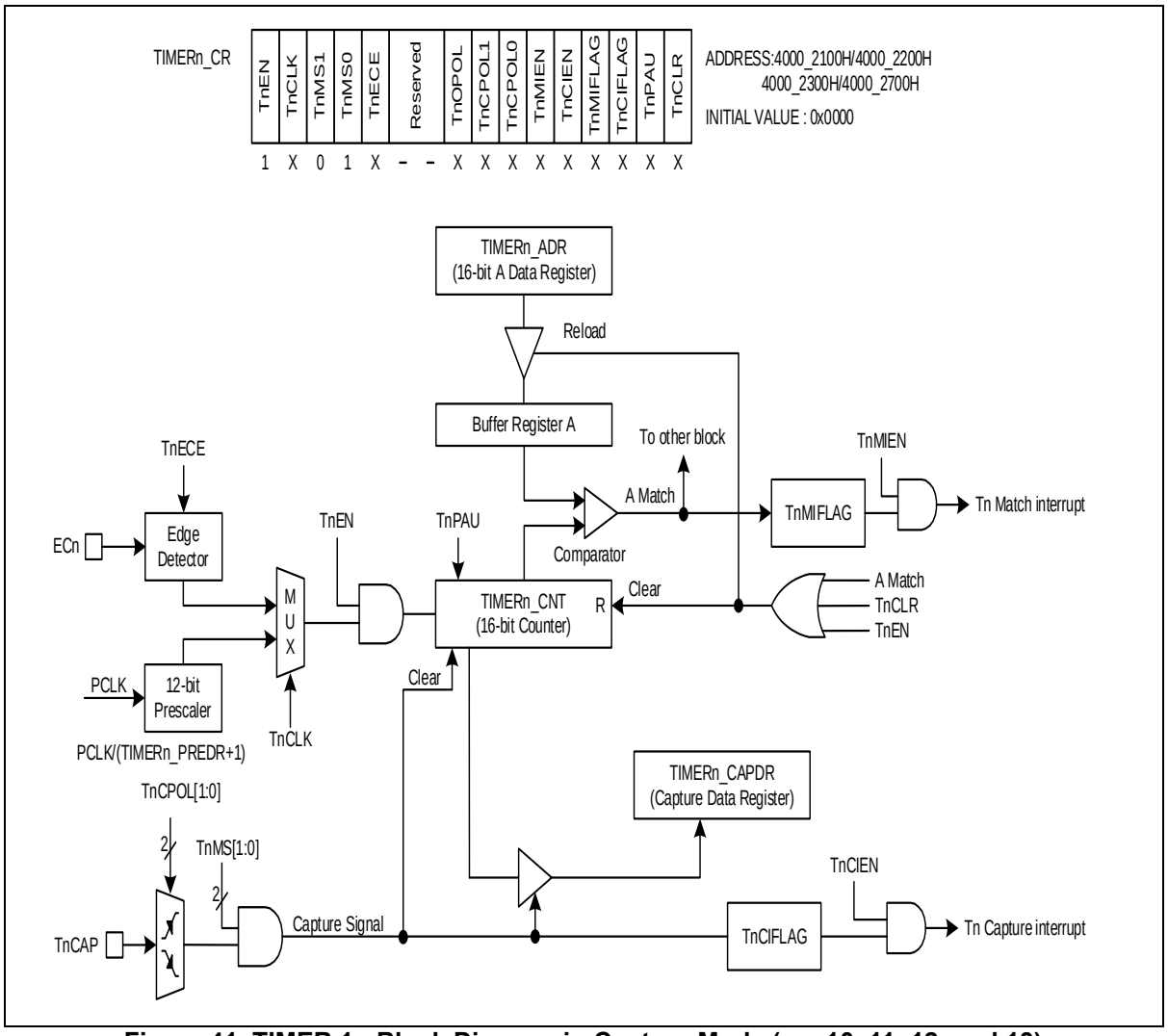


Figure 41. TIMER 1n Block Diagram in Capture Mode (n = 10, 11, 12, and 13)

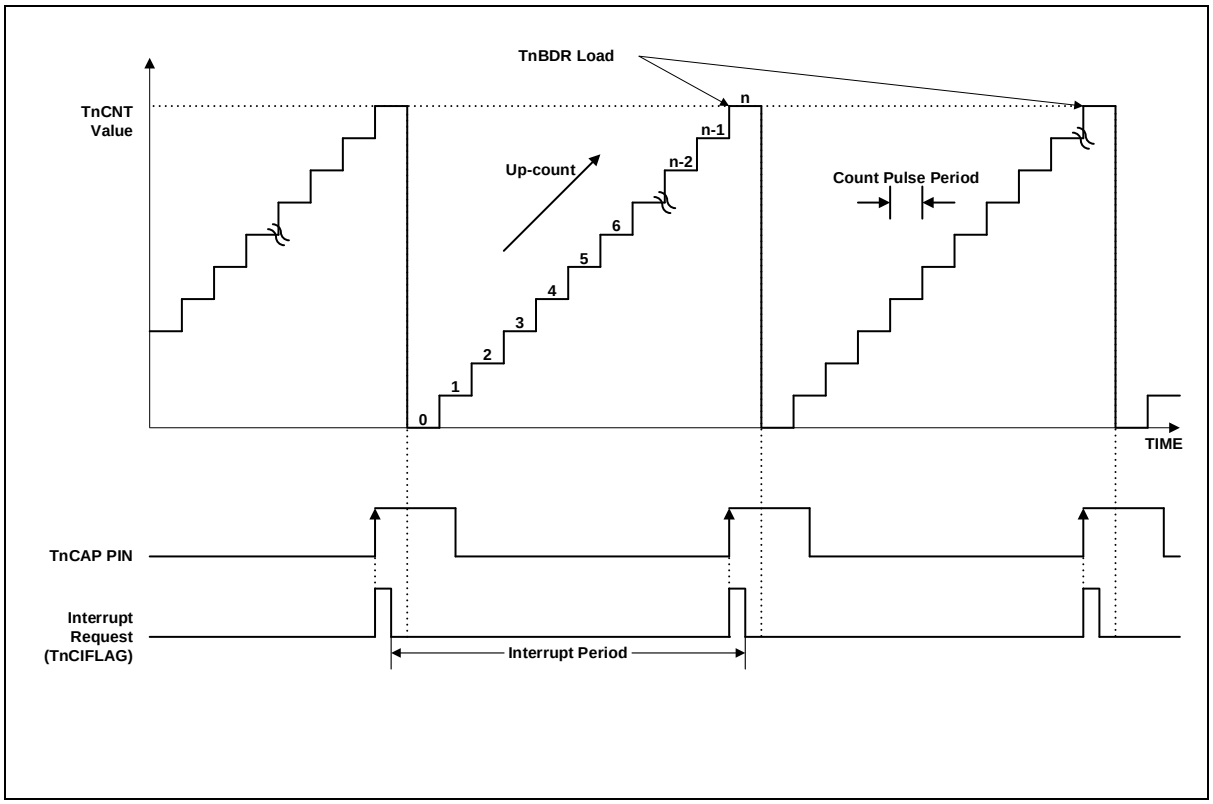


Figure 42. Capture Mode Timing Example of TIMER 1n (n = 10, 11, 12, and 13)

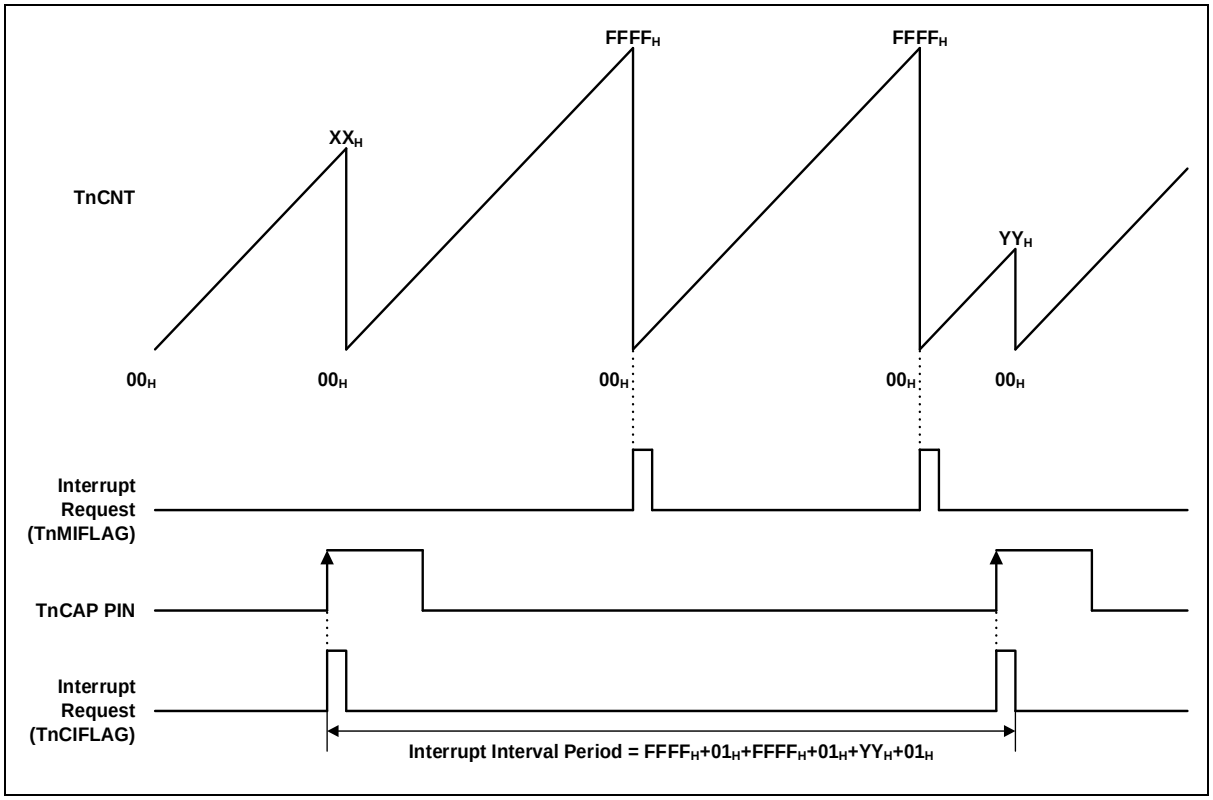
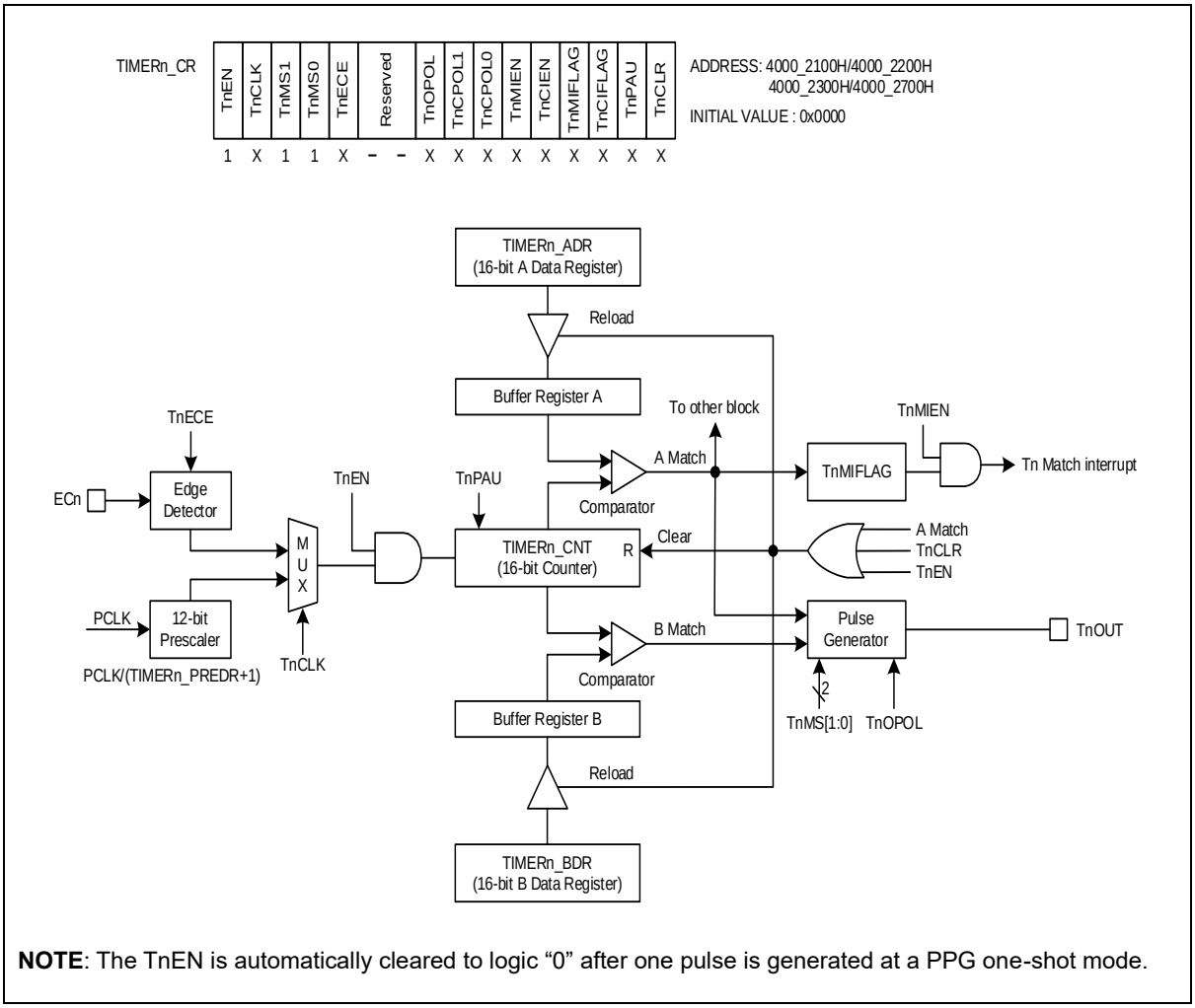


Figure 43. Express Timer Overflow in Capture Mode of TIMER 1n (n = 10, 11, 12, and 13)

10.3.4 16-bit PPG mode

Timer n has a PPG (Programmable Pulse Generation) function. In PPG mode, the TnO pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting corresponding Px_AFSR1 to 'AF1'. Period of the PWM output is determined by the TIMERn_ADR, and duty of the PWM output is determined by the TIMERn_BDR.



NOTE: The TnEN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 44. TIMER 1n Block Diagram in PPG Mode (n = 10, 11, 12, and 13)

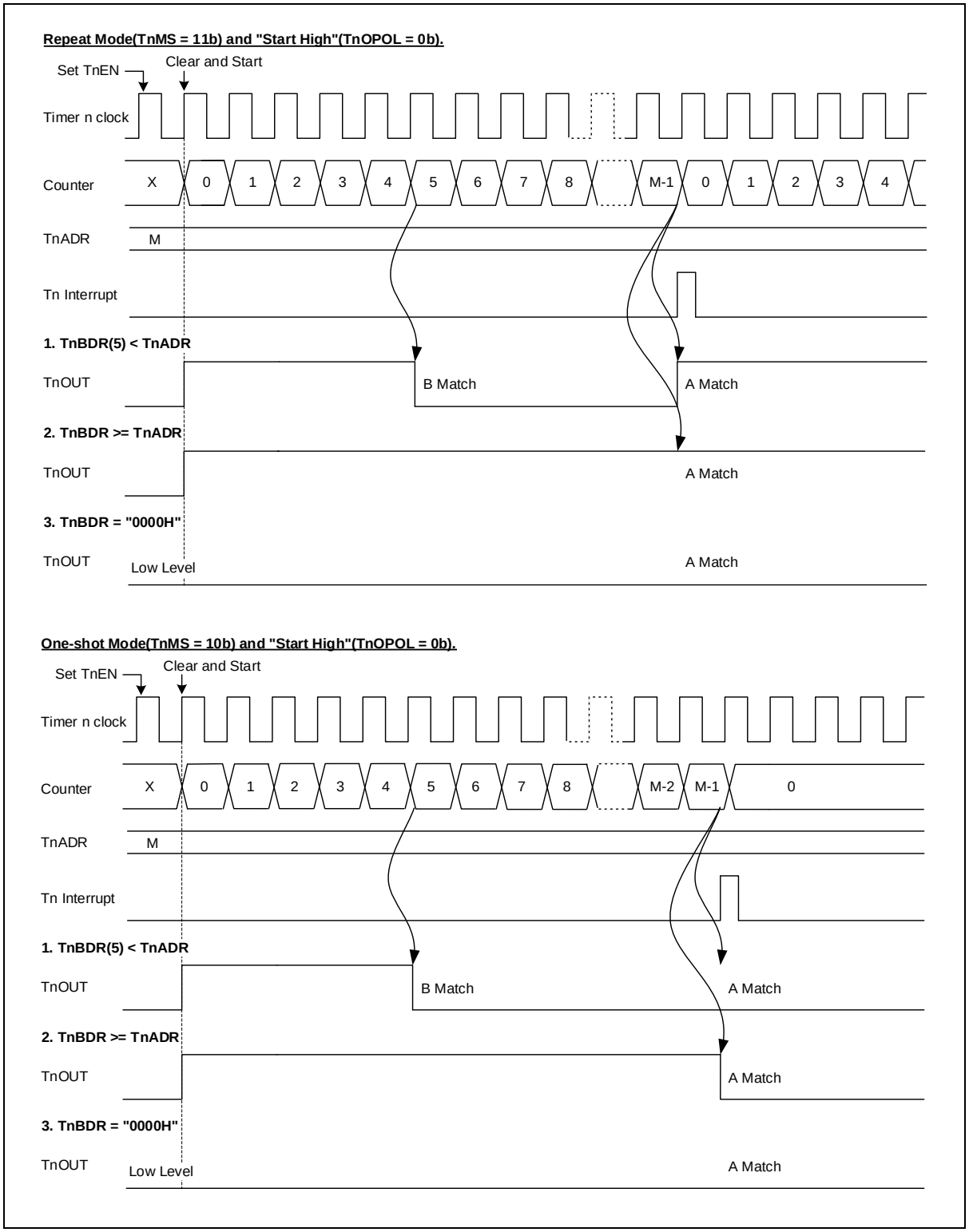


Figure 45. PPG Mode Timing Example of TIMER 1n (n = 10, 11, 12, and 13)

11 32-bit timer20

32-bit timer block comprises 1 channels of 32-bit general purpose timers. Each channel has an independent 32-bit counter and a dedicated prescaler that feeds a counting clock. 32-bit timer supports periodic timer, PWM pulse, one-shot and capture mode. In addition, one more optional free-run timer is provided. Main purpose of 32-bit timer is a periodical tick timer.

32-bit timer of A31G21x series features the followings:

- 32-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- Capture mode
- 12-bit prescaler
- Synchronous start and clear function

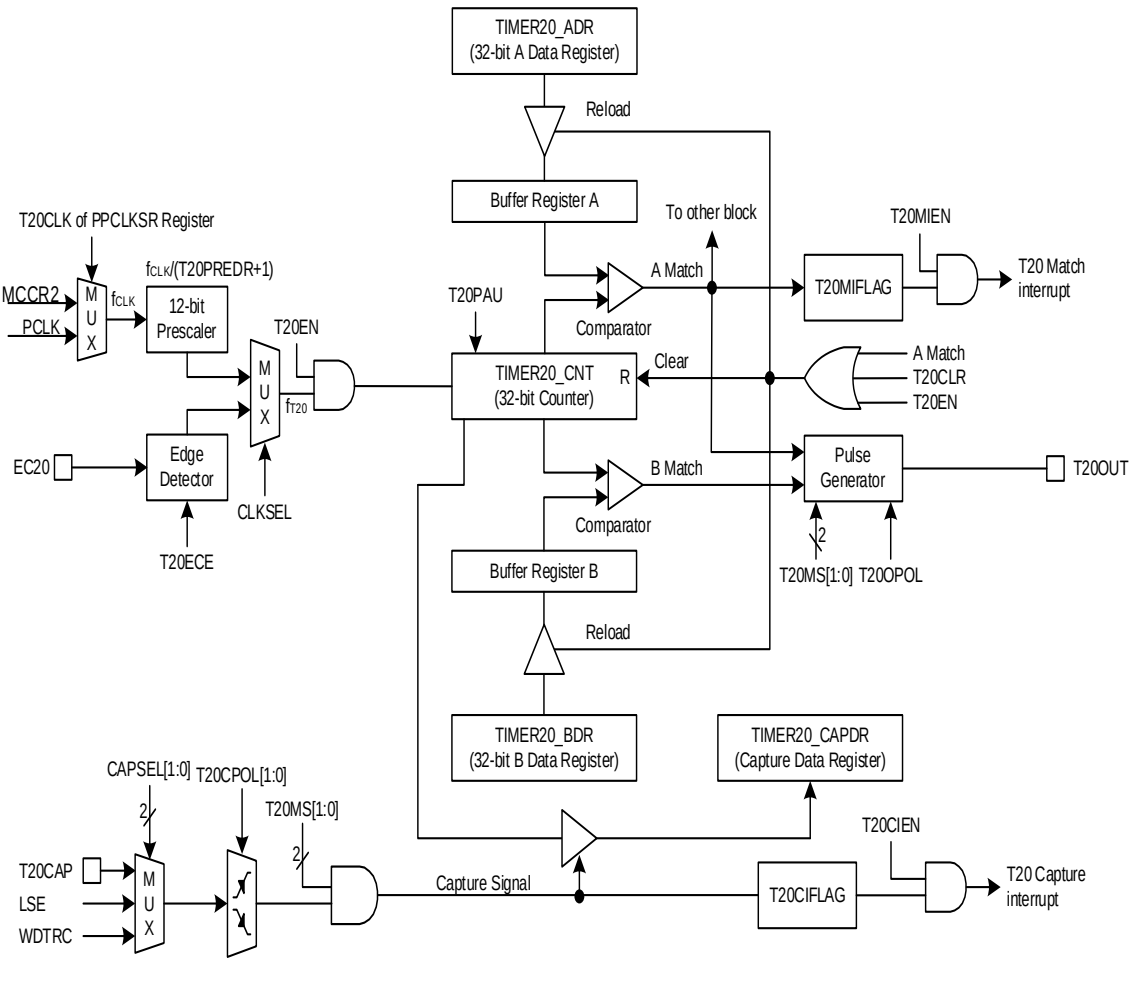
Table 35 introduces pins assigned for 32-bit timer.

Table 35. Pin Assignment of 32-bit Timer: External Pins

Pin name	Type	Description
EC20	I	Timer 20 external clock input
T20C	I	Timer 20 capture input
T20O	O	Timer/PWM/one-shot output

11.1 32-bit timer block diagram

In this section, 32-bit timer is described in a block diagram in Figure 46.



NOTES:

1. The T20EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. After the T20EN bit is set, the counting of TIMER20_CNT may be delayed 2 clocks or more.

Figure 46. 32-bit Timer20 Block Diagram

11.2 Registers

Base address of 32-bit timer is introduced in the followings:

Table 36. Base Address of 32-bit Timer

Name	Base address
TIMER20	0x4000_2500

Table 37. TIMER 20 Register Map

Name	Offset	Type	Description	Reset value	Ref.
TIMER20_CR	0x00	RW	Timer/Counter 20 Control Register	0x0000_0000	11.2.1
TIMER20_ADR	0x04	RW	Timer/Counter 20 A Data Register	0xFFFF_FFFF	11.2.2
TIMER20_BDR	0x08	RW	Timer/Counter 20 B Data Register	0xFFFF_FFFF	11.2.3
TIMER20_CAPDR	0x0C	RO	Timer/Counter 20 Capture Data Register	0x0000_0000	11.2.4
TIMER20_PREDR	0x10	RW	Timer/Counter 20 Prescaler Data Register	0x0000_0FFF	11.2.5
TIMER20_CNT	0x14	RO	Timer/Counter 20 Counter Register	0x0000_0000	11.2.6

11.2.1 TIMER20_CR: timer/counter 20 control register

Timer module should be configured properly before running. When a target purpose is defined, the timer can be configured in the TIMER20_CR register. After configuring TIMER20_CR, a user can start or stop the timer function by using TIMER20_CR.

TIMER20_CR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_CR=0x4000_2500

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		T20EN	T20CLK	T20MS	T20ECE	CAPSEL	T20OPOL	T20CPOL	T20MIEN	T20CIEN	T20MIFLAG	T20CIFLAG	T20PAU	T20CLR
-		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

15	T20EN	Timer 20 Operation Enable bit.
	0	Disable timer 20 operation.
	1	Enable timer 20 operation. (Counter clear and start)
14	T20CLK	Timer 20 Clock Selection bit.
	0	Select an internal prescaler clock.
	1	Select an external clock.
	NOTE: This bit should be changed during T20EN bit is "0b".	
13 12	T20MS	Timer 20 Operation Mode Selection bits.
	00	Timer/Counter mode. (T20OUT: toggle at A-match)
	01	Capture mode. (The A-match interrupt can occur)
	10	PPG one-shot mode. (T20OUT: Programmable pulse output)
	11	PPG repeat mode. (T20OUT: Programmable pulse output)
	NOTE: This bit should be changed during T20EN bit is "0b".	
11	T20ECE	Timer 20 External Clock Edge Selection bit.
	0	Select falling edge of external clock.
	1	Select rising edge of external clock.
10 9	CAPSEL	Timer 20 Capture Signal Selection bits.
	00	Select an external capture signal.
	01	Select the LSE (External sub oscillator) signal.
	10	Select the WDTRC (Watch-dog timer RC oscillator) signal.
	11	Not used
	NOTE: This bit should be changed during T20EN bit is "0b".	
8	T20OPOL	T20OUT Polarity Selection bit.
	0	Start high. (T20OUT is low level at disable)
	1	Start low. (T20OUT is high level at disable)
7 6	T20CPOL	Timer 20 Capture Polarity Selection bits.
	00	Capture on falling edge.
	01	Capture on rising edge.
	10	Capture on both of falling and rising edge.
	11	Reserved.
5	T20MIEN	Timer 20 Match Interrupt Enable bit.
	0	Disable timer 20 match interrupt.
	1	Enable timer 20 match interrupt.
4	T20CIEN	Timer 20 Capture Interrupt Enable bit.
	0	Disable timer 20 capture interrupt.
	1	Enable timer 20 capture interrupt.
3	T20MIFLAG	Timer 20 Match Interrupt Flag bit.
	0	No request occurred.
	1	Request occurred, This bit is cleared to '0' when write '1'.
2	T20CIFLAG	Timer 20 Capture Interrupt Flag bit.
	0	No request occurred.
	1	Request occurred, This bit is cleared to '0' when write '1'.
1	T20PAU	Timer 20 Counter Temporary Pause Control bit.
	0	Continue counting.
	1	Temporary pause.
0	T20CLR	Timer 20 Counter and Prescaler Clear bit.

	0	No effect.
	1	Clear timer 20 counter and prescaler. (Automatically cleared to "0b" after operation)

11.2.2 TIMER20_ADR: timer/counter 20 A data register

TIMER20_ADR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_ADR=0x4000_2504

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																															
0xFFFFFFFF																															
RW																															

31 0	ADATA	Timer/Counter 20 A Data bits. The range is 0x0002 to 0xFFFFFFFF. **NOTE:** Do not write "0000H" in the TIMER20_ADR register when PPG mode.

11.2.3 TIMER20_BDR: timer/counter 20 B data register

TIMER20_BDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_BDR=0x4000_2508

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																															
0xFFFFFFFF																															
RW																															

31 0	BDATA	Timer/Counter 20 B Data bits. The range is 0x0000 to 0xFFFFFFFF.

11.2.4 TIMER20_CAPDR: timer/counter 20 capture data register

TIMER20_ CAPDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_CAPDR=0x4000_250C

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																															
0x00000000																															
RO																															

31 0	CAPD	Timer/Counter 20 Capture Data bits.

11.2.5 TIMER20_PREDR: timer/counter 20 prescaler data register

TIMER20_PREDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_PREDR=0x4000_2510

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	
-		0xFFF	
-		RW	

11 0	PRED	Timer/Counter 20 Prescaler Data bits.

11.2.6 TIMER20_CNT: timer/counter 20 counter register

TIMER20_CNT is a 32-bit register, and able to do 32/16/8-bit access.

TIMER20_CNT=0x4000_2514

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			
0x00000000			
RO			

31 0	CNT	Timer/Counter 20 Counter bits.

11.3 Functional description

11.3.1 Timer counter 20

Timer/counter 20 can be clocked by an internal or an external clock source (EC20). Its clock source is selected by a clock selection logic which is controlled by clock selection bits (T20CLK).

- TIMER 20 clock source: {fCLK/(TIMER20_PREDR +1), EC20

In capture mode, by T20CAP, data is captured into input capture data register (TIMER20_CAPDR). TIMER 20 results the comparison between a counter and the data register through T20OUT port in timer/counter mode. In addition, TIMER 20 outputs PWM wave form through T20OUT port in the PPG mode.

Table 38 introduces various operating modes of TIMER 20 according to the value of timer/counter register.

Table 38. TIMER 20 Operating Modes

T20EN	Alternative mode	T20MS[1:0]	TIMER20_PREDR	TIMER 20
1	PC_AFSR1 = AF1	00	0xXXX	32-bit Timer/Counter Mode
1	PC_AFSR1 = AF2	01	0xXXX	32-bit Capture Mode
1	PC_AFSR1 = AF1	10	0xXXX	32-bit PPG Mode(one-shot mode)
1	PC_AFSR1 = AF1	11	0xXXX	32-bit PPG Mode(repeat mode)

11.3.2 32-bit timer/counter mode

32-bit timer/counter mode is selected by control register as shown in Figure 47. The 32-bit timer has a counter and data register.

The counter register is increased by internal or external clock input. TIMER 20 can use the clock input with 12-bit prescaler division rates (TIMER20_PREDR) and external clock (EC20). When each value of TIMER20_CNT and TIMER20_ADR are identical in TIMER 20, a match signal is generated and the interrupt of Timer 20 occurs. The TIMER20_CNT values are automatically cleared by the match signal. It can be also cleared by software (T20CLR).

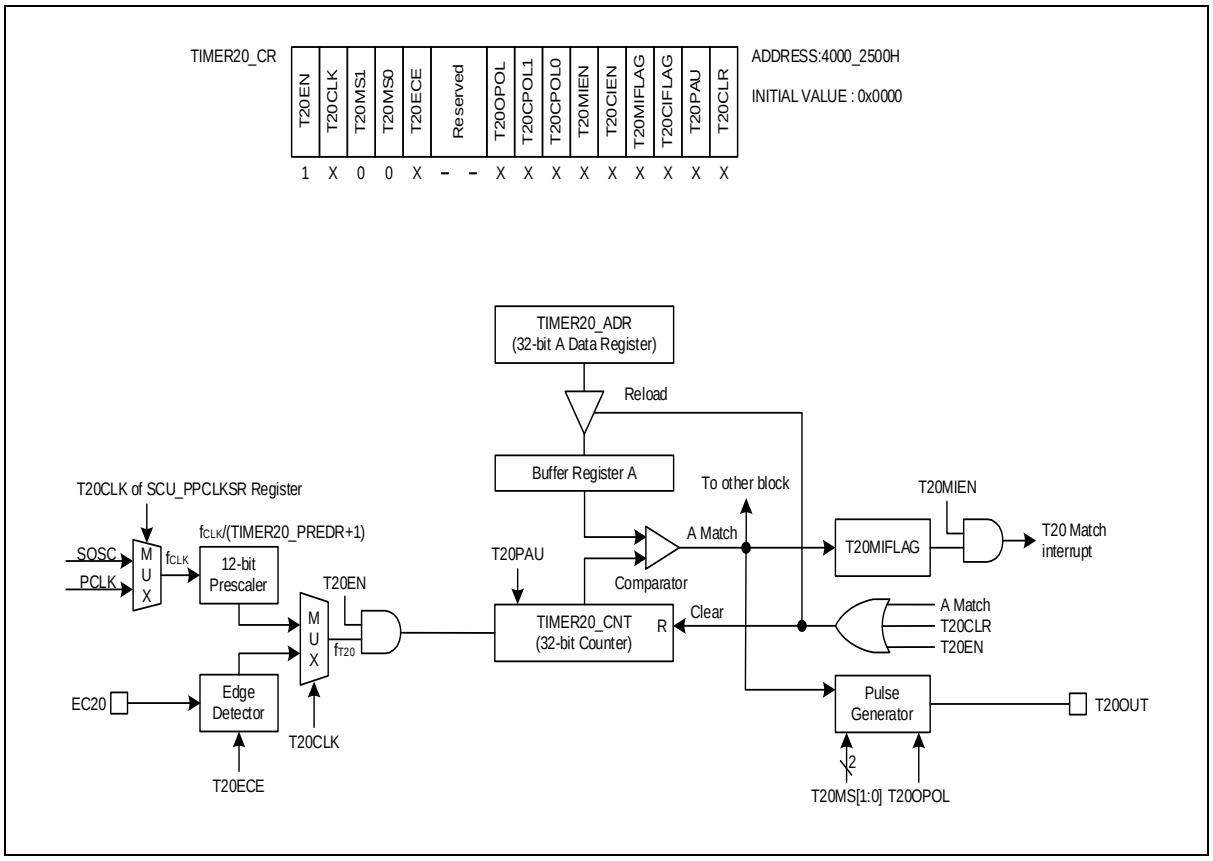


Figure 47. TIMER 20 Block Diagram in Timer/Counter Mode

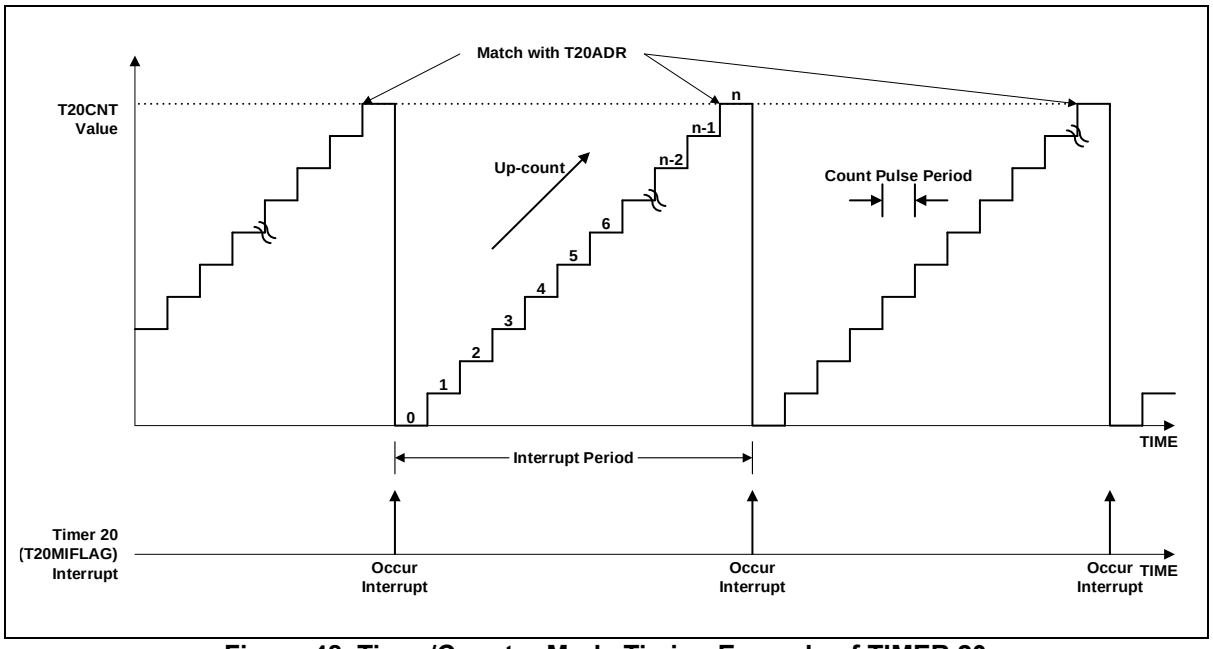


Figure 48. Timer/Counter Mode Timing Example of TIMER 20

11.3.3 32-bit capture mode

Timer 20 capture mode is evoked by setting T20MS[13:12] as '01'. It can use an internal clock as a clock source. Basically, it has the same function as 32-bit timer/counter mode, and the interrupt occurs when TIMER20_CNT is equal to TIMER20_ADR. TIMER20_CNT value can be cleared by software (T20CLR).

A timer interrupt in capture mode is very useful when pulse width of captured signal is wider than the maximum period of the timer. The capture result is loaded into TIMER20_BDR. In the TIMER 20 capture mode, TIMER 20 output (T20OUT) waveform is not available.

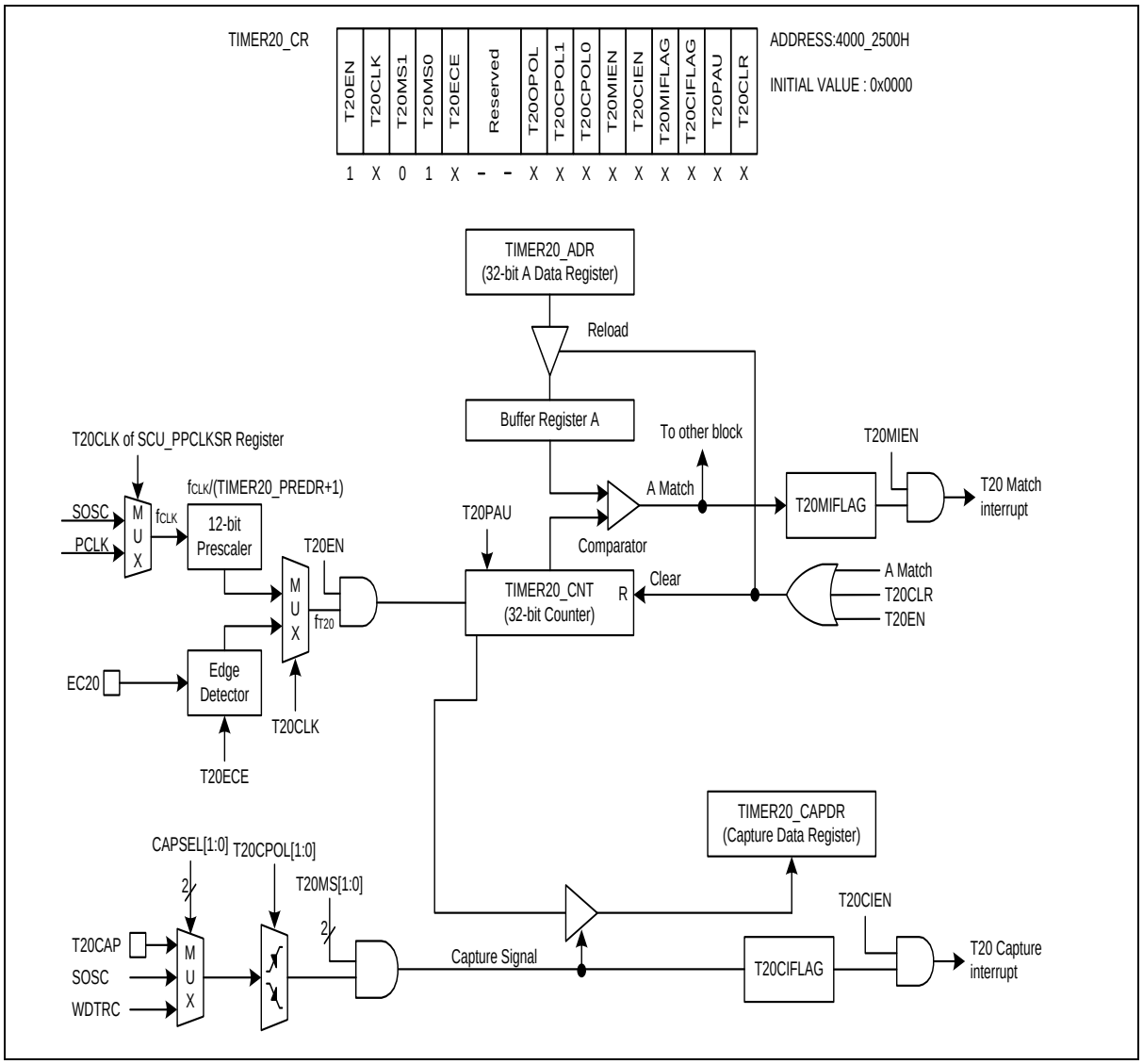


Figure 49. TIMER 20 Block Diagram in Capture Mode

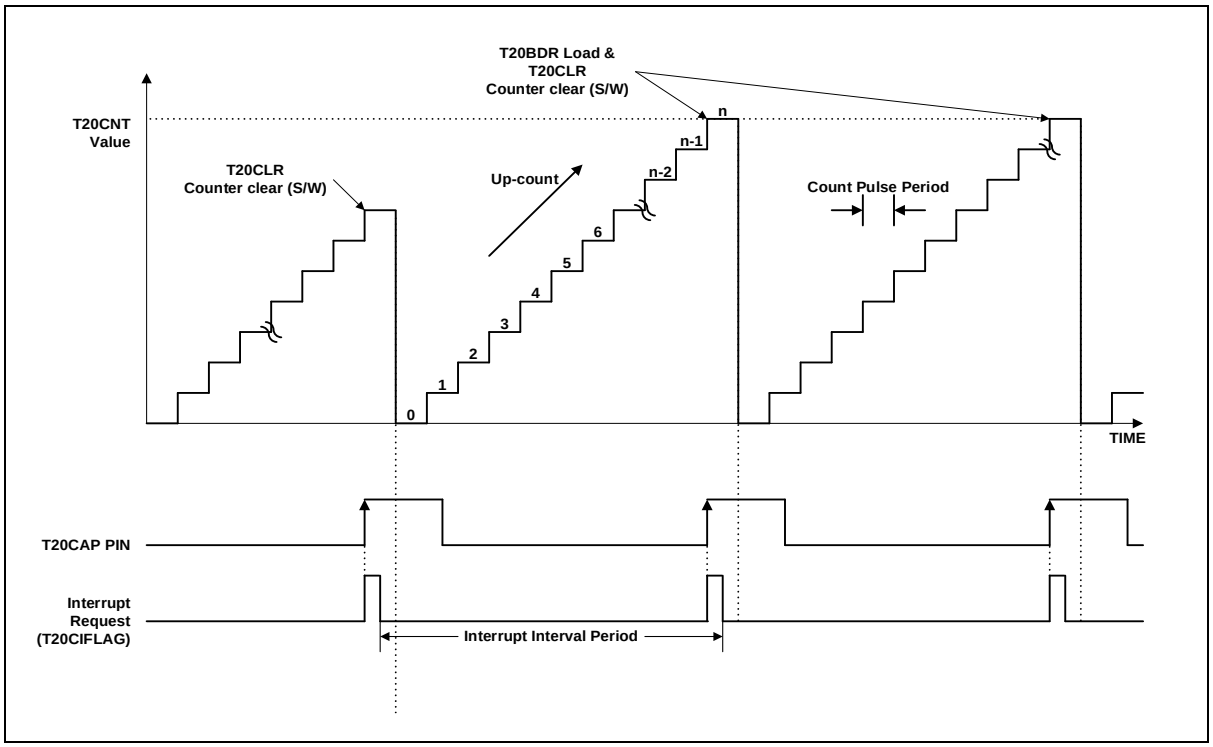


Figure 50. Capture Mode Timing Example of TIMER 20

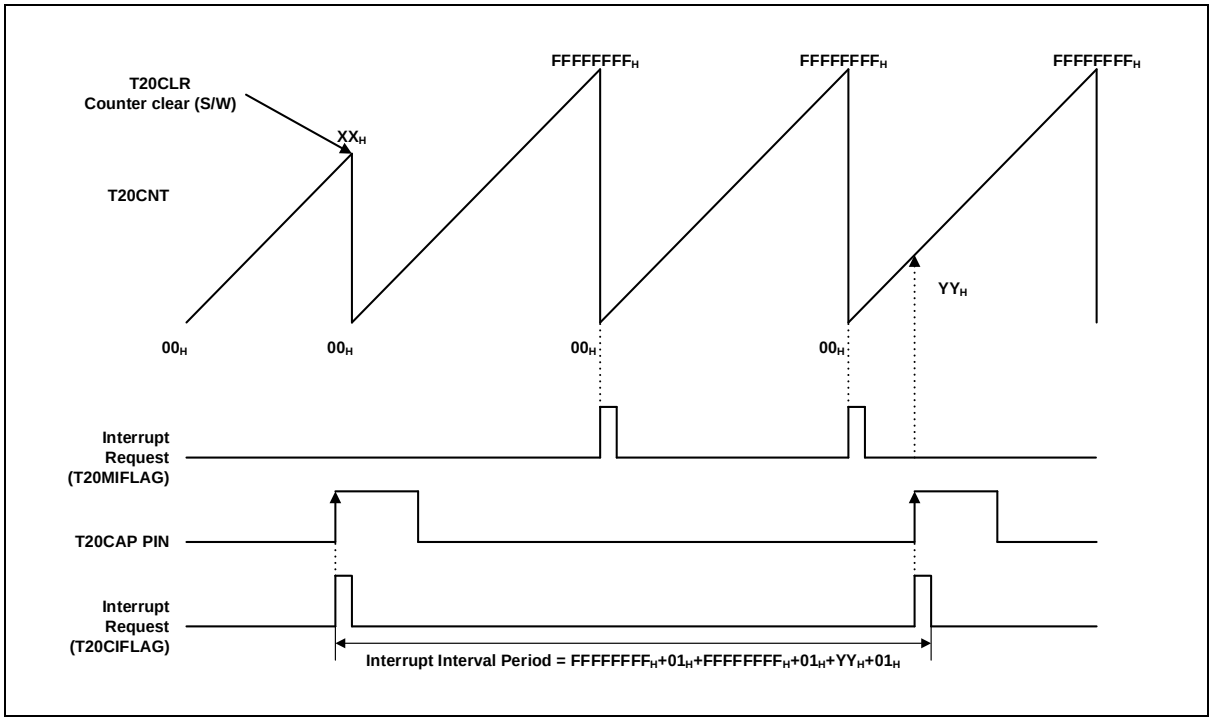
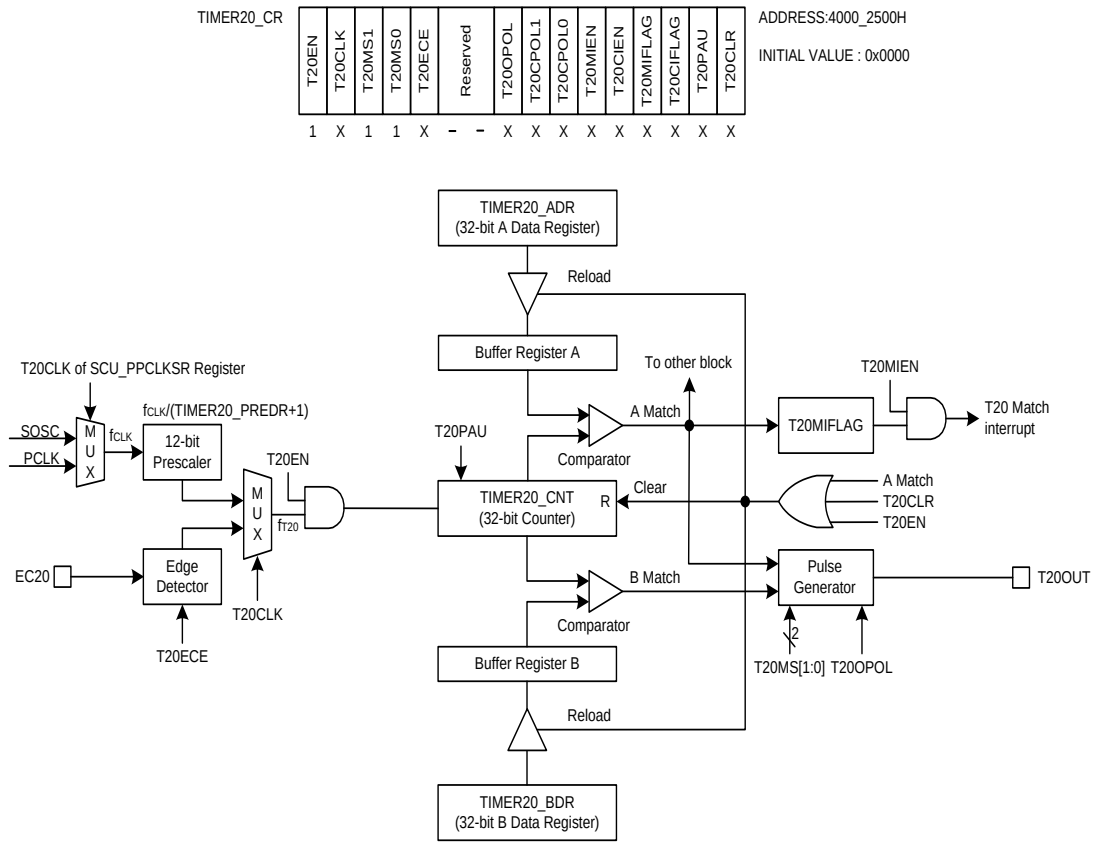


Figure 51. Express Timer Overflow in Capture Mode of TIMER 20

11.3.4 32-bit PPG mode

Timer 20 has a PPG (Programmable Pulse Generation) function. In PPG mode, the T20OUT pin outputs up to 32-bit resolution PWM output. This pin should be configured as a PWM output by setting corresponding PC_AFSRx to 'AF1'. Period of the PWM output is determined by the TIMER20_ADR, and duty of the PWM output is determined by the TIMER20_BDR.



NOTE: The T20EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 52. TIMER 20 Block Diagram in PPG Mode

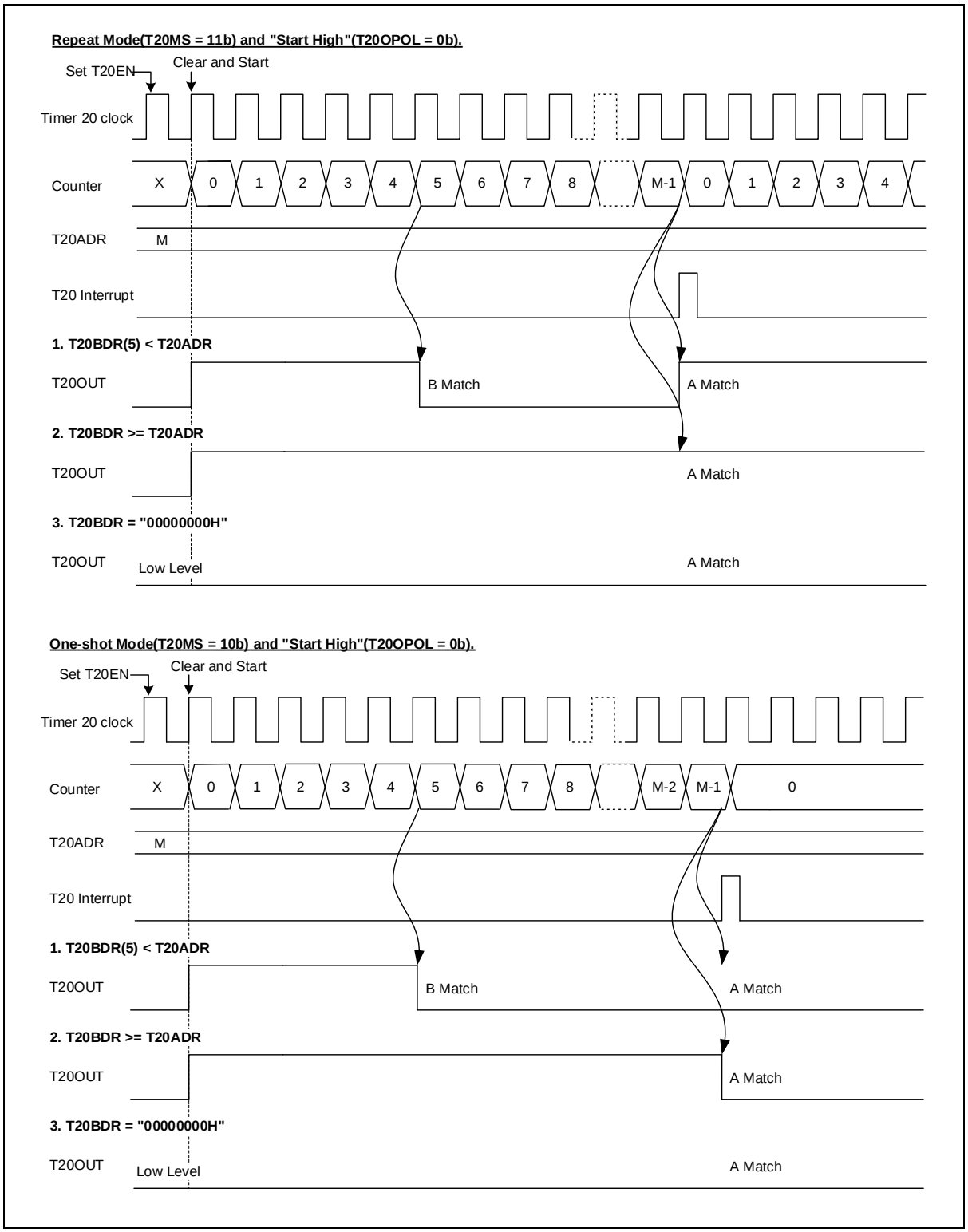


Figure 53. PPG Mode Timing Example of TIMER 20

12 32-bit timer21

32-bit timer block comprises 1 channels of 32-bit general purpose timers. Each channel has an independent 32-bit counter and a dedicated prescaler that feeds a counting clock. 32-bit timer supports periodic timer, PWM pulse, one-shot and capture mode. In addition, one more optional free-run timer is provided. Main purpose of 32-bit timer is a periodical tick timer.

32-bit timer of A31G21x series features the followings:

- 32-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- Capture mode
- 12-bit prescaler
- Synchronous start and clear function

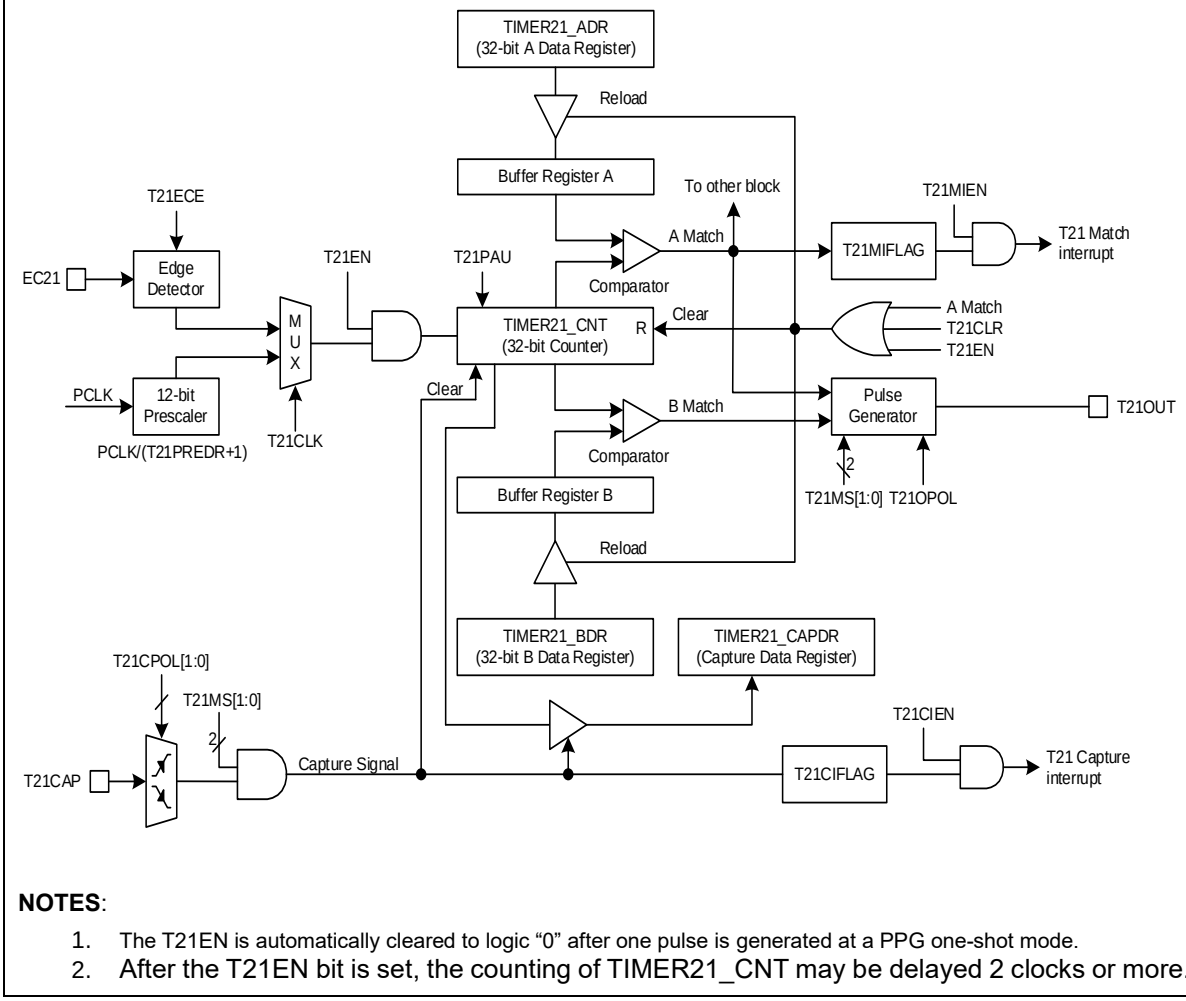
Table 39 introduces pins assigned for 32-bit timer.

Table 39. Pin Assignment of 32-bit Timer: External Pins

Pin name	Type	Description
EC21	I	Timer 21 external clock input
T21C	I	Timer 21 capture input
T21O	O	Timer/PWM/one-shot output

12.1 32-bit timer block diagram

In this section, 32-bit timer is described in a block diagram in Figure 54.



NOTES:

1. The T21EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.
2. After the T21EN bit is set, the counting of TIMER21_CNT may be delayed 2 clocks or more.

Figure 54. 32-bit Timer21 Block Diagram

12.2 Registers

Base address of 32-bit timer is introduced in the followings:

Table 40. Base Address of 32-bit Timer

Name	Base address
TIMER21	0x4000_2600

Table 41. TIMER 20 Register Map

Name	Offset	Type	Description	Reset value	Ref.
TIMER21_CR	0x00	RW	Timer/Counter 21 Control Register	0x0000_0000	12.2.1
TIMER21_ADR	0x04	RW	Timer/Counter 21 A Data Register	0xFFFF_FFFF	12.2.2
TIMER21_BDR	0x08	RW	Timer/Counter 21 B Data Register	0xFFFF_FFFF	12.2.3
TIMER21_CAPDR	0x0C	RO	Timer/Counter 21 Capture Data Register	0x0000_0000	12.2.4
TIMER21_PREDR	0x10	RW	Timer/Counter 21 Prescaler Data Register	0x0000_0FFF	12.2.5
TIMER21_CNT	0x14	RO	Timer/Counter 21 Counter Register	0x0000_0000	12.2.6

12.2.1 TIMER21_CR: timer/counter 20 control register

Timer module should be configured properly before running. When a target purpose is defined, the timer can be configured in the TIMER21_CR register. After configuring TIMER21_CR, a user can start or stop the timer function by using TIMER21_CR.

TIMER21_CR is a 32-bit register, and able to do 32/16/8-bit access.

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		T21EN	T21CLK	T21MS	T21ECE	Reserved	T21OPOL	T21CPOL	T21MIEN	T21CIEN	T21MIFLAG	T21CIFLAG	T21PAU	T21CLR
-		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	T21EN	Timer 21 Operation Enable bit.
	0	Disable timer 21 operation.
	1	Enable timer 21 operation. (Counter clear and start)
14	T21CLK	Timer 21 Clock Selection bit.
	0	Select an internal prescaler clock.
	1	Select an external clock.
	NOTE: This bit should be changed during T21EN bit is "0b".	
13 12	T21MS	Timer 21 Operation Mode Selection bits.
	00	Timer/Counter mode. (T21OUT: toggle at A-match)
	01	Capture mode. (The A-match interrupt can occur)
	10	PPG one-shot mode. (T21OUT: Programmable pulse output)
	11	PPG repeat mode. (T21OUT: Programmable pulse output)
	NOTE: This bit should be changed during T21EN bit is "0b".	
11	T21ECE	Timer 21 External Clock Edge Selection bit.
	0	Select falling edge of external clock.
	1	Select rising edge of external clock.
8	T21OPOL	T21OUT Polarity Selection bit.
	0	Start high. (T21OUT is low level at disable)
	1	Start low. (T21OUT is high level at disable)
7 6	T21CPOL	Timer 21 Capture Polarity Selection bits.
	00	Capture on falling edge.
	01	Capture on rising edge.
	10	Capture on both of falling and rising edge.
	11	Reserved.
5	T21MIEN	Timer 21 Match Interrupt Enable bit.
	0	Disable timer 21 match interrupt.
	1	Enable timer 21 match interrupt.
4	T21CIEN	Timer 21 Capture Interrupt Enable bit.
	0	Disable timer 21 capture interrupt.
	1	Enable timer 21 capture interrupt.
3	T21MIFLAG	Timer 21 Match Interrupt Flag bit.
	0	No request occurred.
	1	Request occurred, This bit is cleared to '0' when write '1'.
2	T21CIFLAG	Timer 21 Capture Interrupt Flag bit.
	0	No request occurred.
	1	Request occurred, This bit is cleared to '0' when write '1'.
1	T21PAU	Timer 21 Counter Temporary Pause Control bit.
	0	Continue counting.
	1	Temporary pause.
0	T21CLR	Timer 21 Counter and Prescaler Clear bit.
	0	No effect.
	1	Clear timer 21 counter and prescaler. (Automatically cleared to "0b" after operation)

12.2.2 TIMER21_ADR: timer/counter 20 A data register

TIMER21_ADR is a 32-bit register, and able to do 32/16/8-bit access.

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																															
0xFFFFFFFF																															
RW																															

31 0	ADATA	Timer/Counter 21 A Data bits. The range is 0x00000002 to 0xFFFFFFFF. **NOTE**: Do not write "0000H" in the TIMER21_ADR register when PPG mode.

12.2.3 TIMER21_BDR: timer/counter 20 B data register

TIMER21_BDR is a 32-bit register, and able to do 32/16/8-bit access.

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																															
0xFFFFFFFF																															
RW																															

31 0	BDATA	Timer/Counter 21 B Data bits. The range is 0x00000000 to 0xFFFFFFFF.

12.2.4 TIMER21_CAPDR: timer/counter 20 capture data register

TIMER21_ CAPDR is a 32-bit register, and able to do 32/16/8-bit access.

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																															
0x00000000																															
RO																															

31 0	CAPD	Timer/Counter 21 Capture Data bits.

12.2.5 TIMER21_PREDR: timer/counter 20 prescaler data register

TIMER21_PREDR is a 32-bit register, and able to do 32/16/8-bit access.

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
-			0xFFF
-			RW

11 0	PRED	Timer/Counter 21 Prescaler Data bits.

12.2.6 TIMER21_CNT: timer/counter 20 counter register

TIMER21_CNT is a 32-bit register, and able to do 32/16/8-bit access.

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			
0x00000000			
RO			

31 0	CNT	Timer/Counter 21 Counter bits.

12.3 Functional description

12.3.1 Timer counter 21

Timer/counter 21 can be clocked by an internal or an external clock source (EC21). Its clock source is selected by a clock selection logic which is controlled by clock selection bits (T21CLK).

- TIMER 21 clock source: {fCLK/(TIMER21_PREDR +1), EC21

In capture mode, by T21CAP, data is captured into input capture data register (TIMER21_CAPDR). TIMER 21 results the comparison between a counter and the data register through T20OUT port in timer/counter mode. In addition, TIMER 21 outputs PWM wave form through T21OUT port in the PPG mode.

Table 42 introduces various operating modes of TIMER 20 according to the value of timer/counter register.

Table 42. TIMER 20 Operating Modes

T21EN	Alternative mode	T21MS[1:0]	TIMER21_PREDR	TIMER 21
1	PC_AFSR1 = AF1	00	0xXXX	32-bit Timer/Counter Mode
1	PC_AFSR1 = AF2	01	0xXXX	32-bit Capture Mode
1	PC_AFSR1 = AF1	10	0xXXX	32-bit PPG Mode (one-shot mode)
1	PC_AFSR1 = AF1	11	0xXXX	32-bit PPG Mode (repeat mode)

12.3.2 32-bit timer/counter mode

32-bit timer/counter mode is selected by control register as shown in Figure 55. The 32-bit timer has a counter and data register.

The counter register is increased by internal or external clock input. TIMER 21 can use the clock input with 12-bit prescaler division rates (TIMER21_PREDR) and external clock (EC21). When each value of TIMER21_CNT and TIMER21_ADR are identical in TIMER 21, a match signal is generated and the interrupt of Timer 20 occurs. The TIMER21_CNT values are automatically cleared by the match signal. It can be also cleared by software (T21CLR).

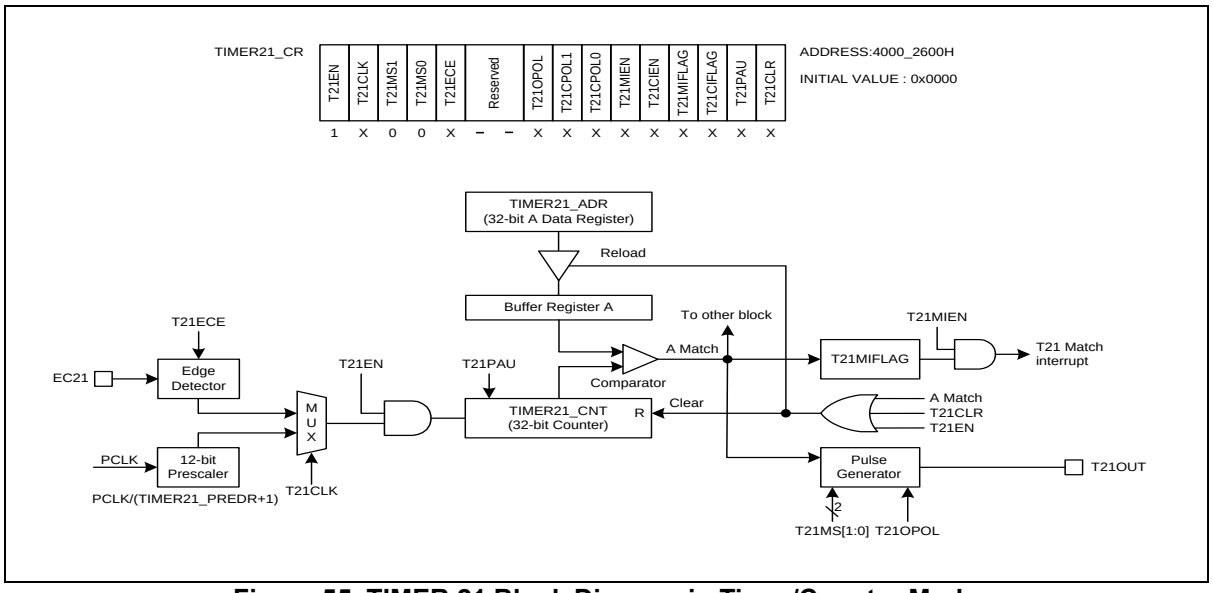


Figure 55. TIMER 21 Block Diagram in Timer/Counter Mode

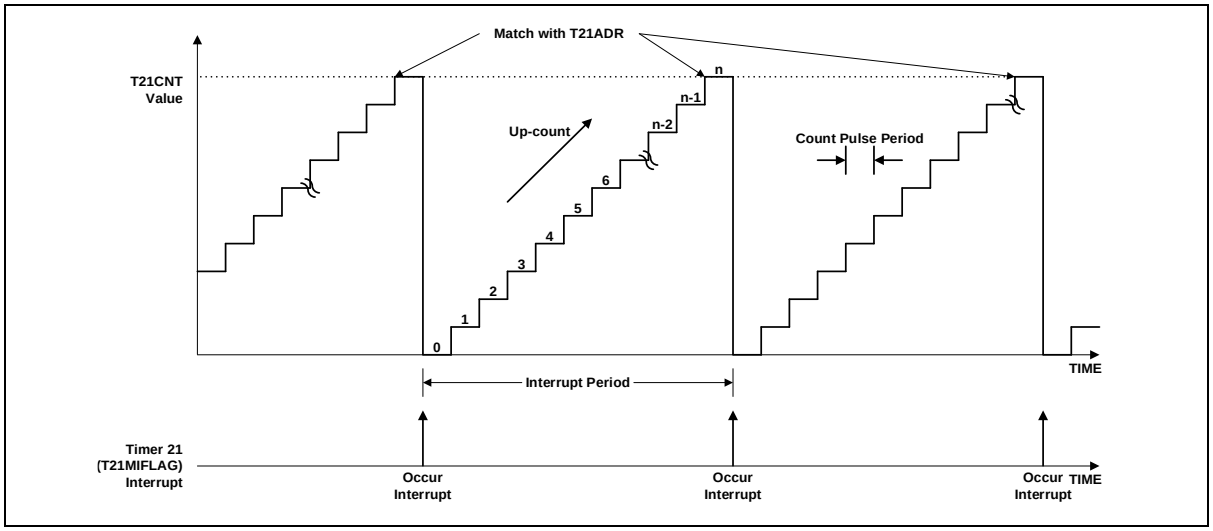


Figure 56. Timer/Counter Mode Timing Example of TIMER 21

12.3.3 32-bit capture mode

Timer 21 capture mode is evoked by setting T21MS[13:12] as '01'. It can use an internal clock as a clock source. Basically, it has the same function as 32-bit timer/counter mode, and the interrupt occurs when TIMER21_CNT is equal to TIMER21_ADR. TIMER21_CNT value can be cleared by software (T21CLR).

A timer interrupt in capture mode is very useful when pulse width of captured signal is wider than the maximum period of the timer. The capture result is loaded into TIMER21_BDR. In the TIMER 21 capture mode, TIMER 21 output (T21OUT) waveform is not available.

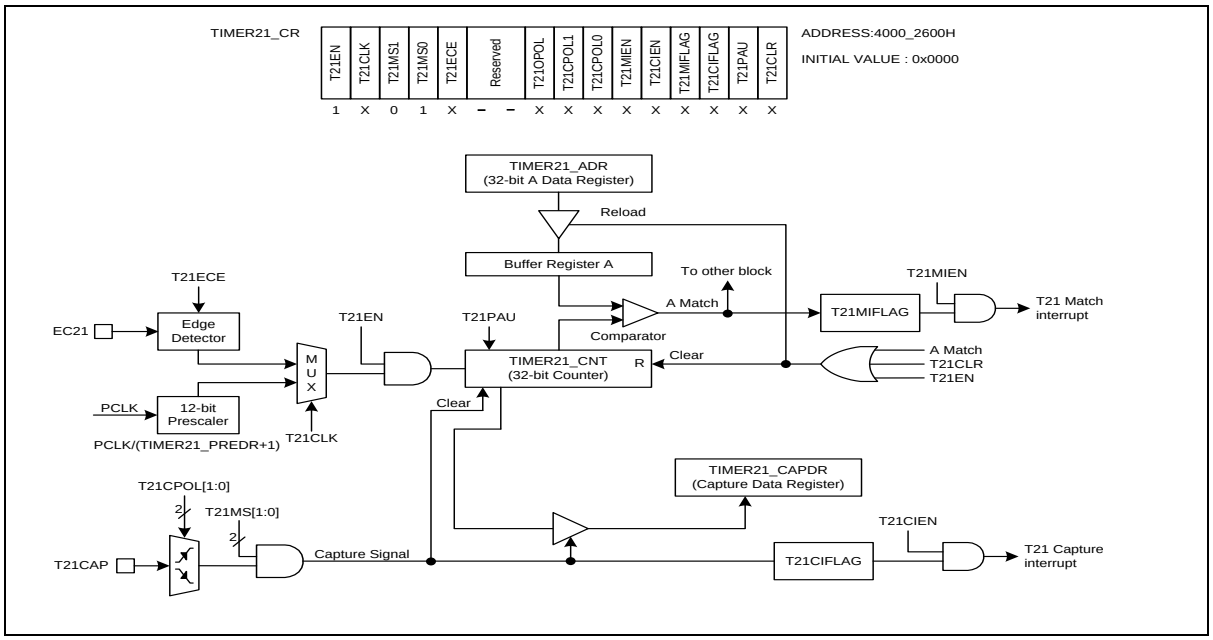


Figure 57. TIMER 21 Block Diagram in Capture Mode

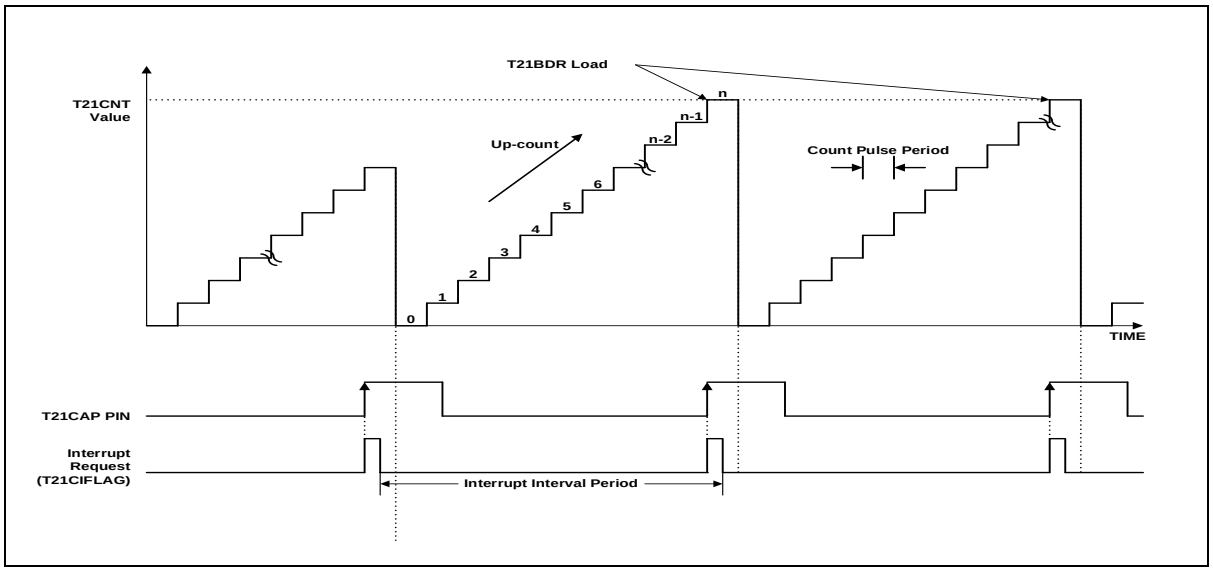


Figure 58. Capture Mode Timing Example of TIMER 21

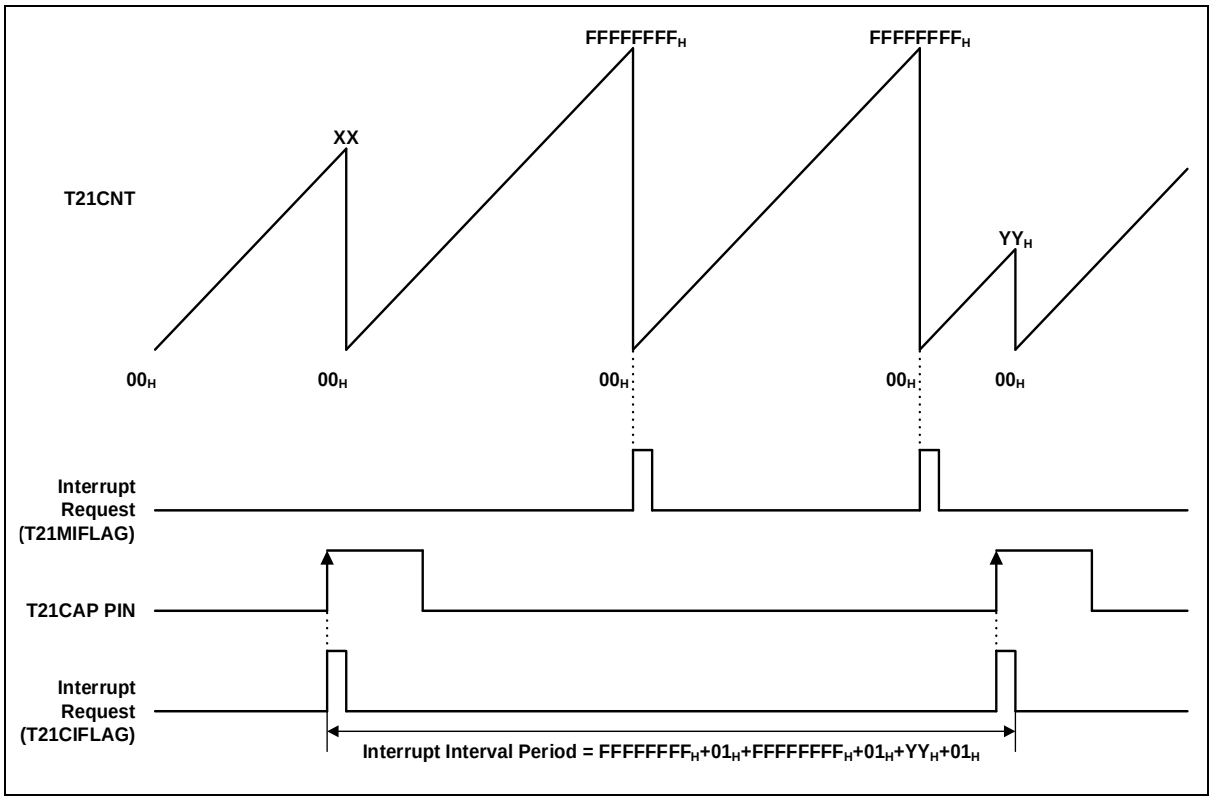
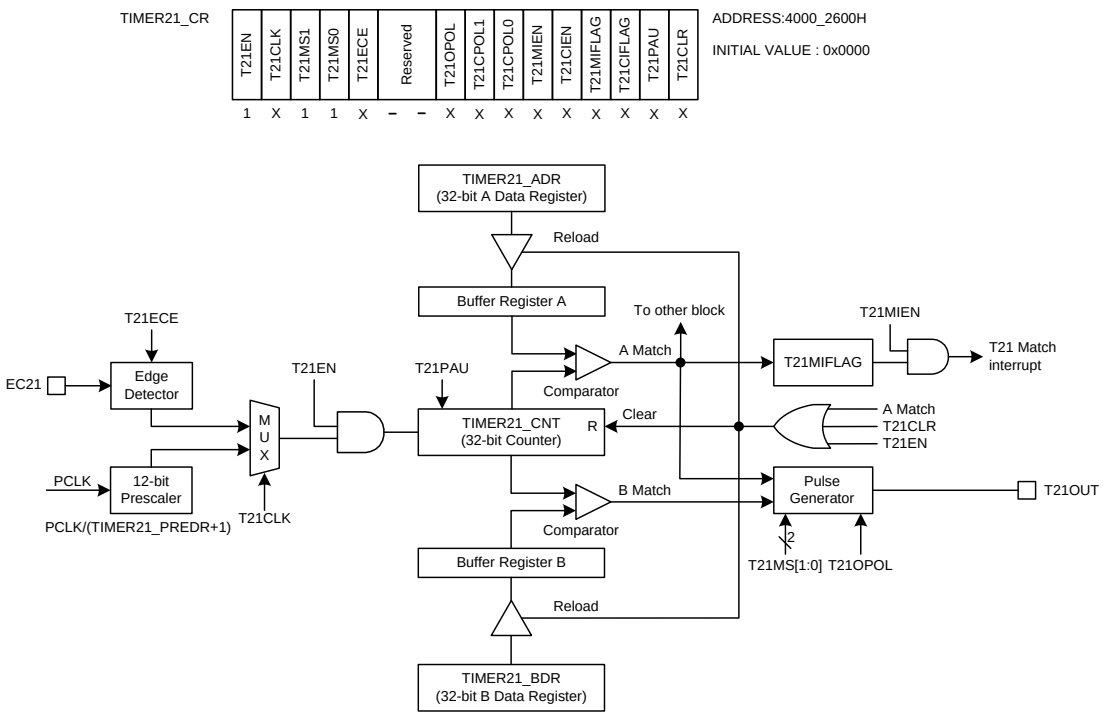


Figure 59. Express Timer Overflow in Capture Mode of TIMER 21

12.3.4 32-bit PPG mode

Timer 21 has a PPG (Programmable Pulse Generation) function. In PPG mode, the T21OUT pin outputs up to 32-bit resolution PWM output. This pin should be configured as a PWM output by setting corresponding PC_AFSRx to 'AF1'. Period of the PWM output is determined by the TIMER21_ADR, and duty of the PWM output is determined by the TIMER21_BDR.



NOTE: The T21EN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 60. TIMER 21 Block Diagram in PPG Mode

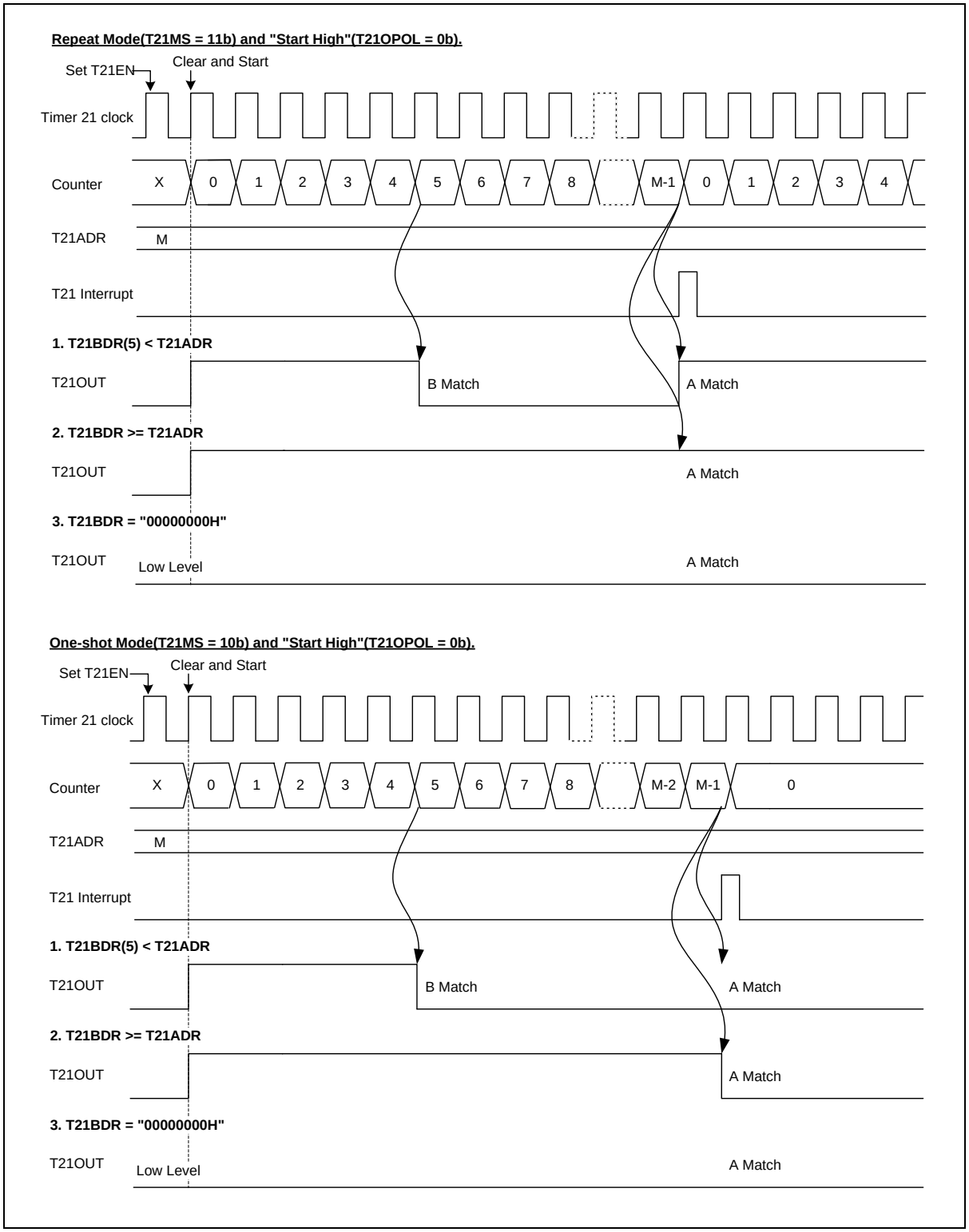


Figure 61. PPG Mode Timing Example of TIMER 21

13 16-bit timer count 30

Timer counter 30 of A31G21x series consist of a multiplexer, a comparator, 16-bit sized timer data registers A/B/C, and many other registers used for its operation. Main features are listed in the followings:

- 16-bit up/down-counter
- Periodic timer mode
- Back-to-Back mode
- Capture mode
- 12-bit prescaler

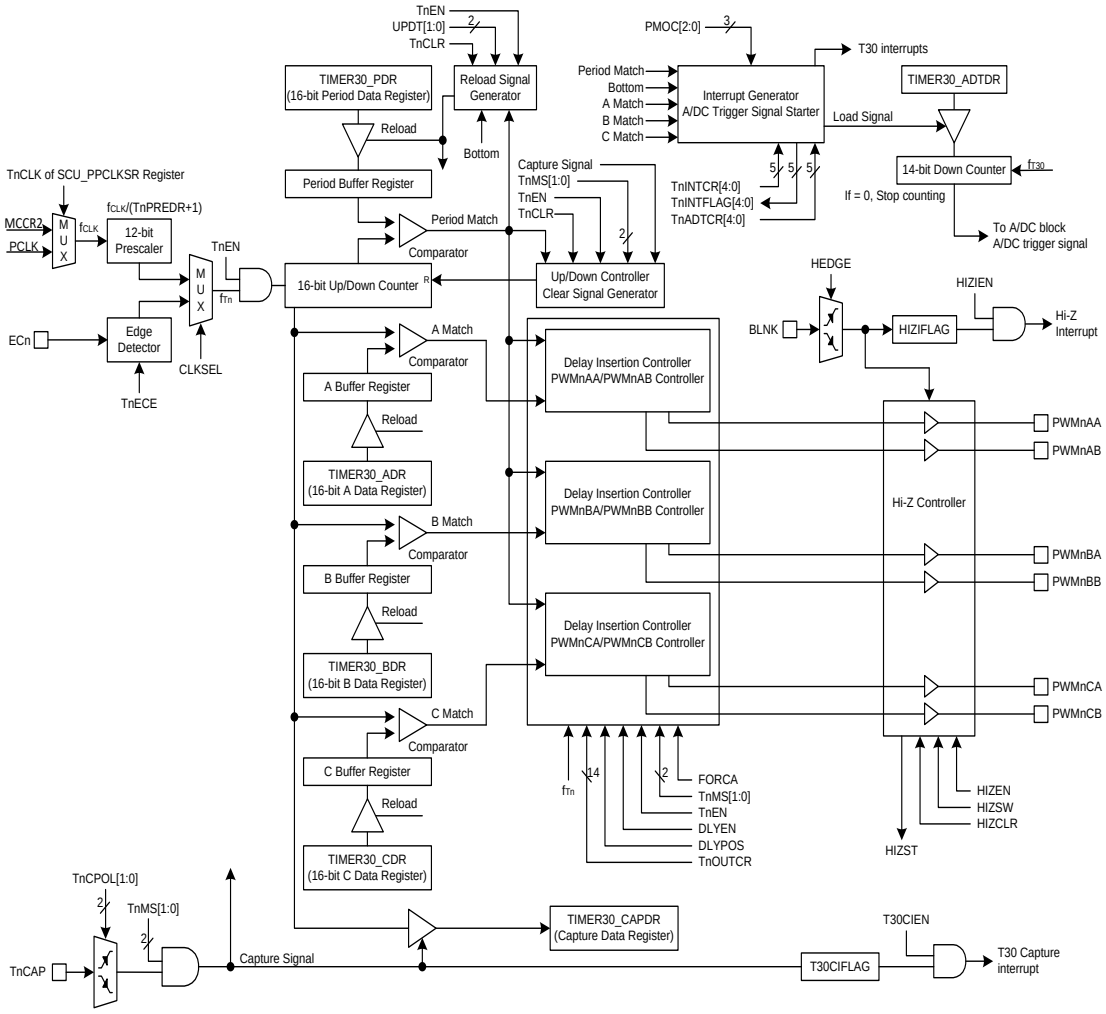
Table 43 introduces pins assigned for the timer counter 30.

Table 43. Pin Assignment of Timer Counter 30: External Pins

Pin name	Type	Description
EC30	I	External clock input
T30CAP	I	Capture input
BLNK	I	External sync signal input
PWM30AA	O	PWM output
PWM30AB	O	PWM output
PWM30BA	O	PWM output
PWM30BB	O	PWM output
PWM30CA	O	PWM output
PWM30CB	O	PWM output

13.1 Timer counter 30 block diagram

In this section, timer counter 30 is introduced in a block diagram.



NOTES:

1. The Period/A/B/C match interrupts except for Bottom can occur in the interval mode.
2. The Period match interrupt only can occur in the capture mode.
3. The 16-bit up/down counter in the capture mode is cleared to "0x0000" after the counter value is loaded to the TIMER30_CAPDR register at valid edge.
4. The HIZSW (high-impedance output software setting) bit can be used for debugging mode. That is, All T30PWM outputs can be high-impedance state with setting the HIZSW bit to '1' by the UI of debugger.
5. After the T30EN bit is set, the counting of TIMER30_CNT may be delayed 2 clocks or more.

Figure 62. Timer Counter 30 Block Diagram

13.2 Registers

Base address of 3-phase PWM timer 30 is introduced in the followings:

Table 44. Base Address of Timer Counter 30

Name	Base address
TIMER30	0x4000_2400

Table 45. Timer Counter 30 Register Map

Name	Offset	Type	Description	Reset value	Ref.
TIMER30_CR	0x0000	RW	Timer/Counter 30 Control Register	0x0000_0000	13.2.1
TIMER30_PDR	0x0004	RW	Timer/Counter 30 Period Data Register	0x0000_FFFF	13.2.2
TIMER30_ADR	0x0008	RW	Timer/Counter 30 A Data Register	0x0000_FFFF	13.2.3
TIMER30_BDR	0x000C	RW	Timer/Counter 30 B Data Register	0x0000_FFFF	13.2.4
TIMER30_CDR	0x0010	RW	Timer/Counter 30 C Data Register	0x0000_FFFF	13.2.5
TIMER30_CAPDR	0x0014	RO	Timer/Counter 30 Capture Data Register	0x0000_0000	13.2.6
TIMER30_PREDR	0x0018	RW	Timer/Counter 30 Prescaler Data Register	0x0000_0FFF	13.2.7
TIMER30_CNT	0x001C	RO	Timer/Counter 30 Counter Register	0x0000_0000	13.2.8
TIMER30_OUTCR	0x0020	RW	Timer/Counter 30 Output Control Register	0x0000_0000	13.2.9
TIMER30_DLY	0x0024	RW	Timer/Counter 30 PWM Output Delay Data Register	0x0000_0000	13.2.10
TIMER30_INTCR	0x0028	RW	Timer/Counter 30 Interrupt Control Register	0x0000_0000	13.2.11
TIMER30_INTFLAG	0x002C	RW	Timer/Counter 30 Interrupt Flag Register	0x0000_0000	13.2.12
TIMER30_HIZCR	0x0030	RW	Timer/Counter 30 High-Impedance Control Register	0x0000_0000	13.2.13

Table 45. Timer Counter 30 Register Map (continued)

Name	Offset	Type	Description	Reset value	Ref.
TIMER30_ADTCR	0x0034	RW	Timer/Counter 30 A/DC Trigger Control Register	0x0000_0000	13.2.14
TIMER30_ADTDR	0x0038	RW	Timer/Counter 30 A/DC Trigger Generator Data Register	0x0000_0000	13.2.15

13.2.1 TIMER30_CR: timer/counter 30 control register

Timer module should be configured properly before running. When target purpose is defined, the timer can be configured in this register. After configuring this register, you can start or stop the timer function by TIMER30_CR register.

TIMER30_CR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CR=0x4000_2400

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		T30EN	T30CLK	T30MS	T30ECE	FORCA	DLYEN	DLYPOS	T30CPOL	UPDT	PMOC	T30CLR
-		0	0	00	0	0	0	0	00	00	000	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	T30EN	Timer 30 Operation Enable bit.
		0 Disable timer 30 operation.
		1 Enable timer 30 operation. (Counter clear and start)
14	T30CLK	Timer 30 Clock Selection bit.
		0 Select an internal prescaler clock.
		1 Select an external clock.
		NOTE: This bit should be changed during T30EN bit is "0b".
13 12	T30MS	Timer 30 Operation Mode Selection bits.
		00 Interval mode. (All match interrupts can occur)
		01 Capture mode. (The Period-match interrupt can occur)
		10 Back-to-back mode. (All interrupts can occur)
		11 Not used.
		NOTE: This bit should be changed during T30EN bit is "0b".
11	T30ECE	Timer 30 External Clock Edge Selection bit.
		0 Select falling edge of external clock.
		1 Select rising edge of external clock.
10	FORCA	Timer 30 Output Mode Selection bit. This bit should be changed when the T30EN is "0b".
		0 6-Channel mode (The PWM30xA/PWM30xB pins are output according to the T30xDR registers, respectively)
		1 Force A-Channel mode (The all PWM30xA/PWM30xB pins are output according to the only TIMER30_ADR registers)
9	DLYEN	Delay Time Insertion Enable bit.
		0 Disable to insert delay time to the PWM30xA/PWM30xB.
		1 Enable to insert delay time to the PWM30xA/PWM30xB.
8	DLYPOS	Delay Time Insertion Position.
		0 Insert at front of PWM30xA and at back of PWM30xB pins.
		1 Insert at back of PWM30xA and at front of PWM30xB pins.
7 6	T30CPOL	Timer 30 Capture Polarity Selection bits.
		00 Capture on falling edge.
		01 Capture on rising edge.
		10 Capture on both of falling and rising edge.
		11 Reserved
6 4	UPDT	Data Reload Time Selection bits.
		00 Update data to buffer at the time of writing.
		01 Update data to buffer at period match.
		10 Update data to buffer at bottom.
		11 Not used.
3 1	PMOC	Period Match Interrupt Occurrence Selection.
		000 Once every period match.

		001	Once every 2 period match.
		010	Once every 3 period match.
		011	Once every 4 period match.
		100	Once every 5 period match.
		101	Once every 6 period match.
		110	Once every 7 period match.
		111	Once every 8 period match.
		NOTE:	A period match counter is cleared as 0x00 when the T3nCLR bit is set.
0	T30CLR	Timer 30 Counter and Prescaler Clear bit.	
		0	No effect.
		1	Clear timer 30 counter and prescaler (Automatically cleared to "0b" after operation)

13.2.2 TIMER30_PDR: timer/counter 30 period data register

TIMER30_PDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_PDR=0x4000_2404

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		PDATA	
-		0xFFFF	
-		RW	

15 0	PDATA	Timer/Counter 30 Period Data bits. The range is 0x0002 to 0xFFFF. **NOTE:** Do not write "0000H" in the TIMER30_PDR register when PPG mode.

13.2.3 TIMER30_ADR: timer/counter 30 A data register

TIMER30_ADR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_ADR=0x4000_2408

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	
-		0xFFFF	
-		RW	

15 0	ADATA	Timer/Counter 30 A Data bits. The range is 0x0000 to 0xFFFF.

13.2.4 TIMER30_BDR: timer/counter 30 B data register

TIMER30_BDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_BDR=0x4000_240C

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	
-		0xFFFF	
-		RW	

15 0	BDATA	Timer/Counter 30 B Data bits. The range is 0x0000 to 0xFFFF.

13.2.5 TIMER30_CDR: timer/counter 30 C data register

TIMER30_CDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CDR=0x4000_2410

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		CDATA	
-		0xFFFF	
-		RW	

15 0	CDATA	Timer/Counter 30 C Data bits. The range is 0x0000 to 0xFFFF.

13.2.6 TIMER30_CAPDR: timer/counter 30 capture data register

TIMER30_CAPDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CAPDR=0x4000_2414

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		CAPD	
-		0x0000	
-		RO	

15 0	CAPD	Timer/Counter 30 Capture Data bits.

13.2.7 TIMER30_PREDR: timer/counter 30 prescaler data register

TIMER30_PREDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_PREDR=0x4000_2418

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
-			0xFFF
-			RW

11 0	PRED	Timer/Counter 30 Prescaler Data bits.

13.2.8 TIMER30_CNT: timer/counter 30 counter register

TIMER30_CNT is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_CNT=0x4000_241C

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	
-		0x0000	
-		RO	

15 0	CNT	Timer/Counter 30 Counter bits.

13.2.9 TIMER30_OUTCR: timer/counter 30 output control register

TIMER30_OUTCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_OUTCR=0x4000_2420

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		POLB	POLA	PABOE	PBBOE	PCBOE	PAAOE	PBAOE	PCAOE	Reserved	LVLAB	LVLBB	LVLCB	Reserved	LVLAA	LVLBA	LVLCA
0x0000		0	0	0	0	0	0	0	0	-	0	0	0	-	0	0	0
WO		RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	-	RW	RW	RW

31 16	WTIDKY	Write Identification Key. On writes, write 0xE06C to these bits, otherwise the write is ignored.
15	POLB	PWM30xB Output Polarity Selection bit. (x : A, B and C) 0 Low level start. (The PWM30xB pins are started with low level after counting) 1 High level start. (The PWM30xB pins are started with high level after counting)
14	POLA	PWM30xA Output Polarity Selection bit. (x : A, B and C) 0 Low level start. (The PWM30xA pins are started with low level after counting) 1 High level start. (The PWM30xA pins are started with high level after counting)
13	PABOE	PWM30AB Output Enable bit. 0 Disable output. 1 Enable output.
12	PBBOE	PWM30BB Output Enable bit. 0 Disable output. 1 Enable output.
11	PCBOE	PWM30CB Output Enable bit. 0 Disable output. 1 Enable output.
10	PAAOE	PWM30AA Output Enable bit. 0 Disable output. 1 Enable output.
9	PBAOE	PWM30BA Output Enable bit. 0 Disable output. 1 Enable output.
8	PCAOE	PWM30CA Output Enable bit. 0 Disable output. 1 Enable output.
6	LVLAB	Configure PWM30AB output When Disable. 0 Low level 1 High level
5	LVLBB	Configure PWM30BB output When Disable. 0 Low level 1 High level
4	LVLCB	Configure PWM30CB output When Disable. 0 Low level 1 High level
2	LVLAA	Configure PWM30AA output When Disable. 0 Low level 1 High level
1	LVLBA	Configure PWM30BA output When Disable. 0 Low level 1 High level
0	LVLCA	Configure PWM30CA output When Disable. 0 Low level 1 High level

13.2.10 TIMER30_DLY: timer/counter 30 PWM output delay data register

TIMER30_DLY is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_DLY=0x4000_2424

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			DLY
-			0x000
-			RW

9	DLY	Timer/Counter 30 PWM Delay Data bits. Delay time:
0		(DLY[9:0]+1)÷fT30

13.2.11 TIMER30_INTCR: timer/counter 30 interrupt control register

TIMER30_INTCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_INTCR=0x4000_2428

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				HIZIEN	T30CIEN	T30BTIEN	T30PMIEN	T30AMIEN	T30BMIEN	T30CMIEN
-				0	0	0	0	0	0	0
-				RW	RW	RW	RW	RW	RW	RW

6	HIZIEN	Timer 30 Output High-Impedance Interrupt Enable bit.	
		0	Disable timer 30 output high-impedance interrupt.
		1	Enable timer 30 output high-impedance interrupt.
5	T30CIEN	Timer 30 Capture Interrupt Enable bit.	
		0	Disable timer 30 capture interrupt.
		1	Enable timer 30 capture interrupt.
4	T30BTIEN	Timer 30 Bottom Interrupt Enable bit.	
		0	Disable timer 30 period interrupt.
		1	Enable timer 30 period interrupt.
3	T30PMIEN	Timer 30 Period Match Interrupt Enable bit.	
		0	Disable timer 30 period interrupt.
		1	Enable timer 30 period interrupt.
2	T30AMIEN	Timer 30 A-ch Match Interrupt Enable bit.	
		0	Disable timer 30 A-ch match interrupt.
		1	Enable timer 30 A-ch match interrupt.
1	T30BMIEN	Timer 30 B-ch Match Interrupt Enable bit.	
		0	Disable timer 30 B-ch match interrupt.
		1	Enable timer 30 B-ch match interrupt.
0	T30CMIEN	Timer 30 C-ch Match Interrupt Enable bit.	
		0	Disable timer 30 C-ch match interrupt.
		1	Enable timer 30 C-ch match interrupt.

13.2.12 TIMER30_INTFLAG: timer/counter 30 interrupt flag register

TIMER30_INTFLAG is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_INTFLAG=0x4000_242C

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				HIZIFLAG	T30CIFLAG	T30BTIFLAG	T30PMIFLAG	T30AMIFLAG	T30BMIFLAG	T30CMIFLAG
-				0	0	0	0	0	0	0
-				RW	RW	RW	RW	RW	RW	RW

6	HIZIFLAG	Timer 30 Output High-Impedance Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
5	T30CIFLAG	Timer 30 Capture Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
4	T30BTIFLAG	Timer 30 Bottom Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
3	T30PMIFLAG	Timer 30 Period Match Flag Enable bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
2	T30AMIFLAG	Timer 30 A-ch Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
1	T30BMIFLAG	Timer 30 B-ch Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
0	T30CMIFLAG	Timer 30 C-ch Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.

13.2.13 TIMER30_HIZCR: timer/counter 30 high-impedance control register

TIMER30_HIZCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_HIZCR=0x4000_2430

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			HIZEN	Reserved	HIZSW	Reserved	HEDGE	HIZSTA	HIZCLR
-			0	-	0	0	0	0	0
-			RW	-	RW	RW	RW	RW	RW

7	HIZEN	PWM30xA/PWM30xB Output High-Impedance Enable bit.	
		0	Disable to control the output high-impedance.
		1	Enable to control the output high-impedance.
4	HIZSW	High-Impedance Output Software Setting.	
		0	No effect.
		1	PWM30xA/PWM30xB pins go into high impedance. (Automatically cleared to "0b" after operation)
2	HEDGE	High-Impedance Edge Selection.	
		0	Falling edge of the BLNK pin.
		1	Rising edge of the BLNK pin.
1	HIZSTA	High-Impedance Status.	
		0	Indicates that the pins are not under a Hi-Z state.
		1	Indicates that the pins are under a Hi-Z state.
0	HIZCLR	High-Impedance Output Clear bit.	
		0	No effect.
		1	Clear high-impedance output. (The PWM30xA/PWM30xB pins are back to output and this bit is automatically cleared to "0b" after operation)

13.2.14 TIMER30_ADTCR: timer/counter 30 A/DC trigger control register

TIMER30_ADTCR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_ADTCR=0x4000_2434

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				T30BTTG	T30PMTG	T30AMTG	T30BMTG	T30CMTG
-				0	0	0	0	0
-				RW	RW	RW	RW	RW

4	T30BTTG	Select Timer 30 Bottom for A/DC Trigger Signal Generator.
	0	Disable A/DC trigger signal generator by bottom.
	1	Enable A/DC trigger signal generator by bottom.
3	T30PMTG	Select Timer 30 Period Match for A/DC Trigger Signal Generator.
	0	Disable A/DC trigger signal generator by period match.
	1	Enable A/DC trigger signal generator by period match.
2	T30AMTG	Select Timer 30 A-ch Match for A/DC Trigger Signal Generator.
	0	Disable A/DC trigger signal generator by A-ch match.
	1	Enable A/DC trigger signal generator by A-ch match.
1	T30BMTG	Select Timer 30 B-ch Match for A/DC Trigger Signal Generator.
	0	Disable A/DC trigger signal generator by B-ch match.
	1	Enable A/DC trigger signal generator by B-ch match.
0	T30CMTG	Select Timer 30 C-ch Match for A/DC Trigger Signal Generator.
	0	Disable A/DC trigger signal generator by C-ch match.
	1	Enable A/DC trigger signal generator by C-ch match.

NOTES:

1. A trigger signal generation is not related with the PMOC[2:0] bits of TIMER30_CR register.
2. If several source for trigger is selected, a signal can be lost in case of the trigger generation counter is reloaded by another signal.

13.2.15 TIMER30_ADTDR: timer/counter 30 A/DC trigger generator data register

TIMER30_ADTDR is a 32-bit register, and able to do 32/16/8-bit access.

TIMER30_ADTDR=0x4000_2438

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			ADTDATA
-			0x0000
-			RW

13 0	CNT	Timer/Counter 30 A/DC Trigger Generation Data bits.

13.3 Functional description

13.3.1 Timer counter 30

The timer/counter 30 can be clocked by an internal or an external clock source (EC30). The clock source is selected by a clock selection logic which is controlled by the clock selection bits (T30CLK).

- TIMER 30 clock source: {PCLK/(TIMER30_PREDR +1), EC30

In capture mode, by T30CAP, data is captured into input capture data register (TIMER30_CAPDR).

The PWM wave form to PWM30AA, PWM30AB, PWM30BA, PWM30BB, PWM30CA, PWM3CB Port (6-channel).

Table 46. Timer 30 Operating Modes

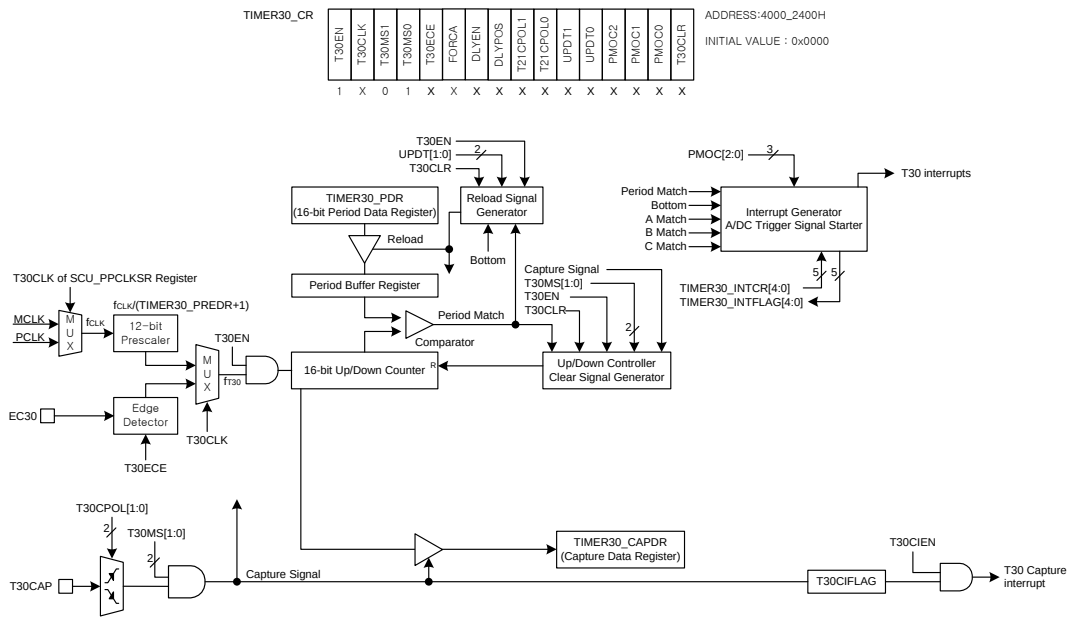
T30EN	T30MS[1:0]	TIMER30_PREDR	Timer 30 MODE
1	00	0xXXX	16-bit Interval Mode
1	01	0xXXX	16-bit Capture Mode
1	10	0xXXX	16-bit back-to-back Mode

13.3.2 Timer 30 capture mode

16-bit timer 30 capture mode is set by configuring T30MS[1:0] as '01'. An internal clock input or an external clock input can be used as a clock source. Basically, the 16-bit timer 30 capture mode has the same function as the 16-bit interval mode has. Interrupts occur when value of TIMER30's 16-bit up/down counter equals to the one of TIMER30_PDR. The 16-bit up/down counter values are automatically cleared by a match signal. It can be cleared by software (T30CLR) too.

The 16-bit timer 30's interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of the timer. The capture result is loaded into TIMER30_CAPDR.

Figure 63 shows 16-bit capture mode of the timer 30.



NOTES:

1. The Period match only can occur in the capture mode.
2. The 16-bit up/down counter in the capture mode is cleared to "0x0000" after the counter value is loaded to the TIMER30_CAPDR register at valid edge.

Figure 63. 16-bit Capture Mode of Timer 30

13.3.3 Timer 30 interval mode

Timer 30 interval mode is set by configuring T30MS[1:0] as '00'. The timer 30 has a counter and data registers. The 16-bit up/down counter is increased by an internal or an external clock input. The timer 30 can use the input clock with 12-bit prescaler division rates (TIMER30_PREDR[11:0]). When the value of TIMER30 16-bit up/down counter and the value of TIMER30_PDR are identical in timer 30, a match signal is generated and the period match interrupt of timer 30 is occurred. The period match interrupt can be occurred which once every 1, 2, 3, 4, 5, 6, 7, or 8 period match (PMOC[2:0]). The 16-bit up/down counter value is automatically cleared by match signal. It can be cleared by software (T30CLR) too.

The timer 30 Interval mode can be operated for BLDC motor control. It has 6-channel pins output up to 16-bit resolution PWM output. When the value of 16-bit up/down counter and TIMER30_PDR are identical in timer 30, a period match signal is generated and the period match interrupt of timer 30 is occurred.

The timer 30 A, B, and C match signals are generated and the A, B, and C match interrupts of timer 30 are occurred, when the 16-bit counter value are identical to the value of TIMER30_xDR. The period and duty of the PWM output is determined by the TIMER30_PDR (PWM period register), and T3xDR (each channel PWM duty register).

- PWM Period = [TIMER30_PDR] X Source Clock
- PWM Duty(A-ch) = [TIMER30_ADR] X Source Clock
- PWM Duty(B-ch) = [TIMER30_BDR] X Source Clock
- PWM Duty(C-ch) = [TIMER30_CDR] X Source Clock

The POLA/POLB bit of TIMER30_OUTCR register decides the polarity of PWM output. If the POLA/POLB bit is set to '1b', the PWM30xA/PWM30xB output is high level start, respectively. And if the POLA/POLB bit is cleared to '0b', the PWM30xA/PWM30xB output is low level start, respectively.

Table 47. PWM Channel Polarity

PxAOE	PxBOE	POLxA	POLxB	PWM3xA Pin put	PWM3xB Pin Output
1	1	0	0	Low level start	Low level start
		0	1	Low level start	High level start
		1	0	High level start	Low level start
		1	1	High level start	High level start

NOTE: Where x = A, B, and C.

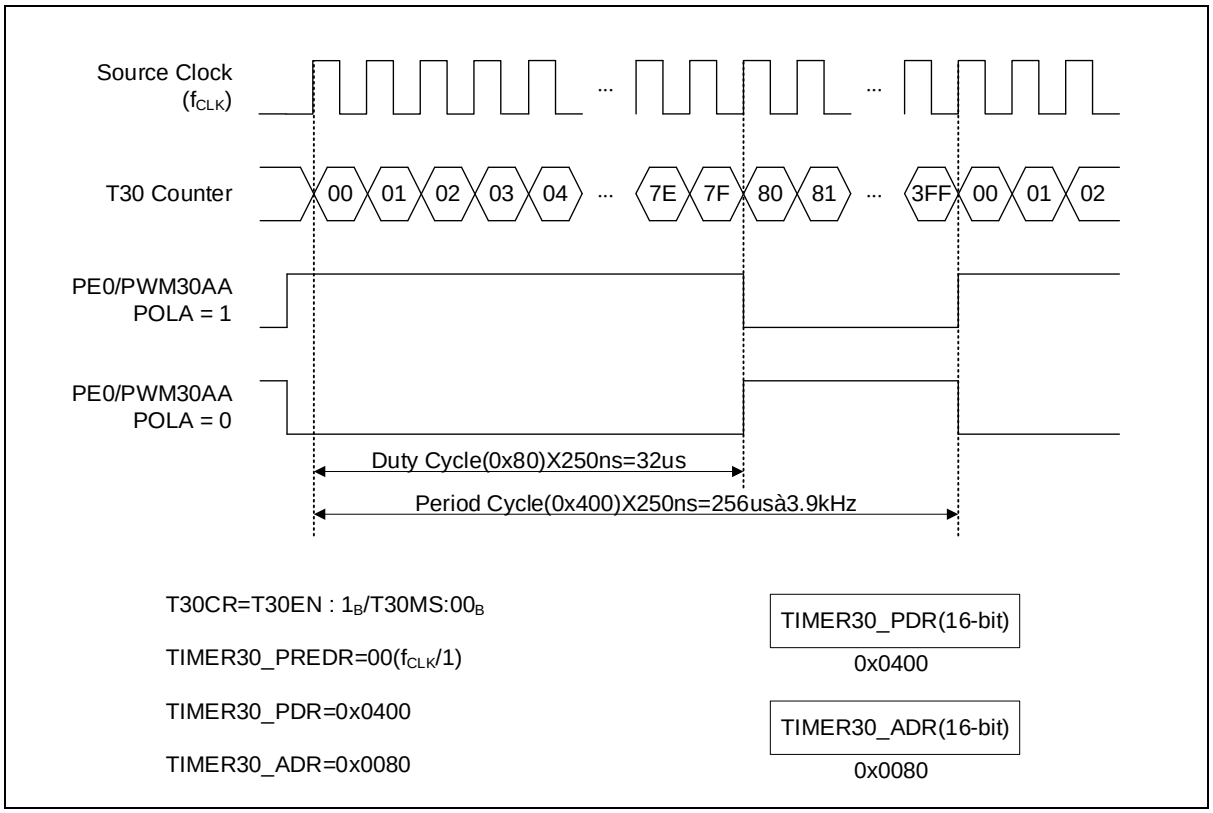
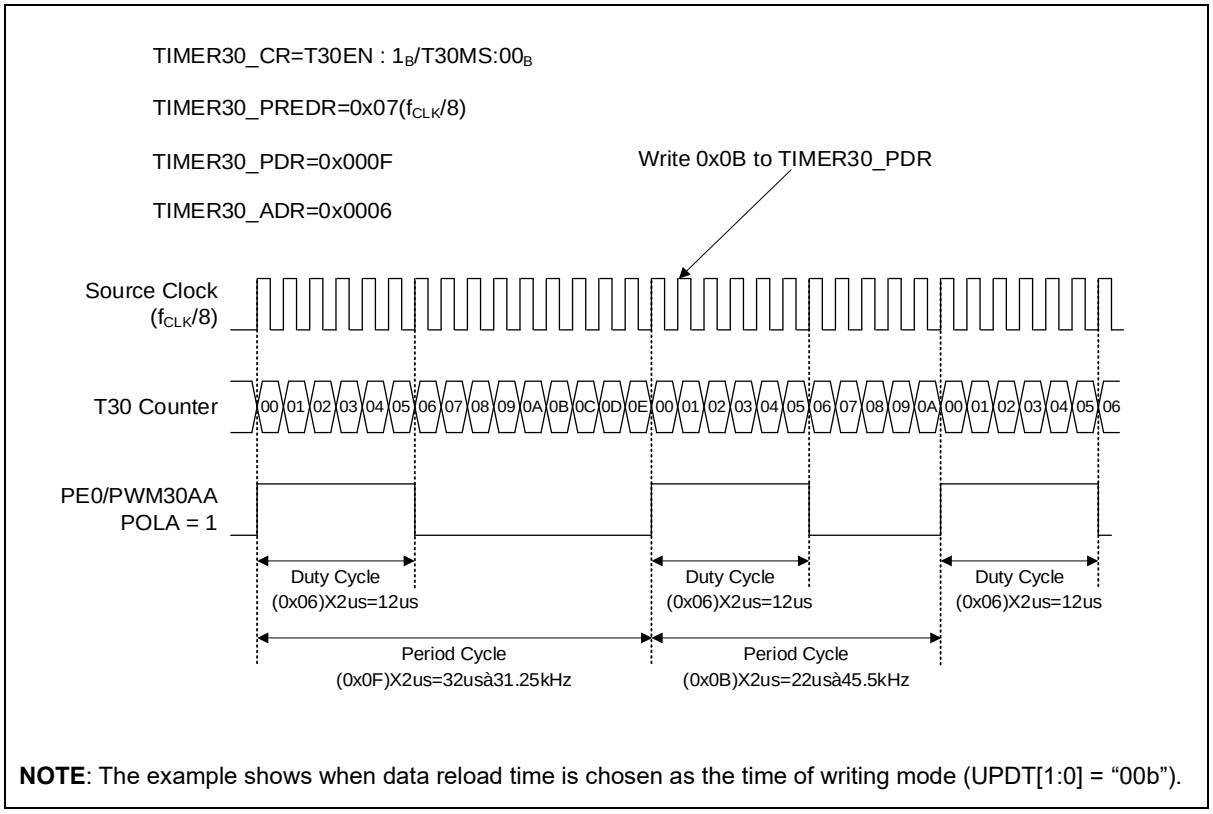


Figure 64. Example of PWM at 4MHZ



NOTE: The example shows when data reload time is chosen as the time of writing mode (UPDT[1:0] = "00b").

Figure 65. Example of Changing the Period in Absolute Duty Cycle at 4MHz

Data reload time selection

The data reload time can choose among "update data to buffer at the time of writing", "update data to buffer at period match", or "update data to buffer at bottom".

PWM output delay

If using the DLYEN bit, DLYPOS bit, and TIMER30_DLY register, it can delay the PWM output. The DLYPOS setting to '0', the delay inserts at front of PWM30xA and at back of PWM30xB pins. The DLYPOS setting to '1', the delay inserts at back of PWM30xA and at front of PWM30xB pins. The settings of DLYEN bit, DLYPOS bit, and TIMER30_DLY register are applied equally to all PWM channels.

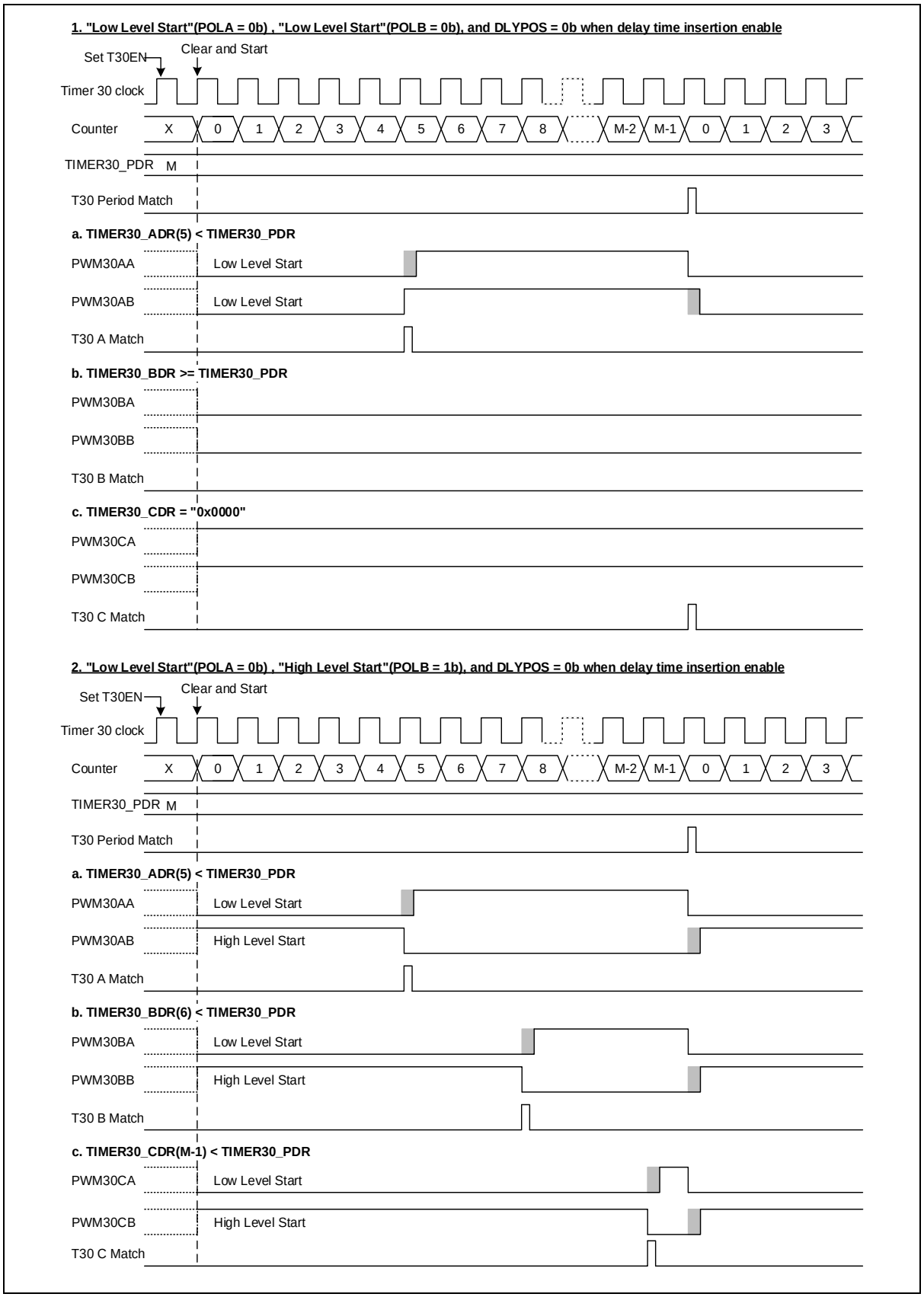


Figure 66. Interval Mode Timing Chart With "DLYPOS = 0"

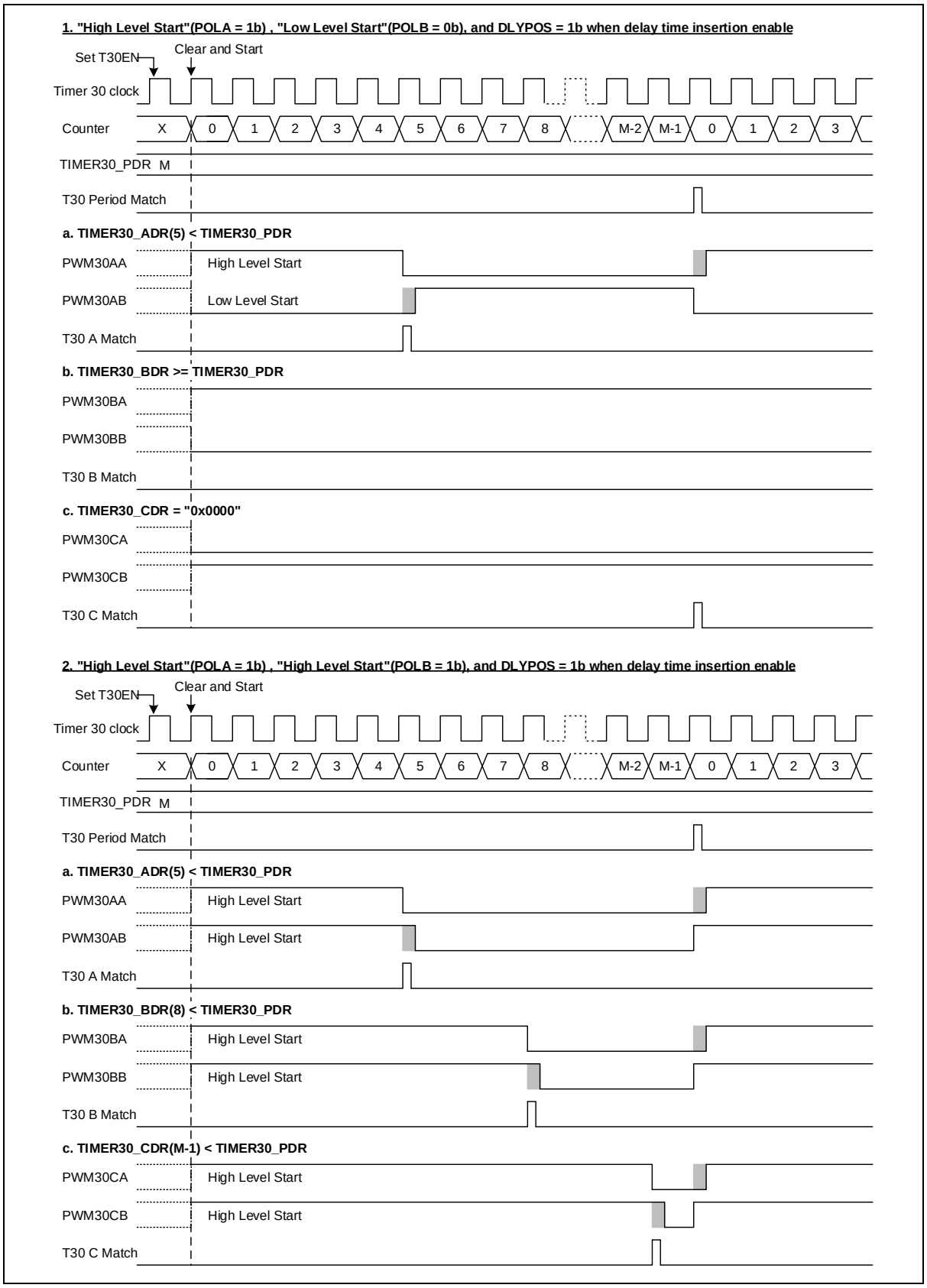


Figure 67. Interval Mode Timing Chart With "DLYPOS = 1

Back-to-back mode

The back-to-back mode is set by T30MS[1:0] as '10'. In the back-to-back mode, the 16-bit up/down counter repeats up/down count. In fact, the effective duty and period becomes twofold of the register set values. If the TIMER30_PDR's data value is set to "0x3210, 16-bit up/down counter will increase until it reaches 0x3210. At this point, a period match signal is generated and the period match interrupt occurs. And then the 16-bit up/down counter will decrease until it reaches 0x0000. At this point, the bottom interrupt occurs. It is repeated in this way.

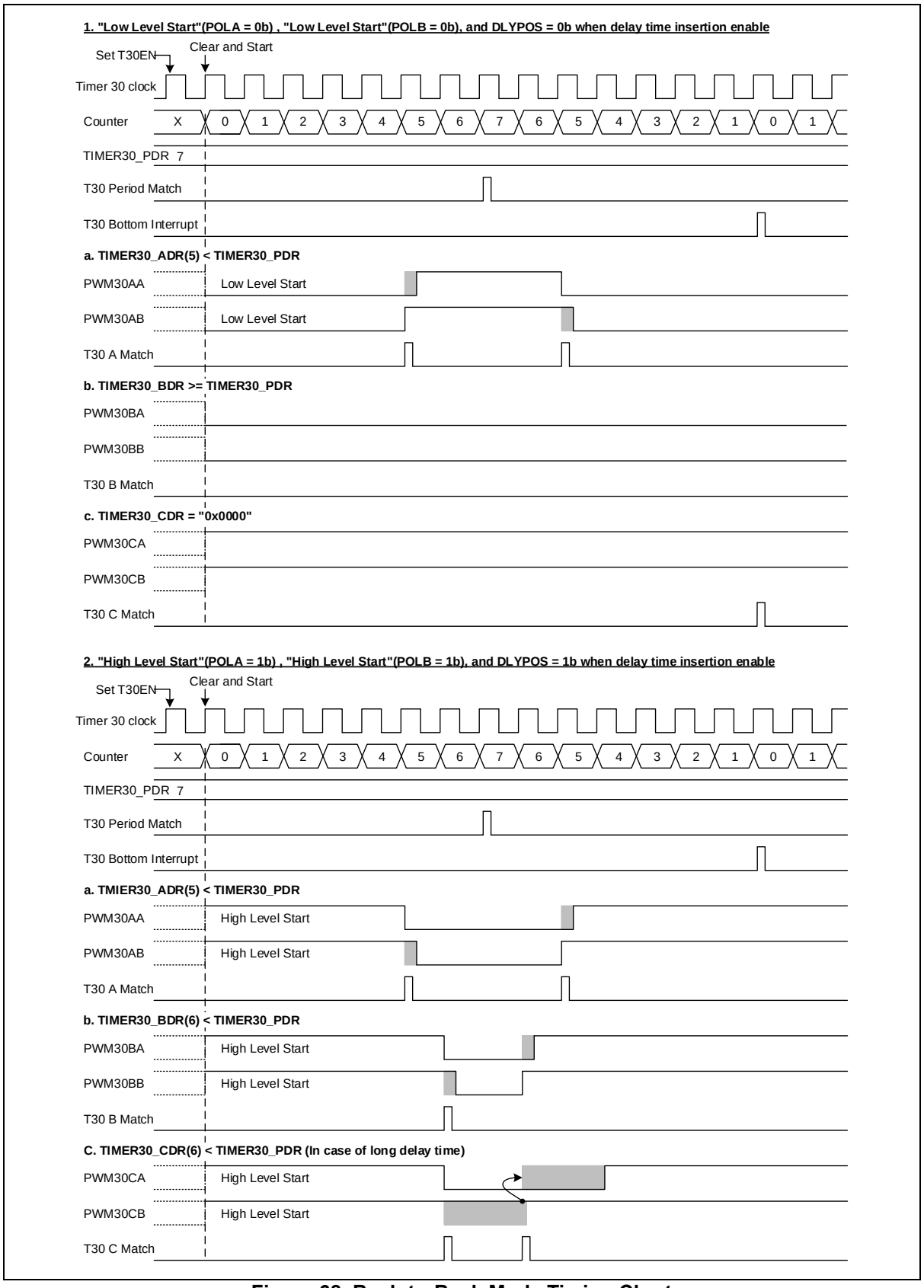


Figure 68. Back-to-Back Mode Timing Chart

Emergency protective function

This protective function is used for emergency stop, when the PWM30xA/PWM30xB output high-impedance enable bit, HIZEN is enabled. When the signal on the external BLNK input pin or internal comparator 3 output goes active (falling or rising edge triggered), the PWM30xA/PWM30xB ports are immediately disabled high-impedance against output and a high-impedance interrupt is occurred. The TIMER30_HIZCR register is used for high-impedance control. The high-impedance source is the external BLNK input pin. The high-impedance edge can be selected by HEDGE bit as falling or rising edge. If the HIZST read value is '1', it indicates that the pins are under a high-impedance state. To return from the high-impedance state, the HIZCLR bit set to '1'. If HIZSW bit is set to '1', PWM30xA/PWM30xB pins go into high impedance by software. It can be used for debugging. (x: A, B and C).

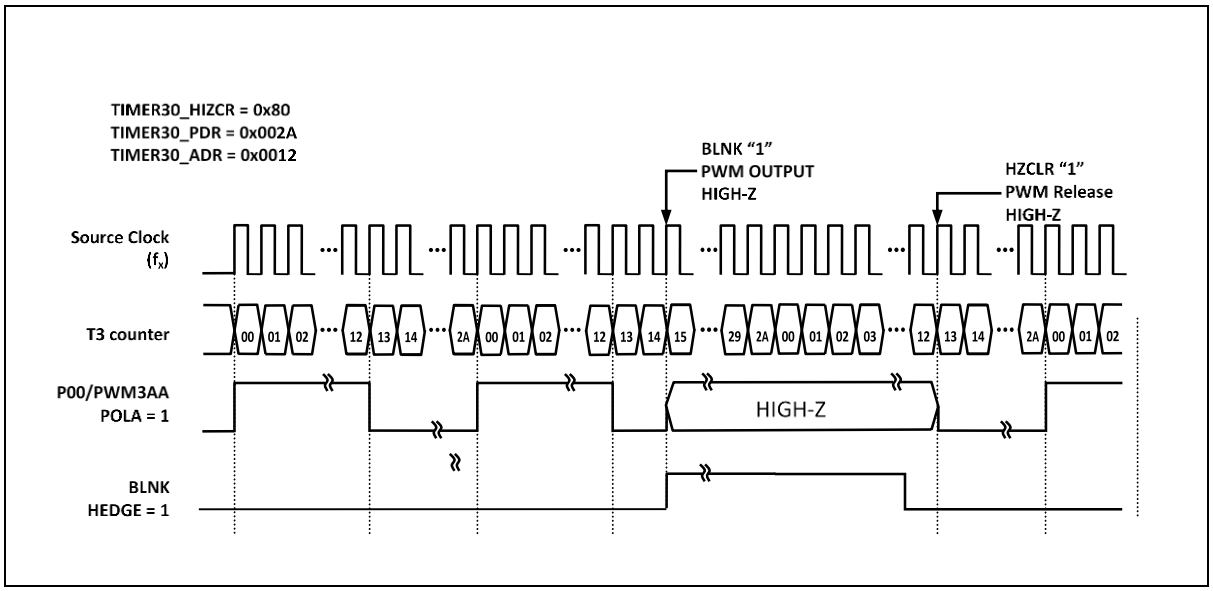


Figure 69. Example of PWM External Synchronization with BLNK Input (x: A, B and C)

Force A-channel mode

If FORCA bit is set to '1', it is possible to enable or disable all PWM output pins through PWM outputs which occur from A-ch duty counter. It is noted that the inversion outputs of A, B, C channel have the same A-ch output waveform.

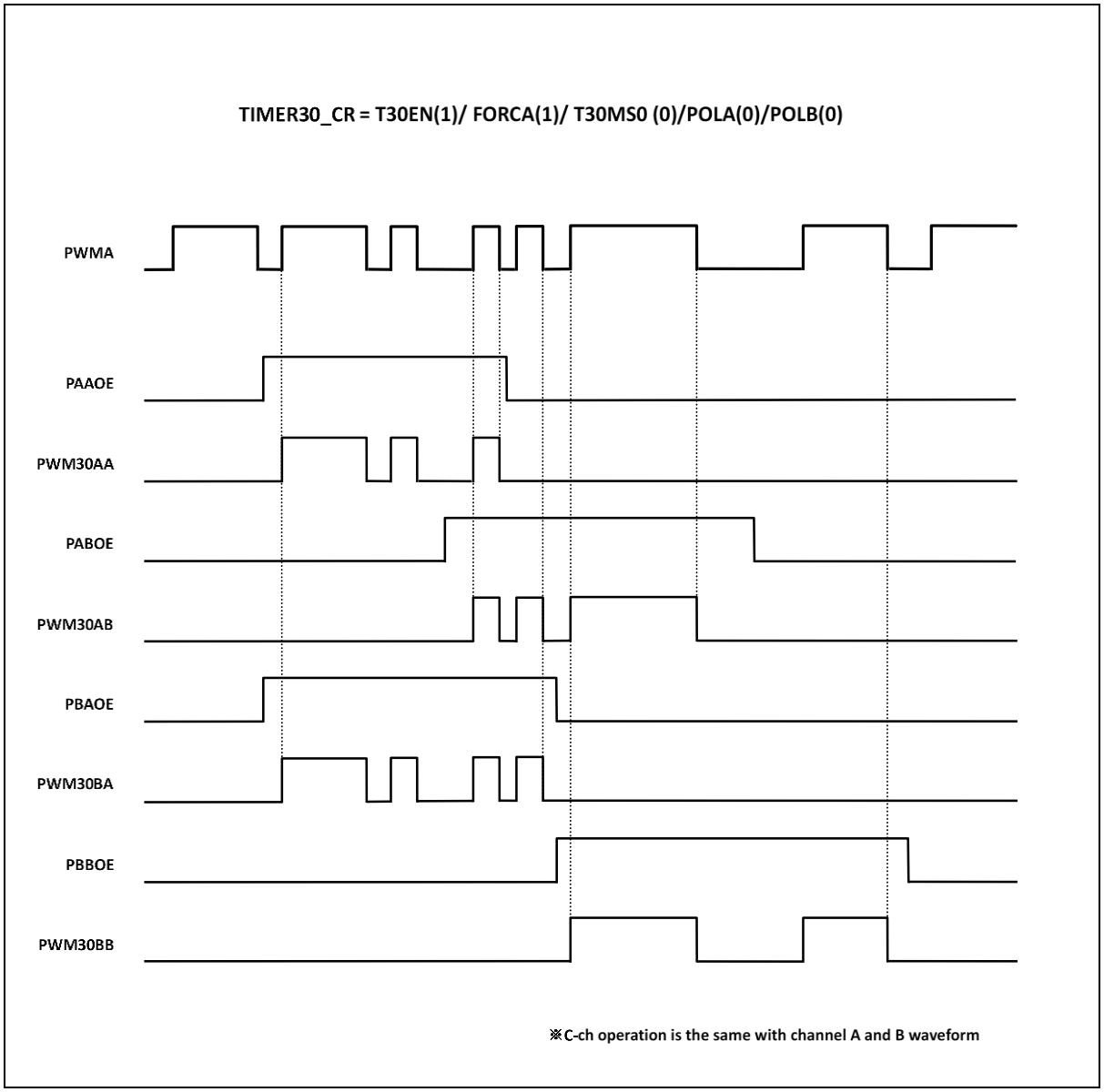


Figure 70. Example of Force A-Channel Mode

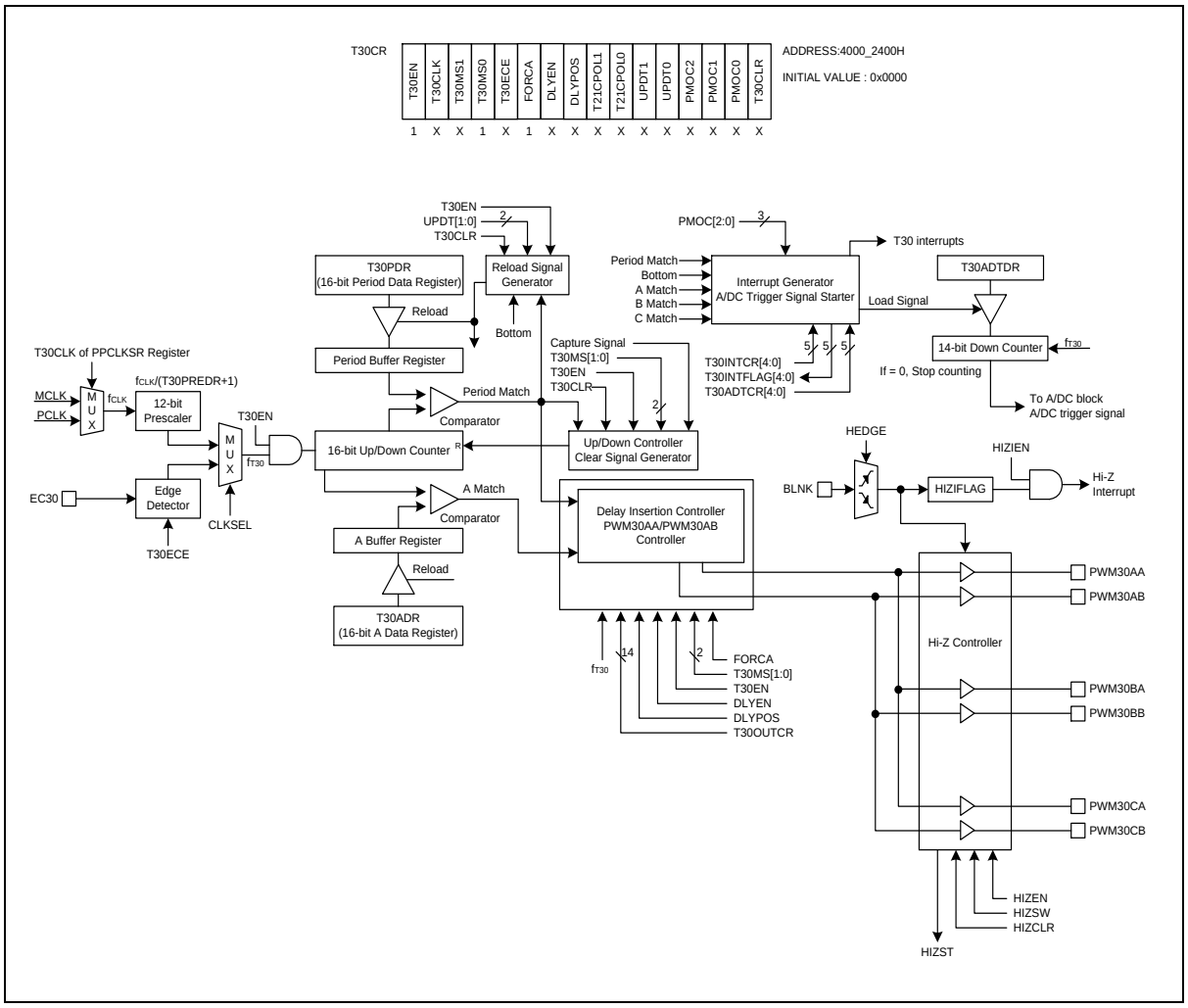


Figure 71. Force A-Channel Mode Block Diagram

6-channel mode

If FORCA bit sets to '0', it is possible to enable or disable PWM output pin and inversion output pin generated through the duty counter of each channel. The inversion output is the reverse phase of the PWM output. A AA/AB output of the A-channel duty register, a BA/BB output of the B-channel duty register, a CA/CB output of the C-channel duty register are controlled respectively.

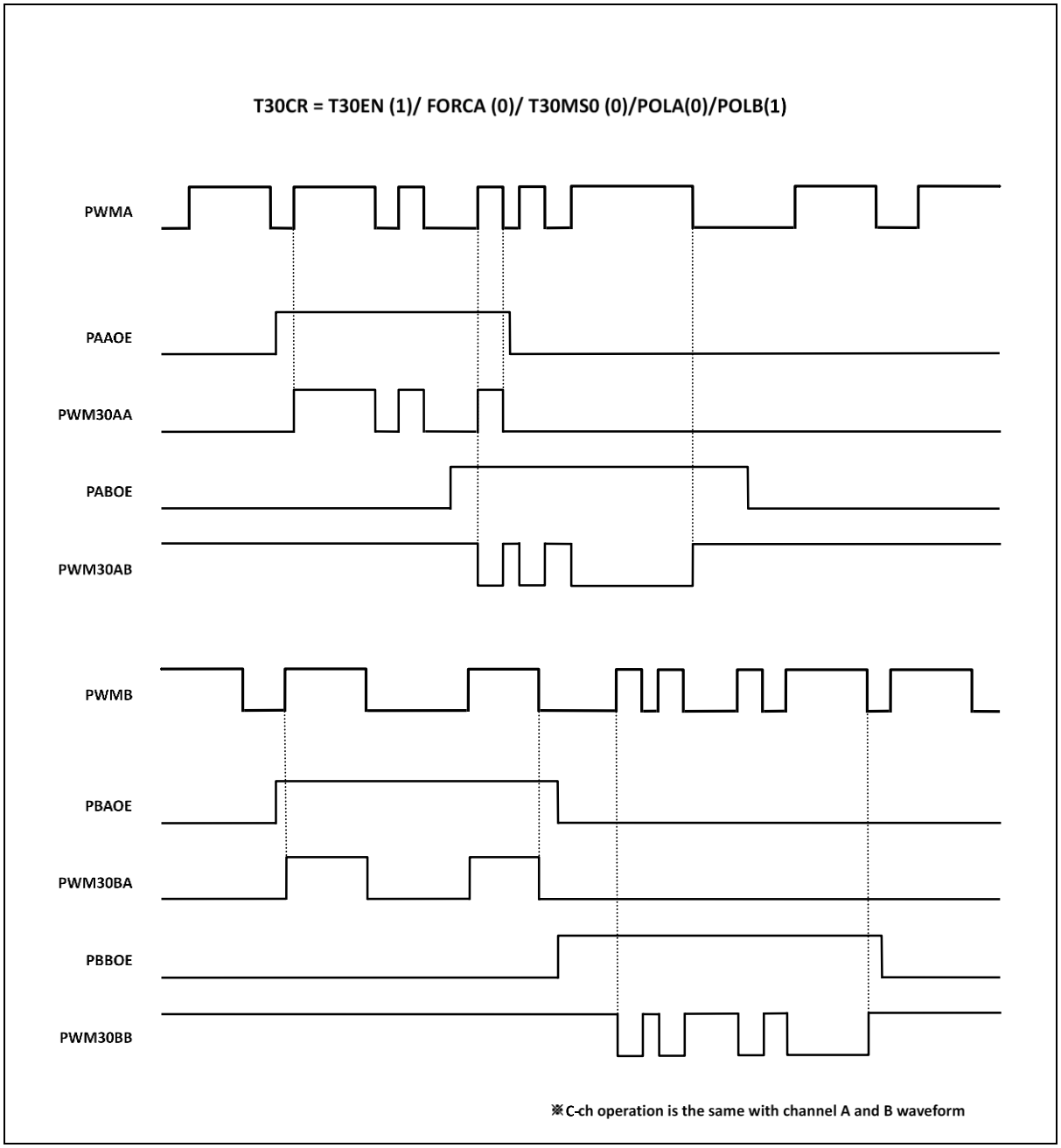


Figure 72. Example of 6-Channel Mode

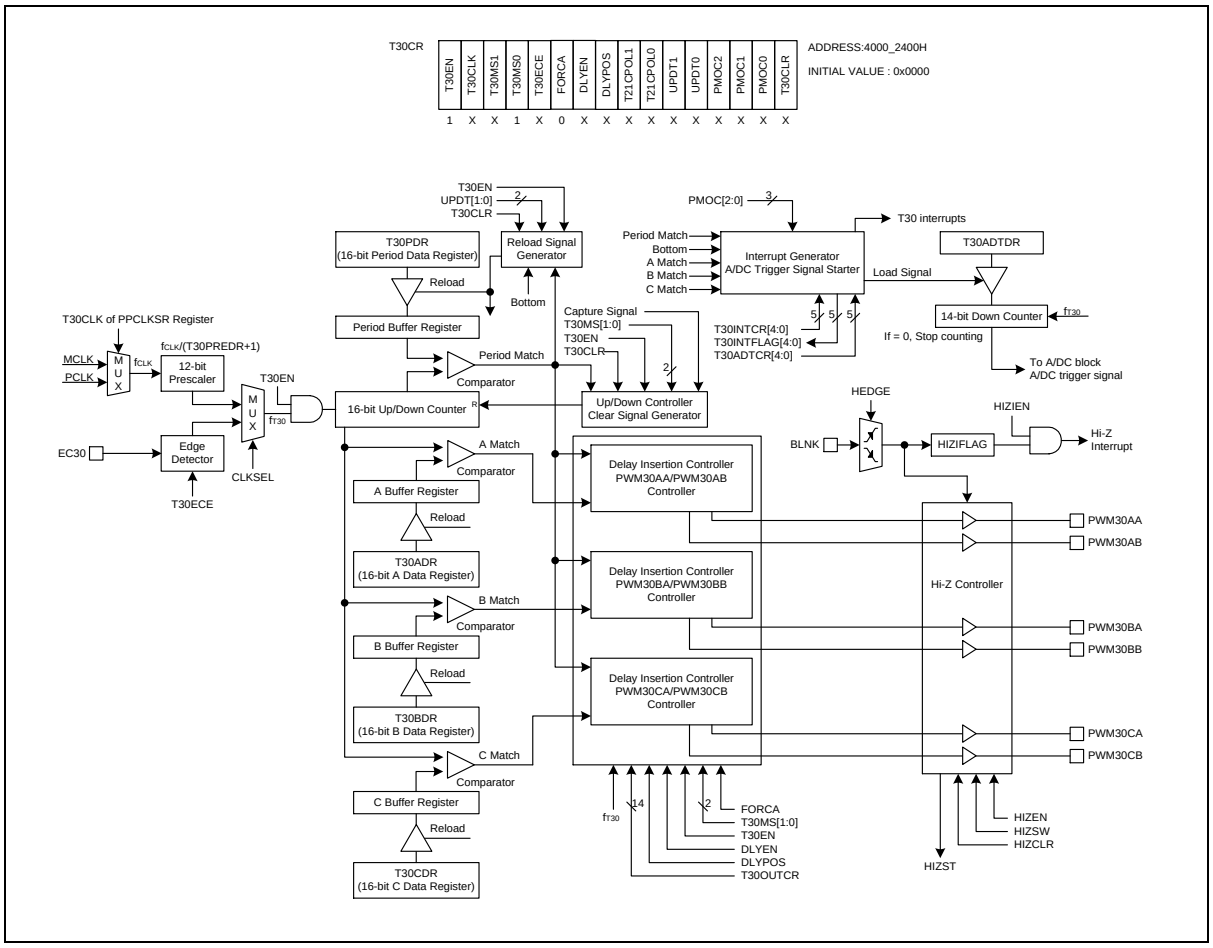


Figure 73. 6-Channel Mode Block Diagram

14 Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are listed below:

- 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 are Supported by Hardware.
- Data OverRun Detection.
- Framing Error Detection.
- Three Separate Interrupts on TX Completion, TX Data Register Empty and RX Completion.
- Double Speed Asynchronous communication mode.

Table 48 introduces pins assigned for the USART.

Table 48. Pin Assignment of USART: External Pins

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)

NOTE: n = 10, 11

14.1 USART block diagram

In this section, USART and SPIN are introduced in block diagrams.

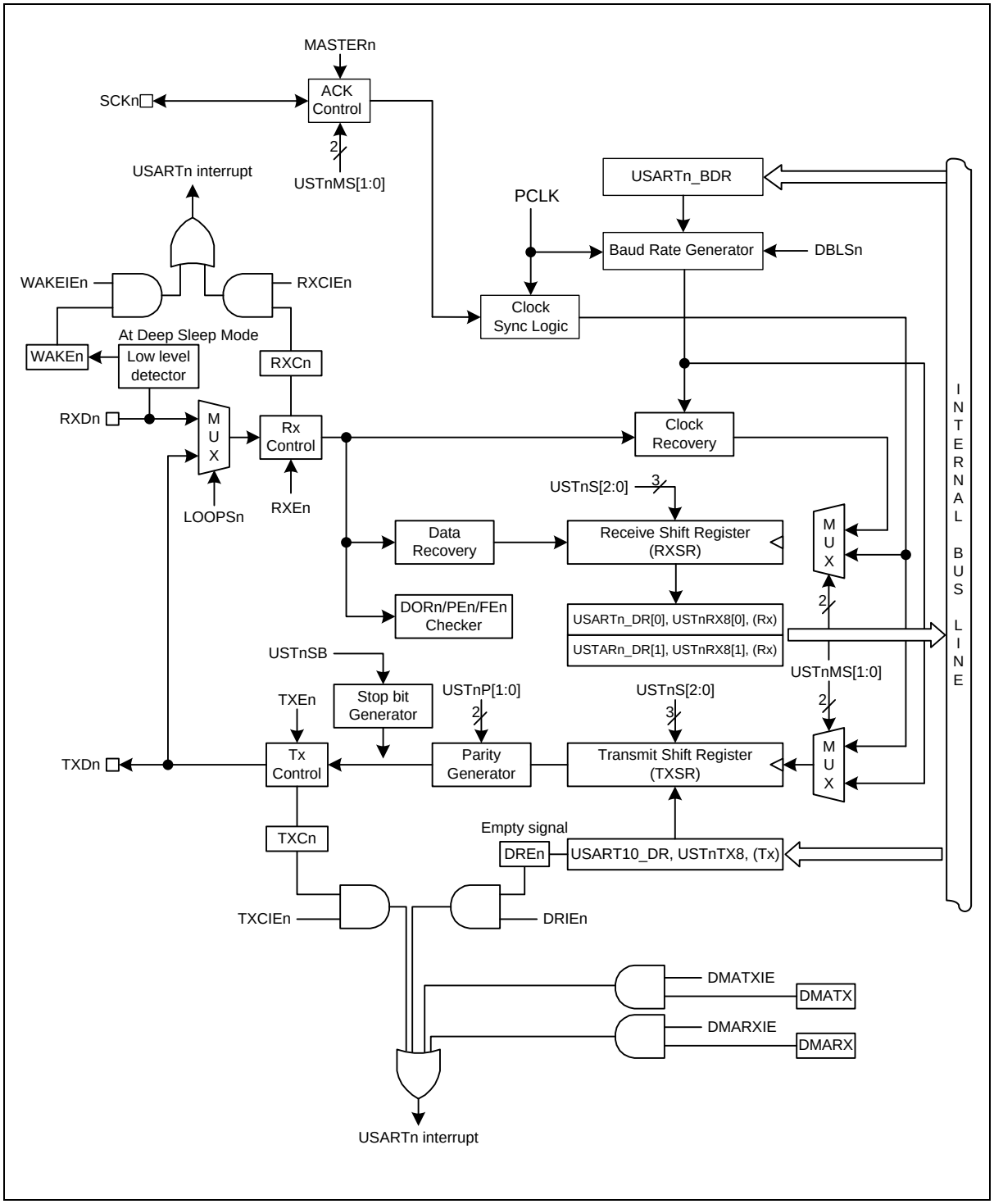


Figure 74. UART Block Diagram (n = 10, 11)

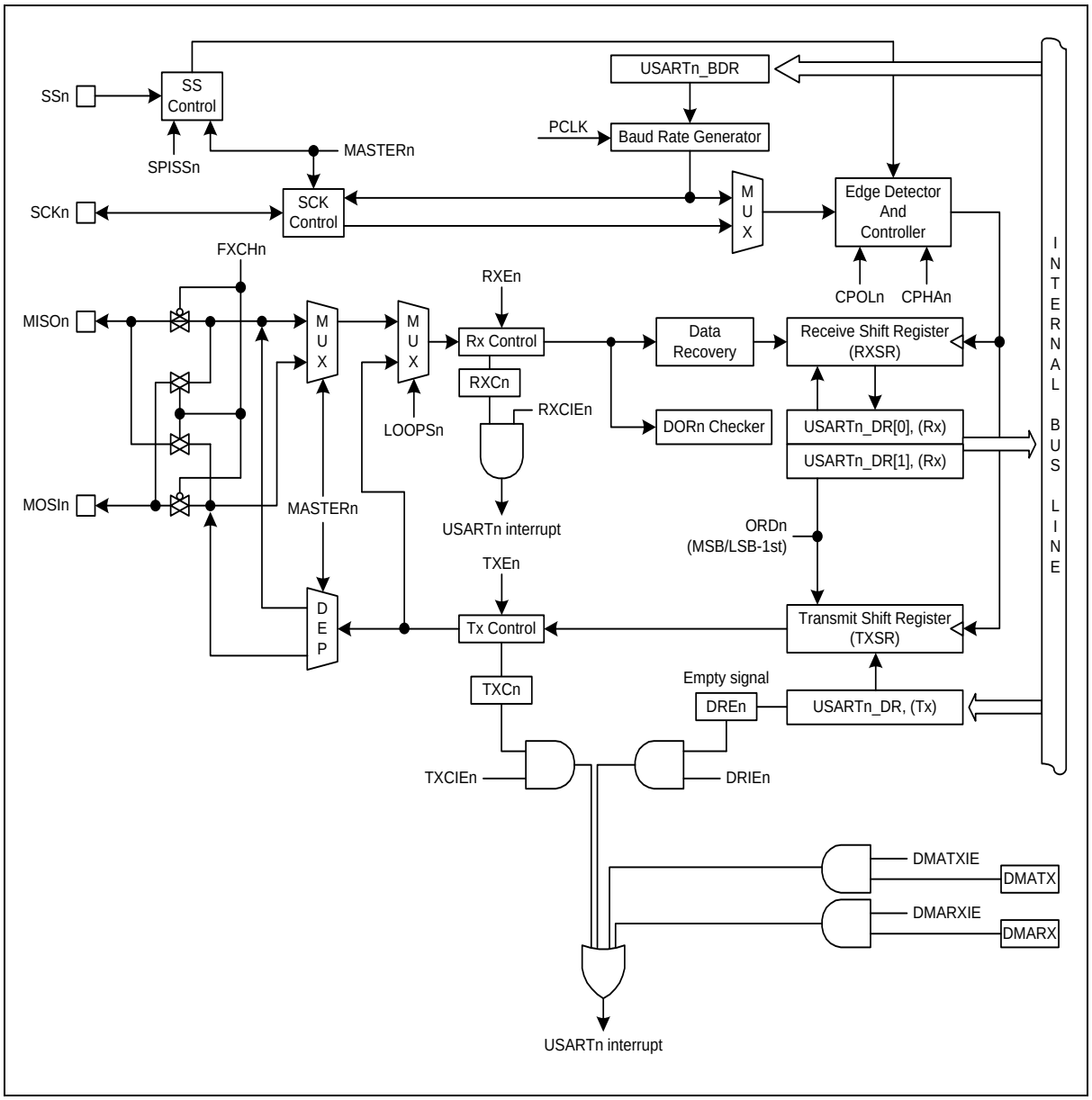


Figure 75. SPIN Block Diagram (n = 10, 11)

14.2 Registers

Base address of USART is introduced in the followings:

Table 49. Base Address of USART

Name	Base address
USART 10	0x4000_3800
USART 11	0x4000_3900

Table 50. USART Register Map

Name	Offset	Type	Description	Reset value	Ref.
USARTn_CR1	0x00	RW	USARTn Control Register 1	0x0000_0000	14.2.1
USARTn_CR2	0x04	RW	USARTn Control Register 2	0x0000_0000	14.2.2
USARTn_ST	0x0C	RW	USARTn Status Register	0x0000_0080	14.2.3
USARTn_BDR	0x10	RW	USARTn Baud Rate Generation Register	0x0000_0FFF	14.2.4
USARTn_DR	0x14	RW	USARTn Data Register	0x0000_0000	14.2.5

NOTE: n = 10, 11

14.2.1 USARTn_CR1: USARTn control register 1

USART module should be configured properly before running. USARTn_CR1 is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11).

USART10_CR1=0x4000_3800, USART11_CR1=0x4000_3900

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	USTnP	USTnS	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	USARTn Operation Mode Selection bits.			
		00	Asynchronous Mode. (UART)		
		01	Synchronous Mode.		
		10	Reserved.		
		11	SPI mode		
13 12	USTnP	Selects Parity Generation and Check method. (only UART mode)			
		00	No parity.		
		01	Reserved.		
		10	Even parity.		
		11	Odd parity.		
11 9	USTnS	Selects the length of data bit in a frame when 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 ACK in synchronous or SPI mode.			
		0	TXD Change @Rising Edge, RXD Change @Falling Edge.		
		1	TXD Change @Falling Edge, RXD Change @Rising Edge.		
6	CPHAn	The CPOLn and this bit determine if data are sampled on the leading or 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 bit.			
		0	Disable the transmit data empty interrupt.		
		1	Enable the transmit data empty interrupt.		
4	TXCIEn	Transmit Complete Interrupt Enable bit.			
		0	Disable transmit complete interrupt.		
		1	Enable transmit complete interrupt.		
3	RXCIEn	Receive Complete Interrupt Enable bit.			
		0	Disable receive complete interrupt.		
		1	Enable receive complete interrupt.		
2	WAKEIEn	Asynchronous Wake-up Interrupt Enable bit in Deep Sleep			

		Mode. When device is in deep sleep mode, if RXDn goes to low level, an interrupt can be requested to wake-up system. (only UART mode)	
		0	Disable asynchronous wake-up interrupt.
		1	Enable asynchronous wake-up interrupt.
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: The CPOLn and CPHAn bits should be changed during the TXEn and RXEn bits are "0b"			

14.2.2 USARTn_CR2: USARTn control register 2

USART module should be configured properly before running. USARTn_CR2 is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11).

USART10_CR2=0x4000_3804, USART11_CR2=0x4000_3904

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	USTnTX8	USTnRX8
-																						0	0	0	0	0	0	0	0	0	0
-																						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

9	USTnEN	Activate USARTn Block by supplying.
	0	Disable USARTn block.
	1	Enable USARTn block.
8	DBLSn	Selects receiver sampling rate. (only UART 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	Control the Loop Back mode of USARTn for test mode.
	0	Normal operation.
	1	Loop Back mode.
5	DISSCKn	In synchronous mode operation, selects the waveform of SCKn output.
	0	SCKn is free-running while USARTn is enabled in synchronous master mode.
	1	SCKn is active while any frame is on transferring.
4	USTnSSEN	This bit controls the SSn pin operation. (only SPI mode)
	0	Disable.
	1	Enable.
3	FXCHn	SPIn port function exchange control bit. (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 bit.
1	USTnTX8	The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Write this bit first before loading the USARTn_DR register.
	0	MSB (9^{th} bit) to be transmitter is '0'.
	1	MSB (9^{th} bit) to be transmitter is '1'.
0	USTnRX8	The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Read this bit first before reading the receive buffer (only UART mode)
	0	MSB (9^{th} bit) to be received is '0'.
	1	MSB (9^{th} bit) to be received is '1'.

14.2.3 USARTn_ST: USARTn status register

USARTn_ST is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11).

USART10_ST =0x4000_380C, USART11_ST =0x4000_390C

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
-			RW	RW	RO	RW	-	RO	RW	RW

7	DREn	Transmit Data Register Empty Interrupt Flag. The DRE flag indicates if the transmit data register (USARTn_DR) is ready to receive new data. If DRE is '1', the data register is empty and ready to be written.
	0	Transmit buffer is not empty.
	1	Transmit buffer is empty. This bit is cleared to '0' when write '1'.
6	TXCn	Transmit Complete Interrupt Flag. This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer.
	0	No request occurred.
	1	Transmit buffer is empty and the data in transmit shift register are shifted out completely. This bit is cleared to '0' when write '1'.
5	RXCn	Receive Complete Interrupt Flag. This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read.
	0	There is no data unread in the receive buffer.
	1	There are more than 1 data in the receive buffer.
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	This bit is set if data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.
	0	No Data OverRun.
	1	Data OverRun detected.
1	Fen	This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read. (only UART mode)
	0	No Frame Error.
	1	Frame Error detected.
0	Pen	This bit is set if the next character in the receive buffer has a Parity Error while parity is checked. This bit is valid until the receive buffer is read. (only UART mode)
	0	No Parity Error.
	1	Parity Error detected.

14.2.4 USARTn_BDR: USARTn baud rate generation register

USARTn_BDR is a 32-bit register, and able to do 32/16/8-bit access (n = 10, 11).

USART10_BDR =0x4000_3810, USART11_BDR =0x4000_3910

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
-			0xFFF
-			RW

11 0	BDATA	The value in this register is used to generate internal baud rate in UART mode or to generate SCK clock in SPI mode. To prevent malfunction, do not write '0' in UART mode and do not write '0' or '1' in synchronous or SPI mode.

14.2.5 USARTn_DR: USARTn data register

USART10_DR =0x4000_3814, USART11_DR =0x4000_3914

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
-			0x00
-			RW

7 0	DATA	The USART 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 USARTn_DR register. Reading the USARTn_DR register returns the contents of the Receive Buffer. Write to this register only when the DRE flag is set. In SPI master mode, the SCK clock is generated when data are written to this register. **NOTE**: This byte won't be written when the block is disabled or the both of TXEn and RXEn bits are "0b".

14.3 Functional description

USART comprises clock generator, transmitter and receiver.

The clock generation logic consists of synchronization logic for external clock input used by synchronizing or SPI slave operation, and the baud rate generator for asynchronous or master (synchronous or SPI) operation.

The transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows continuous transfer of data without any delay between frames.

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

14.3.1 USART clock generation

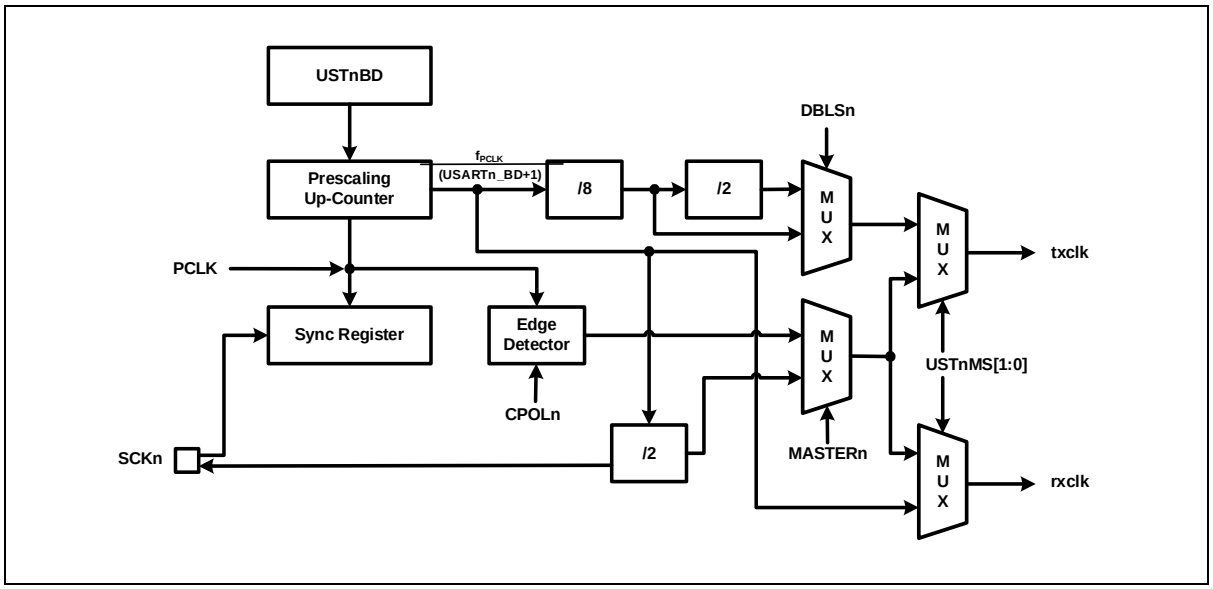


Figure 76. Clock Generation Block Diagram (USARTn, n = 10, 11)

The clock generation logic generates the base clock for the transmitter and receiver. The USART supports four modes of clock operation and those are normal asynchronous, double speed asynchronous, master synchronous and slave synchronous modes.

The clock generation scheme for master SPI and slave SPI mode is the same as master synchronous and slave synchronous operation mode.

USTnMS[1:0] bits in 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 51 shows the equations for calculating baud rate (in bps).

Table 51. Equations for Calculating USART Baud Rate Register Settings

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))

NOTE: n = 10, 11, 12, and 13

14.3.2 External clock (SCKn)

External clocking is used in the synchronous or SPI slave mode of operation.

External clock input from the SCKn pin is sampled by a synchronization logic to remove meta-stability. Output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and receiver. This process introduces two CPU clock period delay. The maximum frequency of the external SCKn pin is limited by 1MHz.

14.3.3 Synchronous mode operation

External clocking is used in the synchronous or SPI slave mode of operation.

When synchronous or SPI mode is used, the SCKn pin will be used as either clock input (slave) or clock output (master). Data sampling and transmitter are issued on the different edge 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.

CPOLn bit in USARTn_CR1 register selects which SCKn clock edge is used for data sampling and which is used for data change. As shown in Figure 77, when CPOLn is zero, the data will be changed at rising edge of SCKn and sampled at falling edge of SCKn.

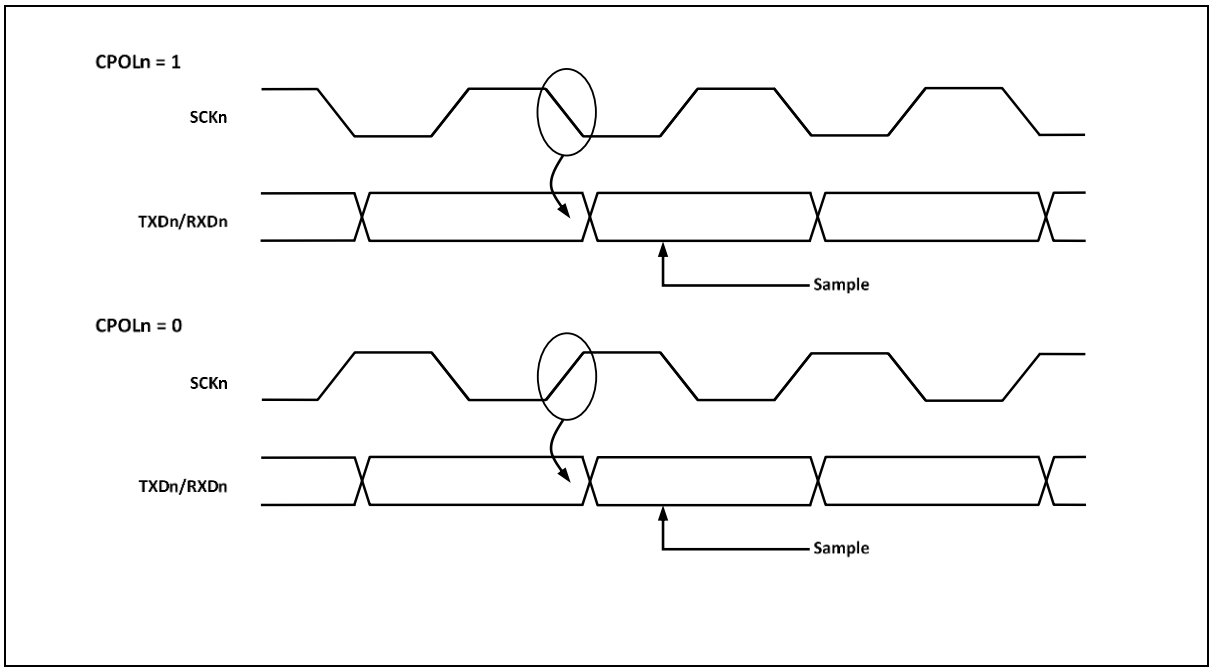


Figure 77. Synchronous Mode SCKn Timing (USARTn, n = 10, 11)

14.3.4 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.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- No, even or odd parity bit.
- 1 or 2 stop bits.

A frame starts with a start bit followed by the least significant data bit (LSB). Next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If 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 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 means high state of data pin.

Figure 78 shows possible combinations of the frame formats. Bits inside brackets are optional.

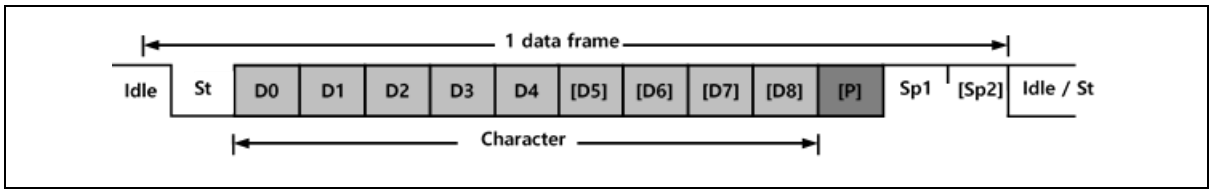


Figure 78. Frame Format (USART)

Single data frame consists of the following bits:

- Idle: No communication on communication line (TXDn/RXDn)
- St: Start bit (Low)
- Dm: Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

Frame format of UART is set by configuring USTnS[2:0], USTnP[1:0] bits in USARTn_CR1 register and USTnSB bit in USARTn_CR2 register. Transmitter and receiver use the same figures (n = 10, 11).

14.3.5 UART parity bit

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. Parity bit is located between the MSB and 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

14.3.6 UART transmitter

UART transmitter is enabled by configuring TXEn bit in USARTn_CR1 register. When the transmitter is enabled, TXDn pin should be set to TXDn function for the serial output pin of UART by the PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD. Baud rate, operation mode and frame format must be set up once before starting any transmission.

In synchronous operation mode, SCKn pin is used as a transmission clock, so it should be selected to function SCKn by PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD (n = 10, 11).

UART sending TX data

A data transmission is initiated by loading the transmit buffer (USARTn_DR register I/O location) with the data to be transmitted. The data be written in transmit buffer 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 with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode, the ninth bit must be written to the USTnTX8 bit in USARTn_CR2 register before it is loaded to the transmit buffer USARTn_DR register (n = 10, 11, 12).

UART transmitter flag and interrupt

The USART transmitter has two flags which indicate its state. One is USART data register empty flag (DREn) and the other is transmit completion flag (TXCn). Both flags can be 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. And also this flag can be cleared by writing '0' to this bit position. Writing '1' to this bit position is prevented.

When the data register empty interrupt enable (DRIEn) bit in USARTn_CR1 register is set and the global interrupt is enabled, USTnST status register empty interrupt is generated while DREn flag is set.

The transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer. The TXCn flag is automatically cleared when the transmit complete interrupt serve routine is executed, or it can be cleared by writing '0' to TXCn bit in USARTn_ST register.

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

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.

UART disabling transmitter

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

14.3.7 UART receiver

USART 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 of UART by PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD. The baud-rate, mode of operation and frame format must be set before serial reception. In synchronous or SPI operation mode the SCKn pin is used as transfer clock input, so it should be selected to do SCKn function by PB_AFSR1/PD_AFSR1 and PB_MOD/PD_MOD. In SPI operation mode the SSn input pin in slave mode can be configured as SSn output pin in master mode. This can be done by setting USTnSSEN bit in USARTn_CR2 register (n = 10, 11).

UART receiving RX data

When UART is in synchronous or asynchronous operation mode, the receiver starts data reception when it detects a valid start bit (LOW) on RXDn pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or 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's the 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 buffer. The receive buffer is read by reading the USARTn_DR register.

If 9-bit characters are used (USTnS[2:0] = "111"), the ninth bit is stored in the USTnRX8 bit position in the USARTn_CR2 register. The ninth bit must be read from the USTnRX8 bit before reading the low 8 bits from the USARTn_TDR register. Likewise, the error flags Fen, DORn, Pen must be read before reading the data from USARTn_DR register. It's because the error flags are stored in the same FIFO position of the receive buffer (n = 10, 11).

UART receiver flag and interrupt

The UART receiver has one flag that indicates the receiver state.

A receive complete (RXCn) flag indicates whether there are 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 a 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. As received data is stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from USARTn_DR register, read the USARTn_ST register first which contains error flags.

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", and 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 receive buffer full condition. DORn occurs when the receive buffer is full, and another new data is presented in the receive shift register which are to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To avoid data loss or clear this flag, read the receive buffer.

The parity error (Pen) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USTnP1=0), the Pen bit is always read "0" (n = 10, 11, 12 and 13).

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, 12).

UART disabling receiver

In contrast to transmitter, disabling the Receiver by clearing RXEn bit makes the Receiver inactive immediately.

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 (n = 10, 11).

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 does sampling and low pass filtering the incoming bits, and removing the noise of RXDn pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times of 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 shown when using the double speed mode (n = 10, 11).

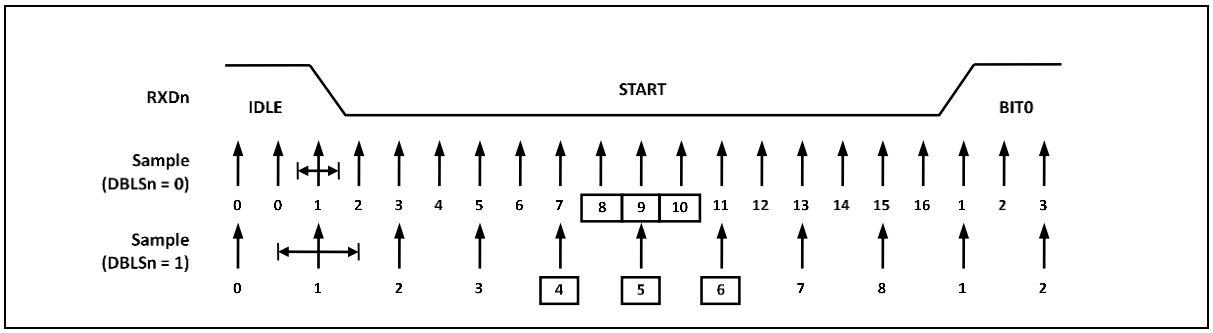


Figure 79. Asynchronous Start Bit Sampling (n = 10, 11)

When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, the start bit condition. After detecting high to low transition on RXDn line, the clock recovery logic uses samples 8, 9 and 10 for normal mode to decide if a valid start bit is received. If more than 2 samples have 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. And 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 same to the clock recovery process. The data recovery logic samples 16 times for each incoming bits 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 to a logic '0' and if more than 2 samples have high levels, the received bit is considered to 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 waiting to find start bit.

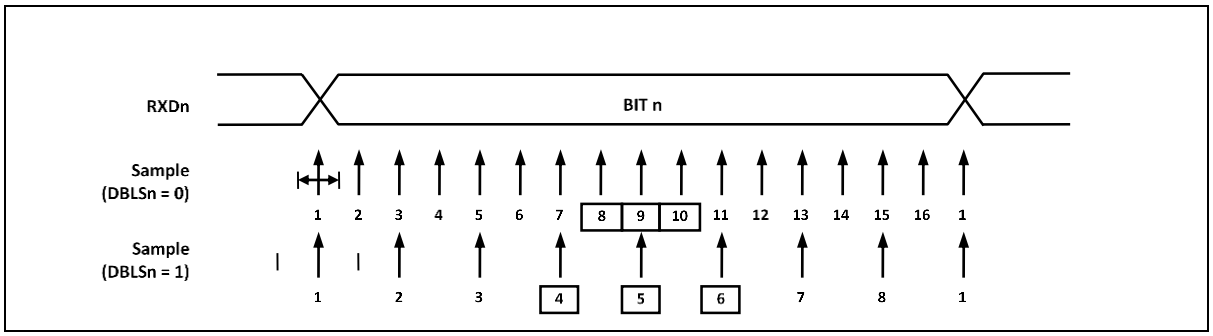


Figure 80. Asynchronous Data and Parity Bit Sampling (n = 10, 11)

The process for detecting stop bit is same as clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (Fen) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXDn line to check a valid high to low transition is detected (start bit detection). (n = 10, 11, 12 and 13).

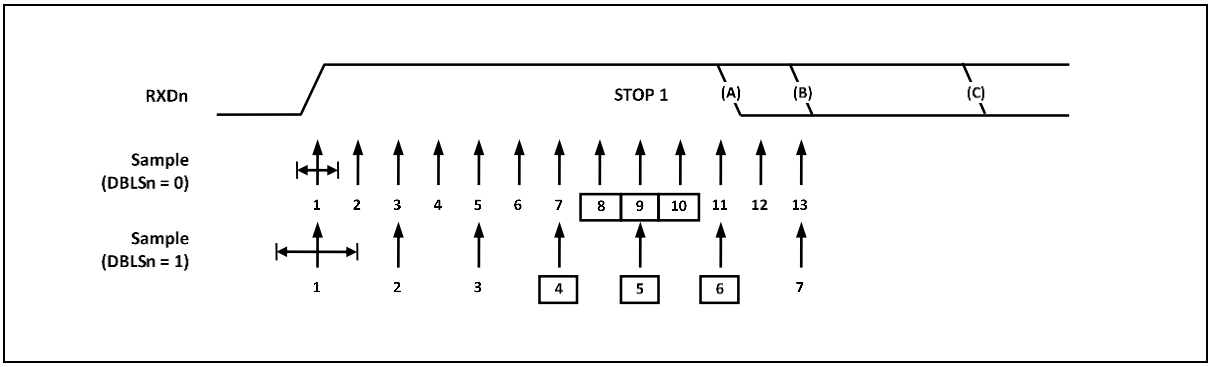


Figure 81. Stop Bit Sampling and Next Start Bit Sampling (n = 10, 11)

14.3.8 SPI mode

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

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

When the SPI mode is enabled by configuring USTnMS[1:0] as "11", the slave select (SSn) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USTnSSEN bit is set to '0'.

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, 12).

14.3.9 SPI clock formats and timing

To accommodate a wide variety if 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. CPOLn selectively insert an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data. Note that CPHAn and CPOLn bits in USARTn_CR1 register have different meanings according to the USTnMS[1:0] bits which decides the operating mode of USART.

Table 52 shows four combinations of CPOLn and CPHAn for SPI modes 10, 11

Table 52. CPOL Functionality

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)

NOTE: n = 10, 11, 12 and 13

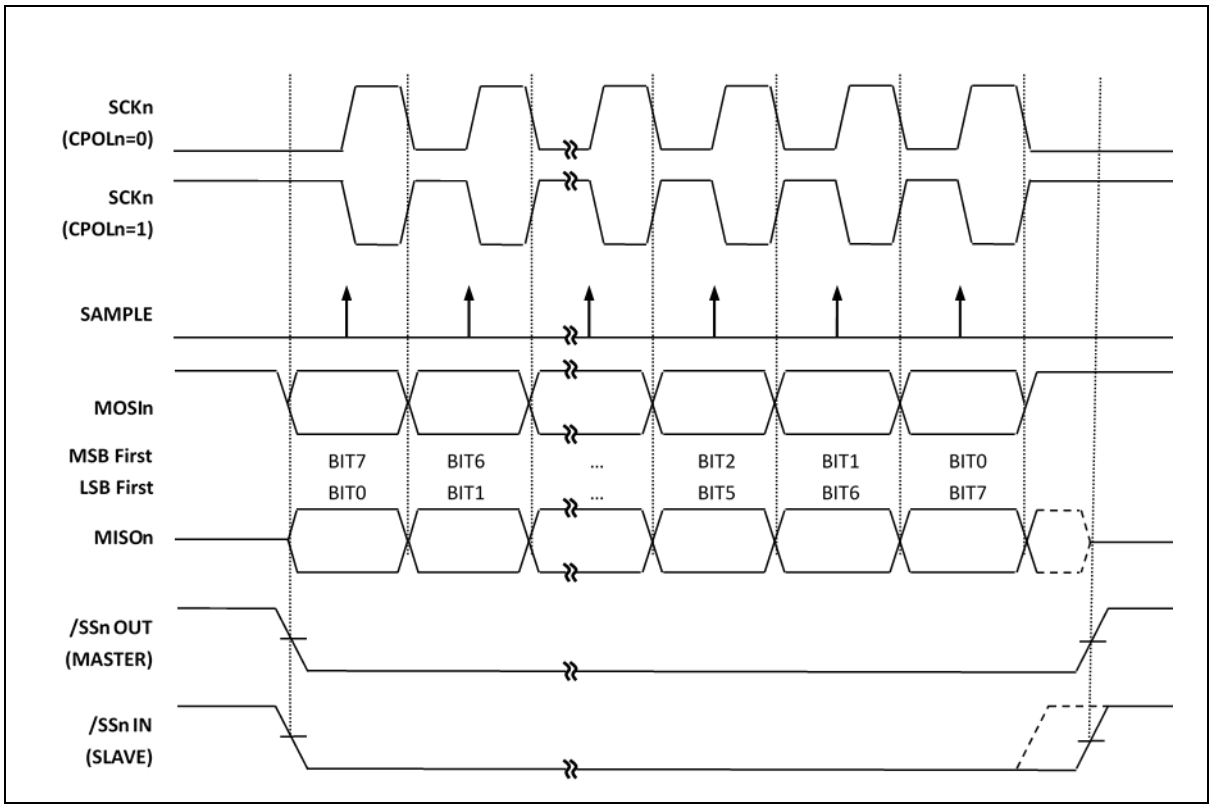


Figure 82. USART SPIn Clock Formats when CPHAn = 0 (n = 10, 11)

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. Unlike the case of CPHAn=1, when CPHAn=0, the slave's SSn input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SSn input (n = 10, 11).

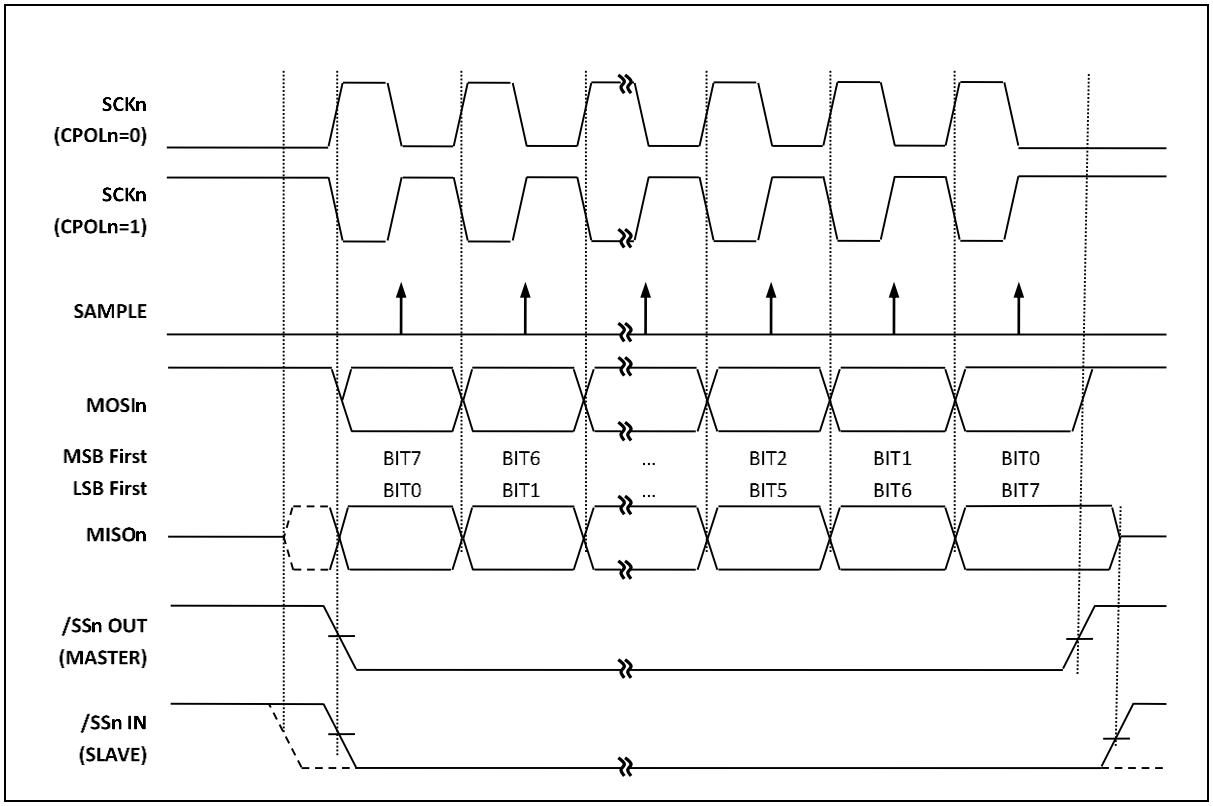


Figure 83. USART SPIn Clock Formats when CPHAn=1 (n = 10, 11)

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. When CPHAn=1, the slave's SSn input is not required to go to its inactive high level between transfers.

Because the SPIn logic reuses the USART resources, SPIn mode of operation is similar to that of synchronous or asynchronous operation. SPIn transfer is initiated by checking for the USART Data Register Empty flag (DREn=1) and then writing a byte of data to the USTnDR Register. In master mode of operation, even if transmission is not enabled (TXEn=0), writing data to the USTnDR register is necessary because the clock SCKn is generated from transmitter block.

15 Universal Synchronous Receiver/Transmitter (UART)

2-channel UART (Universal Asynchronous Receiver/Transmitter) modules are provided. UART operation status including error status can be read from status register. The prescaler which generates proper baud rate, is exist for each UART channel. The prescaler can divide the UART clock source which is PCLK, from 1 to 65535. And baud rate generation is by clock which internally divided by 16 of the prescaled clock and 8-bit precision clock tuning function.

Programmable interrupt generation function will help to control the communication via UART channel

- Compatible with 16450
- Standard asynchronous control bit (start, stop, and parity) configurable
- 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
- Detail status register

Table 53 introduces pins assigned for the USART.

Table 53. Pin Assignment of USART: External Pins

Pin name	Type	Description
TXD0	O	UART Channel 0 transmit output
RXD0	I	UART Channel 0 receive input
TXD1	O	UART Channel 1 transmit output
RXD1	I	UART Channel 1 receive input

15.1 USART block diagram

In this section, USART and SPIN are introduced in block diagrams.

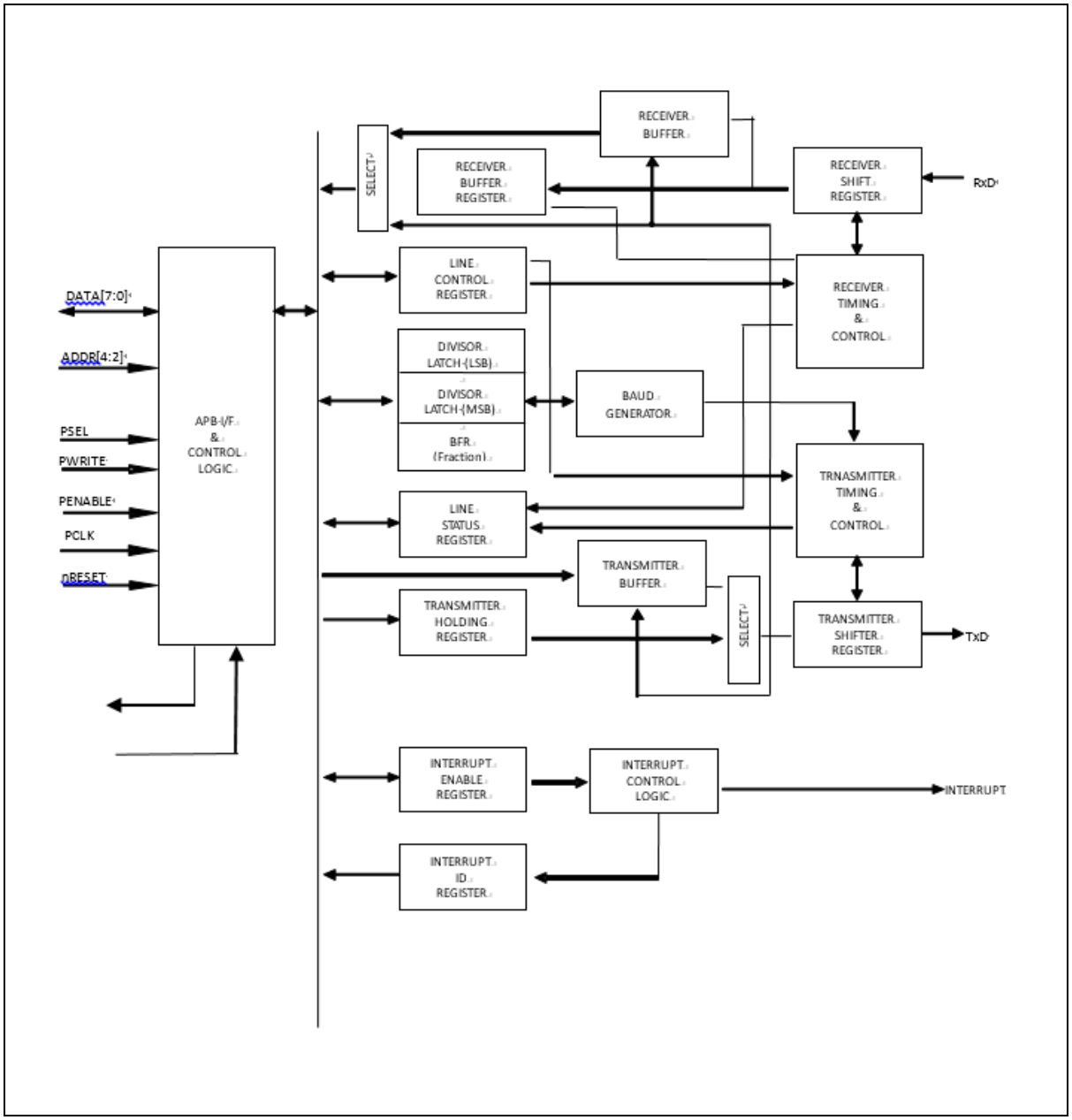


Figure 84. UART Block Diagram

15.2 Registers

Base address of USART is introduced in the followings:

Table 54. Base Address of USART

Name	Base address
UART0	0x4000_4000
UART1	0x4000_4100

Table 55. USART Register Map

Name	Offset	Type	Description	Reset value	Ref.
UARTn_RBR	0x00	RO	Receive data buffer register	0x0000_0000	15.2.1
UARTn_THR	0x00	WO	Transmit data hold register	0x0000_0000	15.2.2
UARTn_IER	0x04	RW	Interrupt enable register	0x0000_0000	15.2.3
UARTn_IIR	0x08	RO	Interrupt ID register	0x0000_0001	15.2.4
UARTn_LCR	0x0C	RW	Line control register	0x0000_0000	15.2.5
UARTn_DCR	0x10	RW	Data Control Register	0x0000_0000	15.2.6
UARTn_LSR	0x14	RO	Line status register	0x0000_0060	15.2.7
UARTn_BDR	0x20	RW	Baud rate Divisor Latch Register	0x0000_0000	15.2.8
UARTn_BFR	0x24	RW	Baud rate Fractional Counter Value	0x0000_0000	15.2.9
UARTn_IDTR	0x30	RW	Inter-frame Delay Time Register	0x0000_0080	15.2.10

NOTE: n = 0, 1

15.2.1 UARTn_RBR: UARTn receive data buffer register

UARTn Receive Data Buffer Register is 32-bit register. Received data will be read out from this register. Maximum length of data is 8 bits. Last data received will be maintained in this register until a new byte is received. This Register is able to 32/16/8-bit access. (UART, n = 0, 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
-			0x00
-			RO

7 0	RBR	UARTn Receive Data Buffer bits

15.2.2 UARTn_THR: UARTn transmit data hold register

UARTn Receive Data Buffer Register is 32-bit register. This Register is able to 32/16/8-bit access. (UART, n = 0, 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
-			0x00
-			WO

7 0	THR	UARTn Transmit Data Hold bits

15.2.3 UARTn_IER: UARTn interrupt enable register

UARTn Interrupt Enable Register is 32-bit register. This Register is able to 32/16/8-bit access.

(UART, n = 0, 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				DTXIEN	DRXIEN	TXEIE	RLSIE	THREIE	DRIE
-				0	0	0	0	0	0
-				RW	RW	RW	RW	RW	RW

5	DTXIEN	DMA transmit done interrupt enable
	0	Receive line status interrupt is disabled
	1	Receive line status interrupt is enabled
4	DRXIEN	DMA receive done interrupt enable
	0	DMA receive done interrupt is disabled
	1	DMA receive done interrupt is enabled
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 bit.
	0	Disable receiver line status interrupt.
	1	Enable receiver line status interrupt.
1	THREIE	Transmit Holding Register Empty Interrupt Enable bit.
	0	Disable transmit holding register empty interrupt.
	1	Enable transmit holding register empty interrupt.
0	DRIE	Data Receive Interrupt Enable bit.
	0	Disable data receive interrupt.
	1	Enable data receive interrupt.

15.2.4 UARTn_IIR: UARTn interrupt ID register

UARTn Interrupt ID Register is 32-bit register. This Register is able to 32/16/8-bit access.

(UART, n = 0, 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	IPEN
-				0	-	00	1
-				RO	-	RO	RO

4	TXE	Transmit Complete Interrupt Source ID bit.
2 1	IID	UARTn Interrupt ID bits. **NOTE**: 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 bit. 0 Interrupt is pending. 1 No interrupt is pending.

Table 56. Interrupt ID and Control

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: After check the above bits, Read data buffer to avoid losing interrupt source.

15.2.5 UARTn_LCR: UARTn line control register

UARTn Line Control Register is 32-bit register. This Register is able to 32/16/8-bit access.

(UART, n = 0, 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
-				0	0	0	0	0	00
-				RW	RW	RW	RW	RW	RW

6	BREAK	Transfer Break Control bit. The TXDn pin will be driven at low state in order to notice the alert to the receiver.
	0	Normal transfer mode.
	1	Break transmit mode.
5	STICKP	Force Parity bit. This bit is effective when the PEN bit is set to "1b".
	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 bit.
	0	Odd parity mode.
	1	Even parity mode.
3	PEN	Parity Bit Transfer Enable bit.
	0	Disable parity transfer.
	1	Enable parity transfer.
2	STOPBIT	Stop Bit Length Selection bit.
	0	1 stop bit.
	1	1.5 or 2 stop bit. 1.5 stop bit in case of 5-bit data length and 2 stop bit in case of 6/7/8-bit data length.
1 0	DLEN	Data Length Selection bits.
	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, or 5 of UARTn_LCR register. The table shows the variation of parity bit generation.

Table 57. Various Configurations to Create a Parity Bit

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"

15.2.6 UARTn_DCR: UARTn data control register

UARTn Data Control Register is 32-bit register. This Register is able to 32/16/8-bit access.

(UART, n = 0, 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 bit.	
		0	Normal mode.
		1	Local loopback mode. TXDn connected to RXDn internally.
3	RXINV	Receive Data Inversion Selection bit.	
		0	Normal receive data input.
		1	Inverted receive data input.
2	TXINV	Transmit Data Inversion Selection bit.	
		0	Normal transmit output.
		1	Inverted transmit output.

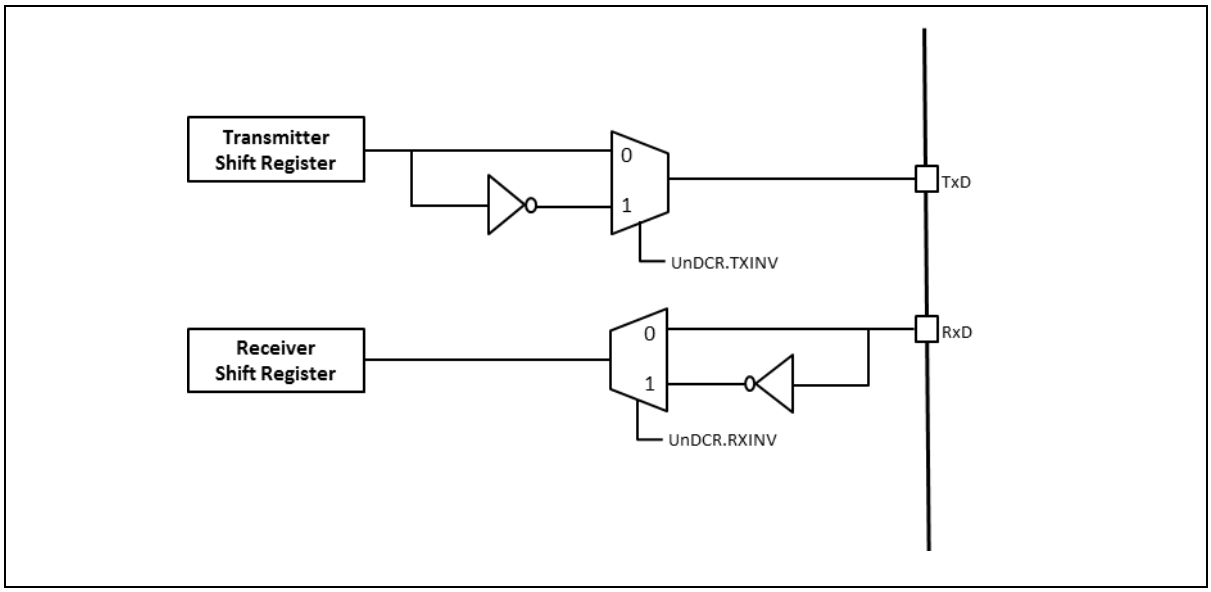


Figure 85. Data Inversion Control Diagram

15.2.7 UARTn_LSR: UARTn line status register

UARTn Line Status Register is 32-bit register. This Register is able to 32/16/8-bit access. (UART, n = 0, 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 bit.
		0 Transmit register has data or is transferring.
		1 Transmit register is empty.
5	THRE	Transmit Holding Empty bit.
		0 Transmit holding register is not empty.
		1 Transmit holding register is empty
		NOTE: This bit will be set to "1b" when it starts transmit.
4	BI	Break Condition Indication bit.
		0 Normal status.
		1 Break condition is detected.
3	FE	Frame Error Indicator bit.
		0 No frame error.
		1 Frame error occurs. The receive character did not have a valid stop bit.
2	PE	Parity Error Indicator bit.
		0 No parity error.
		1 Parity error occurs. The receive character have not correct parity information.
1	OE	Overrun Error Indicator bit.
		0 No overrun error.
		1 Overrun error occurs. Additional data arrive while the RHR is full.
0	DR	Data Receive Indicator bit.
		0 No data in receive holding register.
		1 Data has been received and is saved in the receive holding register.

This register provides the status of data transfers between transmitter and receiver. User can get the line status information from this register and can handle the next process. Bit 1,2,3,4 will arise the line status interrupt when RLSIE bit in UARTnIEN register is set. Other bits can generate its interrupt when its interrupt enable bit in UARTnIEN register is set.

15.2.8 UARTn_BDR: UARTn baud rate divisor latch register

UARTn Baud Rate Divisor Latch Register is 32-bit register. This Register is able to 32/16/8-bit access. (UART, n = 0, 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	
-		0x0000	
-		RW	

15 0	BDR	Baud Rate Divider Latch Value Baud rate = fUARTnCLK/(16 x BDR[15:0] x 2). **NOTE:** The UART block won't work if the BDR[15:0] ≤ 0x0003.

To establish the communication with UART channel, the baud rate should be set properly. The programmable baud rate generation is provided to give from 1 to 65535 divider number. The 16-bits divider register (UARTn_BDR) should be written for expected baud rate. $UART_{clock}$ is PCLK.

Baud rate calculation formula is as below.

$$\mathrm{BDR} = \frac{UART_{clock}}{16 \times \ BaudRate \ \times 2}$$

In case of 40 MHz $UART_{clock}$ speed, the divider value and error rate is described in Table 58.

Table 58. Example of Baud Rate Calculation (without BFR)

$UART_{clock}$ = 48 MHz		
Baud rate	**Divider**	**Error (%)**
1200	1250	0.0%
2400	625	0.0%
4800	312	0.16%
9600	156	0.16%
19200	78	0.16%
38400	39	0.16%
57600	26	0.16%
115200	13	0.16%

15.2.9 UARTn_BFR: UARTn baud rate fraction counter register

UARTn Baud Rate Fraction Counter Register is 32-bit register. This Register is able to 32/16/8-bit access. (UART, n = 0, 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
-			0x00
-			RW

7 0	BFR	Fraction Counter value. 0 Disable fraction counter. N Fraction compensation mode is operating. Fraction counter is incremented by FCNT. **NOTE:** 8-bit fractional counter will count up by FCNT value every (baud rate)/16 period and whenever fractional counter overflow is happen, the divisor value will increment by 1. So this period will be compensated. Then next period, the divisor value will return to original set value.

Table 59. Example of Baud Rate Calculation

$UART_{clock}$ =48 MHz			
Baud rate	**Divider**	**FCNT**	**Error (%)**
1200	1250	0	0.00%
2400	625	0	0.00%
4800	312	128	0.00%
9600	156	64	0.00%
19200	78	32	0.00%
38400	39	16	0.00%
57600	26	10	0.01%
115200	13	5	0.01%

$$\mathrm{FCNT} = \mathrm{Float} * 256$$

FCNT value can be calculated above equation. For example, if the target baud rate is 4800 bps and $UART_{clock}$ is 48MHz case, the BDR value is 312.5. The integer number 312 should be the BDR value and the FCNT value can be calculated by the floating number 0.5 as below.

FCNT = 0.5 * 256 = 128, so the FCNT value is 128.

8-bit fractional counter will be counted up by FCNT value every (baud rate)/16 periods and whenever fractional counter overflow is happened, the divider value will be incremented by 1. So this period will be compensated. Then in next period, the divider value will return to original set value.

15.2.10 UARTn_IDTR: UARTn inter-frame delay time register

UARTn Inter-Frame Delay Time Register is 32-bit register. This Register is able to 32/16/8-bit access. (UART, n = 0, 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
-			1	0	-	000
-			RW	RW	-	RW

7	SMS	Start Bit Multi Sampling Enable bit.
	0	Multi sampling is disable for start bit, Single sample will be done at 8/16 baud rate for the start bit.
	1	Multi sampling is enabled for start bit. Sampling is done 3 times at 7/16, 8/16 and 9/16 baud rate. Dominant value of 3 samples will be selected for the start bit.
6	DMS	Data Bit Multi sampling enable.
	0	Multi sampling is disable for data bit, Single sample will be done at 8/16 baud rate for the data bit.
	1	Multi sampling is enabled for data bit. Sampling is done 3 times at 7/16, 8/16 and 9/16 baud rate. Dominant value of 3 samples will be selected for the data bit.
2 0	WAITVAL	Wait Time Value. Dummy delay can be inserted between 2 Continuous Transmits.
		Wait Time = WAITVAL[2:0]/(Baud Rate)

15.3 Functional description

UART module is compatible with 16450 UART. Additionally, fractional baud rate compensation logic is provided.

It does not have an internal FIFO block.

15.3.1 USART clock generation

UARTs operates at following timing shown below.

If falling edge is detected on the receive line, UART considers it the start bit. From then on, UART oversamples 1-bit 16 times and detects the bit value at the 7th sample.

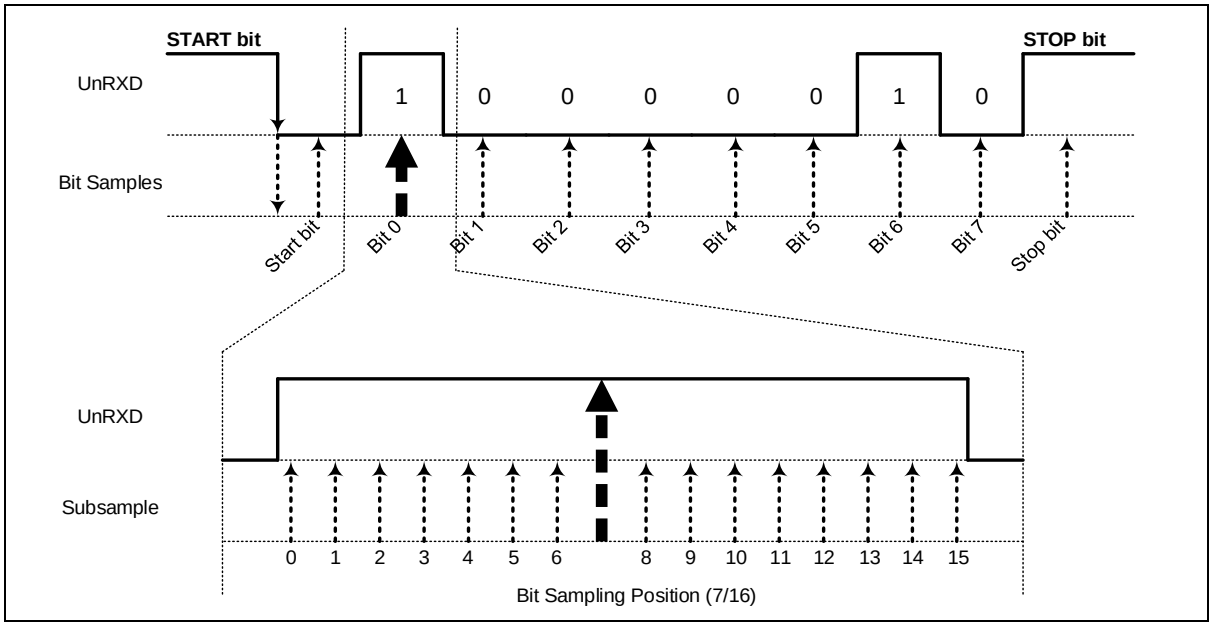


Figure 86. The Sampling Timing of UART Receiver

It is recommended to enable debounce settings in the PCU block to reinforce the immunity of external glitch noise.

15.3.2 Transmitter

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 the DLEN[1:0] filed in UARTn_LCR register.

The parity bit is set according to the PARITY and PEN bit filed in 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 filed in UARTn_LCR register.

The example of transmission data format is below.

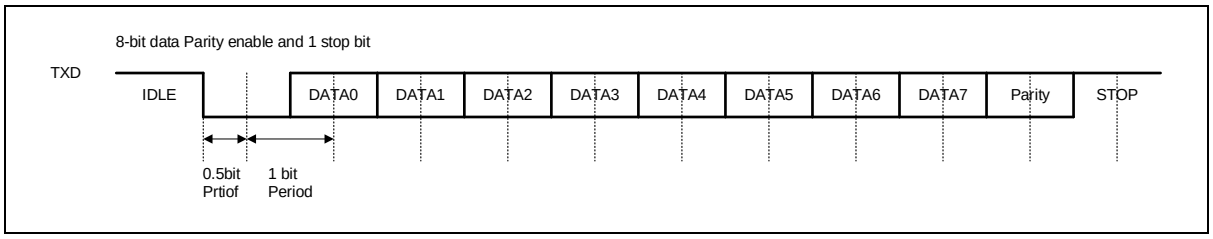


Figure 87. Transmit Data Format Example

15.3.3 Inter-frame delay transmission

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 WATIVAL field

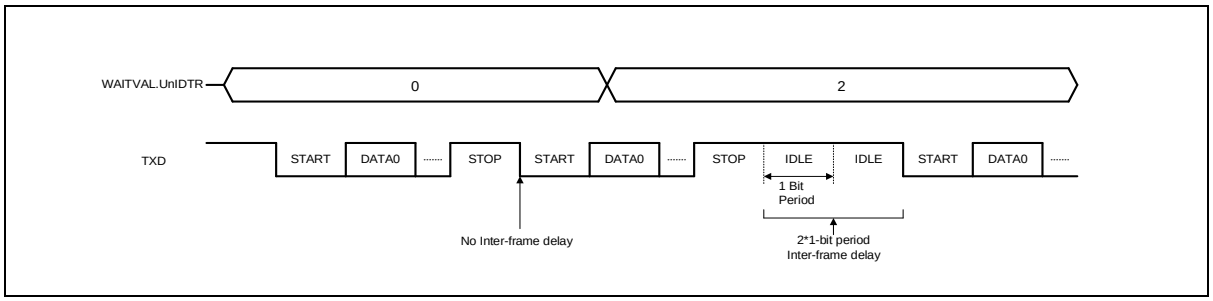


Figure 88. Inter-frame Delay Timing Diagram

15.3.4 Transmit interrupt

Transmit operation makes some kind of interrupt flags. When transmitter holding register is empty, the THRE interrupt flag will be set. When transmitter shifter register is empty, the TXE interrupt flag will be set. User can select which interrupt timing is best for the application.

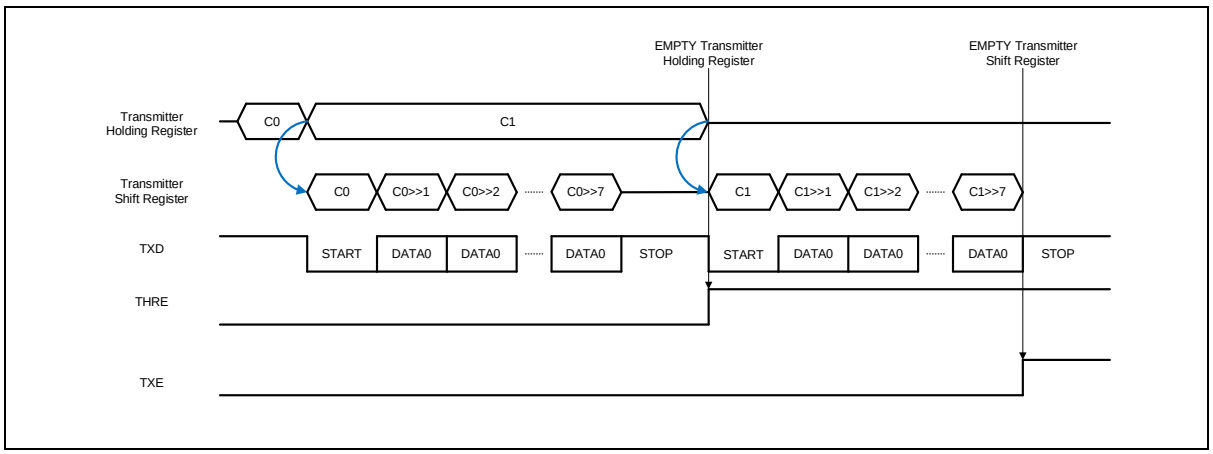


Figure 89. Transmit Interrupt Timing Diagram

16 I2C interface

I2C is one of industrial standard serial communication protocols, and which uses 2 bus lines such as Serial Data Line (SDAn) and Serial Clock Line (SCLn) to exchange data. Because both SDAn and SCLn lines are open-drain output, each line needs pull-up resistor respectively.

I2C features the followings (n = 0 and 1):

- Compatible with I2C bus standard.
- Multi-master operation.
- Up to 400KHz data transfer read speed.
- 7-bit address.
- Support two slave address.
- Both master and slave operation.
- Bus busy detection

Table 60 introduces pins assigned for I2C interface.

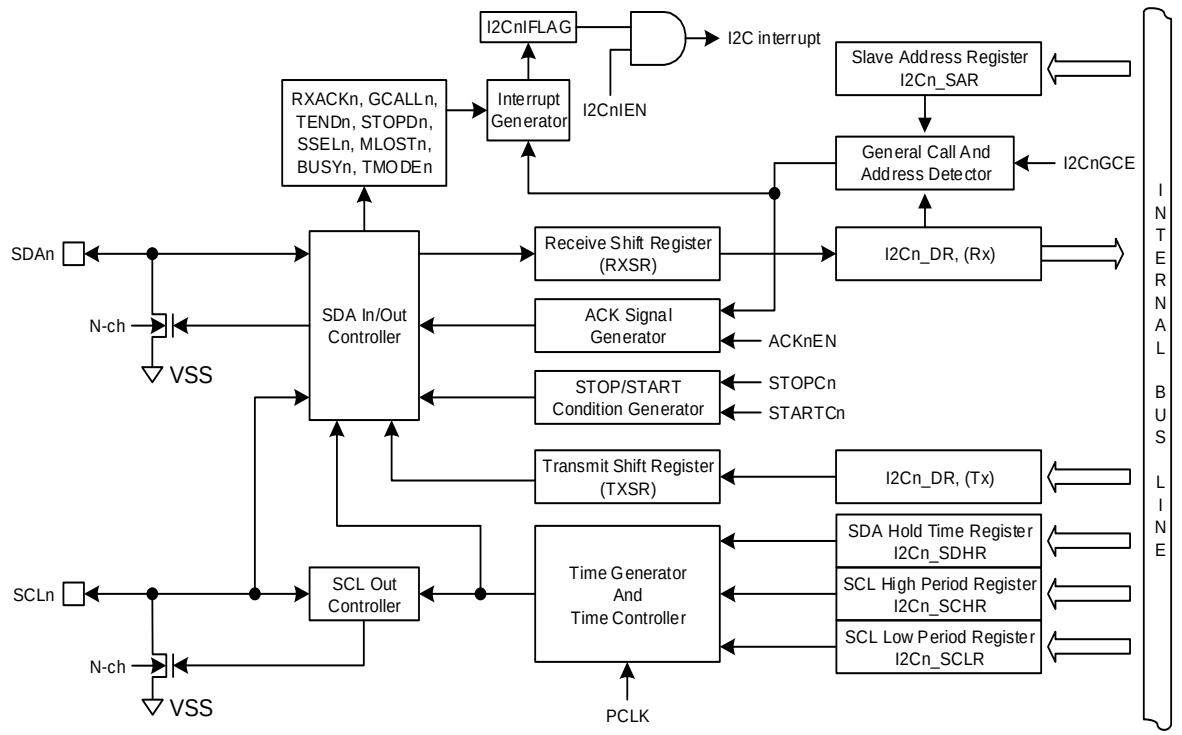
Table 60. Pin Assignment of I2C: External Pins

Pin name	Type	Description
SCL0	I/O	I2C channel 0 Serial clock bus line (open-drain)
SDA0	I/O	I2C channel 0 Serial data bus line (open-drain)
SCL1	I/O	I2C channel 1 Serial clock bus line
SDA1	I/O	I2C channel 1 Serial data bus line

NOTE: n = 0 and 1

16.1 I2C block diagram

In this section, I2C interface block is described in a block diagram.



NOTE: When the corresponding port of I2C is a sub-function for SCL/SDA pin, the SCL/SDA pins are automatically set to the N-channel open-drain outputs and the input latch is read in the case of reading the pins. The corresponding pull-up resistor is determined by the control register.

Figure 90. I2C Block Diagram

16.2 Registers

Base address of I2C is introduced in the followings:

Table 61. Base Address of I2C Interface

Name	Base address
I2C0	0x4000_4800
I2C1	0x4000_4900

Table 62. I2C Register Map

Name	Offset	Type	Description	Reset value	Ref.
I2Cn_CR	0x00	RW	I2Cn Control Register	0x0000_0000	16.2.1
I2Cn_ST	0x04	RW	I2Cn Status Register	0x0000_0000	16.2.2
I2Cn_SAR1	0x08	RW	I2Cn Slave Address Register 0	0x0000_0000	16.2.3
I2Cn_SAR2	0x0C	RW	I2Cn Slave Address Register 1	0x0000_0000	16.2.4
I2Cn_DR	0x10	RW	I2Cn Data Register	0x0000_0000	16.2.5
I2Cn_SDHR	0x14	RW	I2Cn SDA Hold Time Register	0x0000_0001	16.2.6
I2Cn_SCLR	0x18	RW	I2Cn SCL Low Period Register	0x0000_003F	16.2.7
I2Cn_SCHR	0x1C	RW	I2Cn SCL High Period Register	0x0000_003F	16.2.8

NOTE: n = 0 and 1

16.2.1 I2Cn_CR: I2Cn control register

The register can be set to configure I2C operation mode activate I2C transactions. I2Cn Control Register is a 32-bit register. This register is accessible in 32/16/8-bit. (n = 0, 1)

I2C0_CR=0x4000_4800, I2C1_CR=0x4000_4900

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 by supplying.
		0 Disable I2Cn block.
		1 Enable I2Cn block.
6	TXDLYENBn	I2Cn_SDHR Register Control bit.
		0 Enable I2Cn_SDHR register.
		1 Disable I2Cn_SDHR register.
5	I2CnIEN	I2Cn Interrupt Enable bit.
		0 Disable I2Cn interrupt.
		1 Enable I2Cn interrupt.
4	I2CnIFLAG	I2Cn Interrupt Flag bit. This bit is cleared when all interrupt source bits in the I2Cn_ST register are cleared to "0b".
		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. Where x = 0 and 1. 2. When received address packet equals to SLAx[6:0] bits in I2CnSARx register. 3. When received address packet equals to value 0x00 with GCALLn enabled. 4. When I2Cn operates as a receiver (master or slave)
2	IMASTERn	Represent Operation Mode of I2Cn. This bit is cleared to "0b" 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 STOP condition is to be generated.
0	STARTCn	START Condition Generation When I2Cn is master.
		0 No effect.
		1 START or Repeated START condition is to be generated.

16.2.2 I2Cn_ST: I2Cn status register

I2Cn Status Register is 32-bit register. This Register is able to 32/16/8-bit access. (I2C, n = 0, 1)

I2C0_ST=0x4000_4804, I2C1_ST=0x4000_4904

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 "1b" 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.
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.

16.2.3 I2Cn_SAR1: I2Cn slave address register 1

I2Cn Slave Address Register 0 is 32-bit register. This Register is able to 32/16/8-bit access. (I2C, n = 0, 1)

I2C0_SAR1=0x4000_4808, I2C1_SAR1=0x4000_4908

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	GCALLnEN
-			0000000	0
-			RW	RW

7 1	SLAn	These bits configure the slave address 0 in slave mode.
0	GCALLnEN	This bit decides whether I2Cn allows general call address 0 or not in I2Cn slave mode.
	0	Ignore general call address 0.
	1	Allow general call address 0.

16.2.4 I2Cn_SAR2: I2Cn slave address register 2

I2Cn Slave Address Register 1 is 32-bit register. This Register is able to 32/16/8-bit access. (I2C, n = 0, 1)

I2C0_SAR2=0x4000_480C, I2C1_SAR2=0x4000_490C

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	GCALLnEN
-			0000000	0
			RW	RW

7 1	SLAn	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.

16.2.5 I2Cn_DR: I2Cn data register

I2Cn Data Register is 32-bit register. This Register is able to 32/16/8-bit access. (I2C, n = 0, 1)

I2C0_DR=0x4000_4810, I2C1_DR=0x4000_4910

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
-			0x00
-			RW

7 0	DATA	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.

16.2.6 I2Cn_SDHR: I2Cn SDA hold time register

I2Cn SDA Hold Time Register is 32-bit register. This Register is able to 32/16/8-bit access. (I2C, n = 0, 1)

I2C0_SDHR=0x4000_4814, I2C1_SDHR=0x4000_4914

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
-			0x001
-			RW

11 0	HLDT	This register is used to control SDA output timing from the falling edge of SCL. **NOTES:** 1. 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 change in the middle of SCL. 2. In slave mode, configure this register regarding the frequency of SCL from master. 3. The SDA is changed after tPCLK X (I2Cn_SDHR+2) in master mode. So, to insure operation in slave mode, the value tPCLK X (I2Cn_SDHR + 2) must be smaller than the period of SCL.

16.2.7 I2Cn_SCLR: I2Cn SCL low period register

I2Cn_SCL Low Period Register is 32-bit register. This Register is able to 32/16/8-bit access. (I2C, n = 0, 1)

I2C0_SCLR=0x4000_4818, I2C1_SCLR=0x4000_4918

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											
-																				0x03F											
-																				RW											

11 0	SCLL	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 X (4 X I2Cn_SCLR + 2) where tPCLK is the period of PCLK.

16.2.8 I2Cn_SCHR: I2Cn SCL high period register

I2Cn SCL High Period Register is 32-bit register. This Register is able to 32/16/8-bit access. (I2C, n = 0, 1)

I2C0_SCHR=0x4000_481C, I2C1_SCHR=0x4000_491C

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											
-																				0x03F											
-																				RW											

11 0	SCLH	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 X (4 X I2Cn_SCHR + 2) where tPCLK is the period of PCLK.

16.3 Functional description

16.3.1 I2C bit transfer

Data on the SDAn line must be stable during HIGH period of the clock, SCLn. 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.

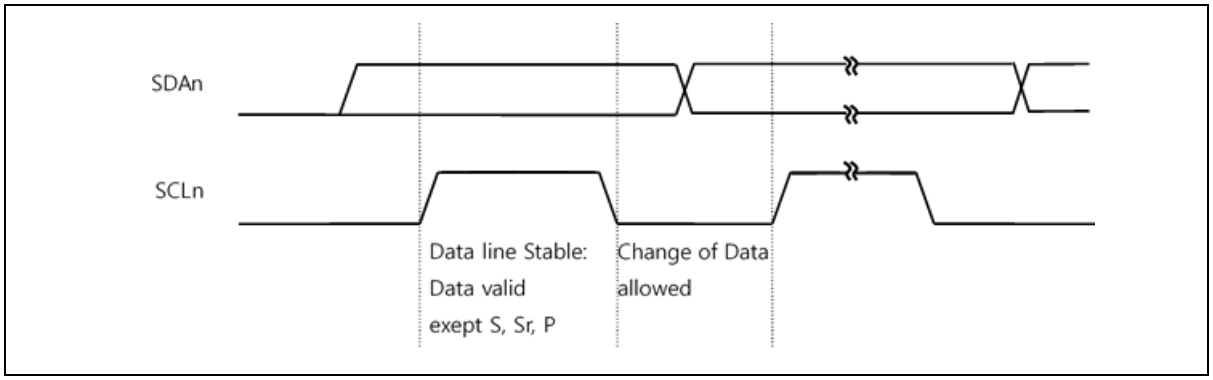


Figure 91. I2C Bus Bit Transfer (n = 0 and 1)

16.3.2 START/repeated START/STOP

One master can issue a START (S) condition to notice other devices connected to the SCLn, SDAn lines that it 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 busy. So, the START and repeated START conditions are functionally identical.

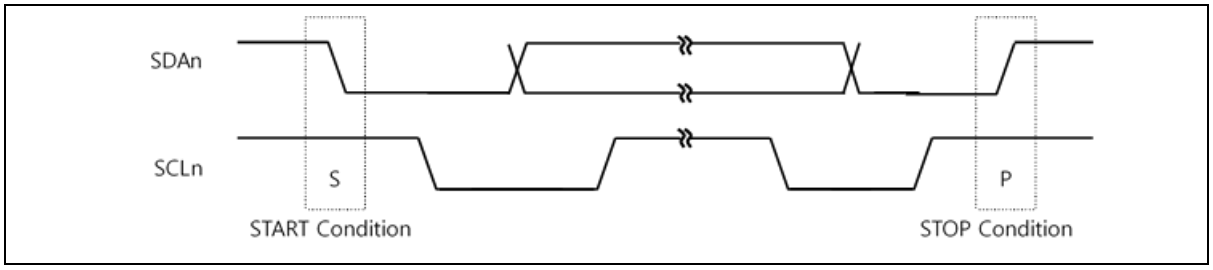


Figure 92. START and STOP Condition (n = 0 and 1)

16.3.3 Data transfer

Every byte put on the SDAn line must be 8-bits long. 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 can't 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.

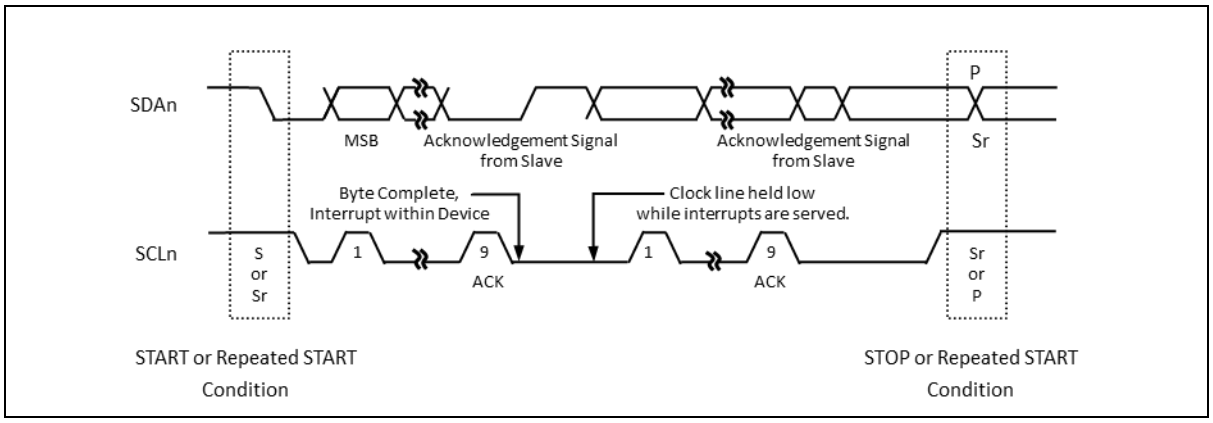


Figure 93. I2C Bus Data Transfer (n = 0 and 1)

16.3.4 Acknowledge

The acknowledge related 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 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's performing some real time function, the data line must be left HIGH by the slave.

And also, 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 any 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.

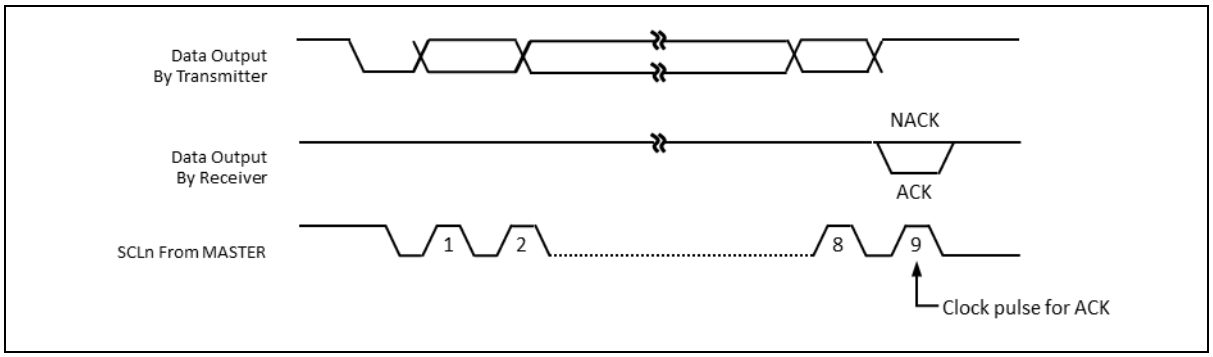


Figure 94. I2C Bus Acknowledge (n = 0 and 1)

16.3.5 Synchronization/ 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 the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output state because the level on the bus doesn't 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.

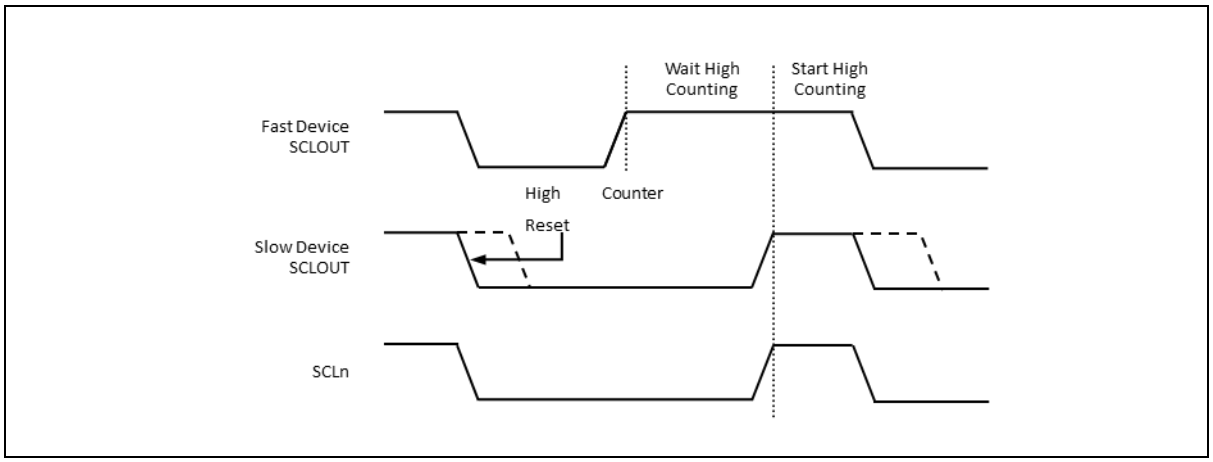


Figure 95. Clock Synchronization during the Arbitration Procedure (n = 0 and 1)

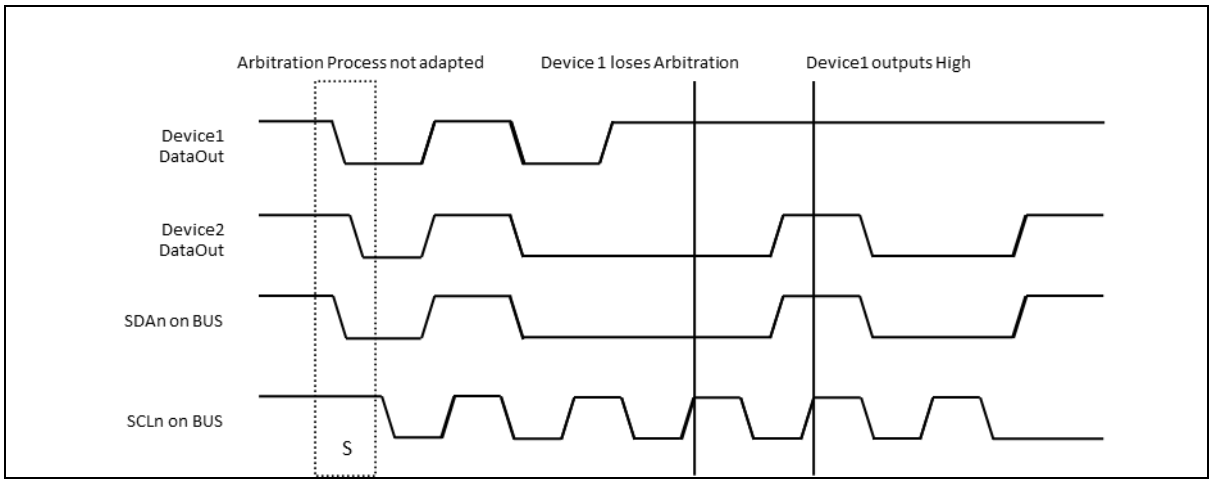


Figure 96. Arbitration Procedure between Two Masters (n = 0 and 1)

16.3.6 I2C operation

The I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events except for a transmission of a START condition. Because the I2C is interrupt based, the application software is free to carry on other operations during an I2C byte transfer.

Note that when an I2C interrupt is generated, I2CnIFLAG flag in I2CnIEN register is set, it is cleared when all interrupt source bits in the I2Cn_ST register are cleared to "0b". When I2C interrupt occurs, the SCLn line is hold LOW until clearing "0b" all interrupt source bits in I2Cn_ST register. 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 4 modes by configuring master/slave, transmitter/receiver. The operating mode is configured by a winning master. A more detailed explanation follows below. (n = 0 and 1)

Master transmitter

To operate I2C in master transmitter, follow the recommended steps below:

1. Enable I2C by setting I2CnEN bit in I2Cn_CR. This provides main clock to the peripheral.
2. Load SLA+W into the I2Cn_DR where SLA is address of slave device and W is transfer direction from the viewpoint of the 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. And 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 the reception of 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 operate 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 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. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (master) can choose one of the following cases regardless of the reception of ACK signal from slave.

   Case 1. Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2Cn_DR.

   Case 2. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC bit in I2Cn_CR.

   Case 3. Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in I2Cn_CR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line.

   For the Case 1, move to step 7. For the Case 2, move to step 9 to handle STOP interrupt. For the Case 3, move back 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 being transmitted. During data transfer, bus arbitration continues.
8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCLn LOW. When I2C loses bus mastership while transmitting data arbitrating 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 the reception of ACK signal from slave.

Case A. Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2Cn_DR.

Case B. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in I2CnCR.

Case C. Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set the STARTCn bit in I2Cn_CR.

After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCL line. For the Case A, move back to step 7. For the Case B, move to step 9 to handle STOP interrupt. For the Case C, move back to step 6 after transmitting the data in I2C_DR, 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 "0" to I2CnST. After this, I2C enters in idle state.

Master receiver

To operate I2C in master receiver, follow the recommended steps below:

1. Enable I2C by setting I2CnEN bit in I2Cn_CR. This provides main clock to the peripheral.
2. Load SLA+R into the I2Cn_DR where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. 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 I2CnSDHR 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.
5. Set the STARTCn bit in I2Cn_CR. This transmits a START condition. And 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 the reception of 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 operate 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 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. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (master) can choose one of the following cases according to the reception of ACK signal from slave.

   Case 1. Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKnEN bit in I2Cn_CR to decide whether I2C Acknowledges the next data to be received or not.

   Case 2. Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOPCn bit in I2Cn_CR.

   Case 3. Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in I2Cn_CR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line. For the Case 1, move to step 7. For the Case 2, move to step 9 to handle STOP interrupt. For the Case 3, move to step 6 after transmitting the data in I2Cn_DR and if transfer direction bit is '0' go to master transmitter section.
7. 1-Byte of data is being received.
8. This is ACK signal processing stage for data packet 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 I2Cn_ST.

Case A. Master continues to receive data from slave. To do this, set ACKnEN bit in I2Cn_CR to acknowledge the next data to be received.

Case B. Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKnEN bit in I2Cn_CR.

Case C. Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPCn bit in I2Cn_CR.

Case D. No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA+R/W into the I2CnDR and set the STARTCn bit in I2Cn_CR.

After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line. For the Case A and B, move to step 7. For the Case C, move to step 9 to handle STOP interrupt. For the Case 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 I2Cn_ST, write "0" value to I2Cn_ST. After this, I2C enters idle state.

Slave transmitter

To operate I2C in slave transmitter, follow the recommended steps below:

1. If the main operating clock (SCLK) 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 x period of SCLK where SDAH is multiple of number of SCLK coming from I2CnSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting I2CnIEN bit and I2CnEN bit in I2Cn_CR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLAn bits in I2Cn_SAR1/2. If the GCALLnEN bit in I2Cn_SAR1/2 is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLAn bits in I2Cn_SAR, I2C enters idle state i.e., waits for another START condition. Else if the address equals to 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 equals to SLAn bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs, load transmit data to I2Cn_DR and clear to "0b" all interrupt source bits in I2Cn_ST to release SCLn line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases:

   Case 1. No ACK signal is detected and I2C waits STOP or repeated START condition.

   Case 2. ACK signal from master is detected. Load data to transmit into I2Cn_DR.

   After doing one of the actions above, clear all interrupt source bits in I2Cn_ST to "0b" to release SCLn line. For the Case 1, move to step 7 to terminate communication. For the Case 2, move back to step 5. In either case, a repeated START condition can be detected. For that case, move back 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 I2Cn_ST, write "0" to I2Cn_ST. After this, I2C enters idle state.

Slave receiver

To operate I2C in slave receiver, follow the recommended steps below:

1. If the main operating clock (SCLK) 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 x period of SCLK where SDAH is multiple of number of SCLK coming from I2Cn_SDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting I2CnIEN bit in I2CnCR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2C_SAR. If the GCALLnEN bit in I2Cn_SAR1/2 is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLAn bits in I2Cn_SAR1/2, I2C enters idle state i.e., waits for another START condition. Else if the address equals to 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 equals to SLA bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs and I2C is ready to receive data, clear to "0b" all interrupt source bits in I2Cn_ST to release SCLn line.
5. 1-Byte of data is being received.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases:

   Case 1. No ACK signal is detected (ACKnEN=0) and I2C waits STOP or repeated START condition.

   Case 2. ACK signal is detected (ACKnEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, clear all interrupt source bits in I2CnST to "0b" to release SCLn line. For the Case 1, move to step 7 to terminate communication. For the Case 2, move back to step 5. In either case, a repeated START condition can be detected. For that case, move back 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 I2CnST, write "0" to I2CnST. After this, I2C enters idle state.

17 Serial Peripheral Interface (SPI)

A channel serial interface is provided for synchronous serial communications with external peripherals. SPI block support both of master and slave mode. Four signals will be used for SPI communication such as SS, SCK, MOSI, and MISO.

SPI of A31G21x series features the followings:

- Master or Slave operation.
- Programmable clock polarity and phase.
- 8, 9, 16, 17-bit wide transmit/receive register.
- 8, 9, 16, 17-bit wide data frame.
- Loop-back mode.
- Programmable start, burst, and stop delay time.
- DMA transfer operation.

Table 63 introduces pins assigned for SPI.

Table 63. Pin Assignment of SPI: External Pins

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)

NOTE: n = 20 and 21

17.1 SPI block diagram

In this section, SPI is described in a block diagram in Figure 97.

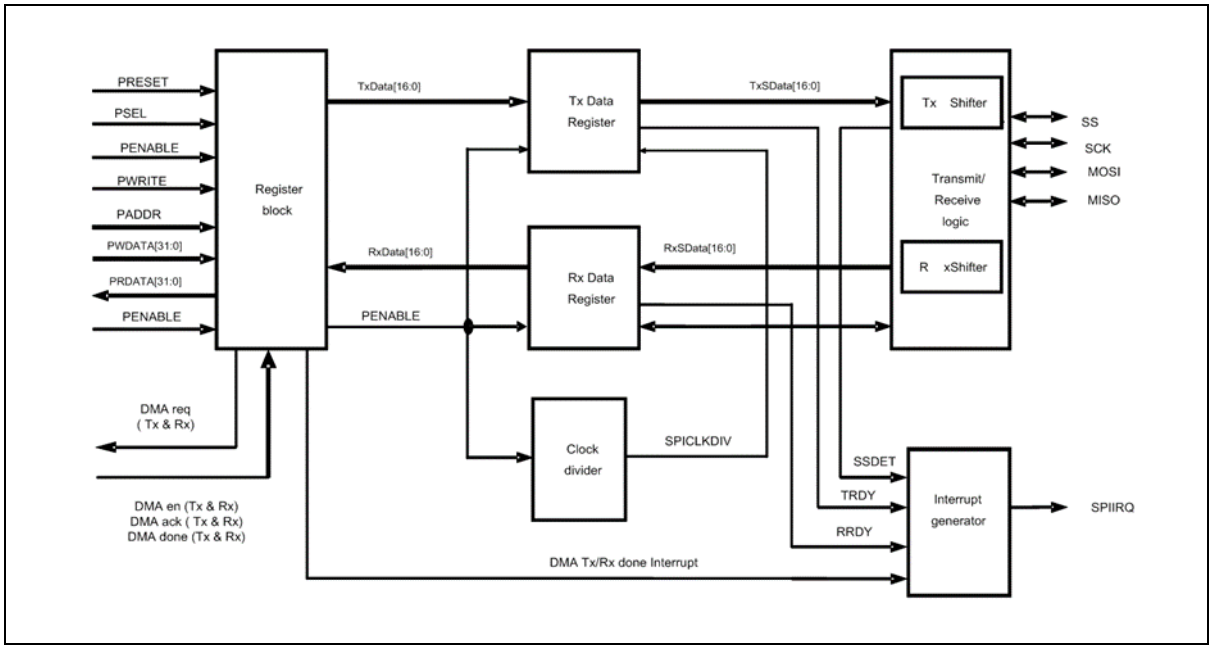


Figure 97. SPI Block Diagram

17.2 Registers

Base address of SPI is introduced in the followings:

Table 64. Base Address of SPI

Name	Base address
SPI20	0x4000_4C00
SPI21	0x4000_4D00

Table 65. SPI Register Map

Name	Offset	Type	Description	Reset value	Reference
SPIn_TDR	0x00	WO	Transmit data register	0x0000_0000	17.2.1
SPIn_RDR	0x00	RO	Receive data register	0x0000_0000	17.2.2
SPIn_CR	0x04	RW	Control register	0x0000_1020	17.2.3
SPIn_SR	0x08	RW	Status register	0x0000_0006	17.2.4
SPIn_BR	0x0C	RW	Baud-rate register	0x0000_00FF	17.2.5
SPIn_EN	0x10	RW	Enable register	0x0000_0000	17.2.6
SPIn_LR	0x14	RW	Delay length register	0x0001_0101	17.2.7

17.2.1 SPIn_TDR: SPI transmit data register

SPIn_TDR is a 17-bit sized read/write register. It contains serial transmit data.

SPI20_TDR=0x4000_4C00, SPI21_TDR=0x4000_4D00

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		TDR	
-		0x00000	
-		WO	

16 0	TDR	Transmit data register

17.2.2 SPIn.RDR: SPI receive data register

SPIn_RDR is a 17-bit sized read/write register. It contains serial receive data.

SPI20_RDR=0x4000_4C00, SPI21_RDR=0x4000_4D00

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		RDR	
-		0x00000	
-		RO	

16 0	RDR	Receive Data Register

17.2.3 SPIn_CR: SPI control register

SPIn_CR is a 20-bit sized read/write register and can be set to configure SPI operation mode.

SPI20_CR=0x4000_4C04, SPI21_CR=0x4000_4D04

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		TXBC	RXBC	DTXIE	DRXIE	SSCIE	TXIE	RXIE	SSMOD	SSOUT	LBE	SSMARK	SSMO	SSPOL	Reserved	MS	MSBF	CPHA	CPOL	BITSZ
-		0	0	0	0	0	0	0	0	1	0	0	0	0	-	1	0	0	0	00
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW

20	TXBC	Whether or not to clear the Tx buffer	
		0	Does not clear.
		1	Clears the Tx buffer.
19	RXBC	Whether or not to clear the Rx buffer clear	
		0	Does not clear.
		1	Clears the Rx buffer.
18	TXDIE	Whether to enable or disable the DMA Tx complete interrupt	
		0	Disables the DMA Tx complete interrupt.
		1	Enables the DMA Tx complete interrupt.
17	RXDIE	Whether to enable or disable the DMA Rx complete interrupt	
		0	Disables the DMA Rx complete interrupt.
		1	Enables the DMA Rx complete interrupt.
16	SSCIE	Whether to enable or disable the SS edge change interrupt	
		0	Disables the nSS interrupt.
		1	Enables interrupts for both nSS edges (L→H, H→L).
15	TXIE	Whether to enable or disable the transmit interrupt	
		0	Disables the transmit interrupt.
		1	Enables the transmit interrupt.
14	RXIE	Whether to enable or disable the receive interrupt	
		0	Disables the receive interrupt.
		1	Enables the receive interrupt.
13	SSMOD	SS auto/manual output selection in master mode	
		0	The SS output cannot be selected with the SSOUT (.CR [12]) bit. Outputs are automatically generated, according to the data input in the normal operating mode of the SS signal.
		1	SS signal outputs are generated, according to the SSOUT value.
12	SSOUT	SS output signal configuration in master mode	
		0	Low SS output
		1	High SS output
11	LBE	Loop-back mode selection in master mode	
		0	Does not use loop-back mode.
		1	Uses loop-back mode.
10	SSMASK	Whether to enable or disable the masking of SS signals in slave mode	
		0	Does not mask the signal from pin SS. Data is received if the SS signal is active.
		1	Masks the signal from pin SS. Data is received on the SCLK edge, regardless of the SS signal's value.
9	SSMO	SS output signal selection	
		0	Disables the SS output signal.
		1	Enables the SS output signal.
8	SSPOL	SS signal polarity selection	
		0	Active-Low.
		1	Active-High.
5	MS	Master or slave selection	
		0	SPI slave mode

		1	SPI master mode
4	MSBF	MSB- or LSB-first transmission selection	
		0	Selects LSB-first transmission.
		1	Selects MSB-first transmission.
3	CPHA	SPI clock phase	
		0	Samples data generated at odd number edges (1, 3, 5…15).
		1	Samples data generated at even number edges (2, 4, 6…16).
2	CPOL	SPI clock polarity	
		0	Selects an active-high clock.
		1	Selects an active-low clock.
1 0	BITSZ	Transmit and receive bit size selection	
		00	8-bit
		01	9-bit
		10	16-bit
		11	17-bit

Table 66. SPI Phase and Clock Configuration

SPIn Mode	CPOLn	CPHAn	Status
0	0	0	• Samples data at rising edge. • Changes data at falling edge.
1	0	1	• Changes data at rising edge. • Samples data at falling edge.
2	1	0	• Samples data at falling edge. • Changes data at rising edge.
3	1	1	• Changes data at falling edge. • Samples data at rising edge.

17.2.4 SPIn_SR: SPI status register

SPIn_SR is a 10-bit sized read/write register. It contains the status of SPI interface.

SPI20_SR=0x4000_4C008, SPI21_SR=0x4000_4D08

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			TXDMAF	RXDMAF	SBUSY	SSDET	SSON	OVRF	UDRF	TXIDLE	TRDY	RRDY
-			0	0	0	0	0	0	0	1	1	0
-			RC	RC	RO	RC	RC	RC	RC	RO	RO	RO

9	TXDMAF	DMA transmit complete flag (DMA → SPI)
		0 Either a DMA transmission is in progress or is not being used.
		1 A DMA transmission has been completed.
8	RXDMAF	DMA receive complete flag (DMA → SPI)
		0 Either a DMA reception is in progress or is not being used.
		1 A DMA reception has been completed.
7	SBUSY	Transmit or receive flag
		0 Idle state
		1 A transmission or reception is in progress.
6	SSDET	SS signal rising or falling edge detection flag
		0 No SS edge has been detected.
		1 An SS edge has been detected. Write a 0 to clear the bit.
5	SSON	SS signal status flag
		0 Disables the SS signal.
		1 Enables the SS signal.
4	OVRF	Receive overrun error flag
		0 No receive overrun error present.
		1 A receive overrun error has been detected. The bit becomes cleared if you read or write to the RDR bit.
3	UDRF	Transmit underrun error flag
		0 No transmit underrun error present.
		1 A transmit underrun error has been detected. The bit becomes cleared if you read or write to the TDR bit.
2	TXIDLE	Transmit flag
		0 An SPI data transmit is in progress.
		1 The SPI is idle.
1	TRDY	Transmit buffer empty flag
		0 The transmit buffer is being used.
		1 The transmit buffer is ready for use. The TRDY bit is cleared if data is written to the TDR.
0	RRDY	Receive buffer ready flag
		0 The receive buffer holds no data.
		1 The receive buffer holds data. The RRDY bit is cleared by reading the RDR bit.

17.2.5 SPIn_BR: SPI baud rate register

SPIn_BR is a 16-bit sized read/write register. Baud rate can be set by writing the register.

SPI20_BR=0x4000_4C0C, SPI21_BR=0x4000_4D0C

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		BR	
-		0x00FF	
-		RW	

15 0	BR	Baud-rate configuration Baud-rate = PCLK/(BR + 1). (The baud-rate must be larger than "0". BR >= 2)

17.2.6 SPIn_EN: SPI enable register

SPIn_EN is a bit sized read/write register. It contains SPI enable bit.

SPI20_EN=0x4000_4C10, SPI21_EN=0x4000_4D10

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				ENABLE
-				0
-				RW

0	ENABLE	Whether to enable or disable the SPI
		Disables the SPI. - If "0" is written to this bit, the SR is initialized, while other registers are not.
		Enables the SPI. - If the bit is set to 1, dummy data in the transmit buffer is shifted. To prevent this, data must be written to the TDR before this bit is set enabled.

17.2.7 SPIn_LR: SPI delay length register

SPIn_LR is a 24-bit sized read/write register. It contains start, burst, and stop length value.

SPI20_LR=0x4000_4C14, SPI1_LR=0x4000_4D14

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	SPL	BTL	ST
-	0x01	0x01	0x01
-	RW	RW	RW

23 16	SPL	Stop length 0x01 ~ 0xFF: 1 ~ 255 SCLKs. (SPL >= 1)
15 8	BTL	Burst length 0x01 ~ 0xFF: 1 ~ 255 SCLKs. (BTL >= 1)
7 0	STL	Start length 0x01 ~ 0xFF: 1 ~ 255 SCLKs. (STL >= 1)

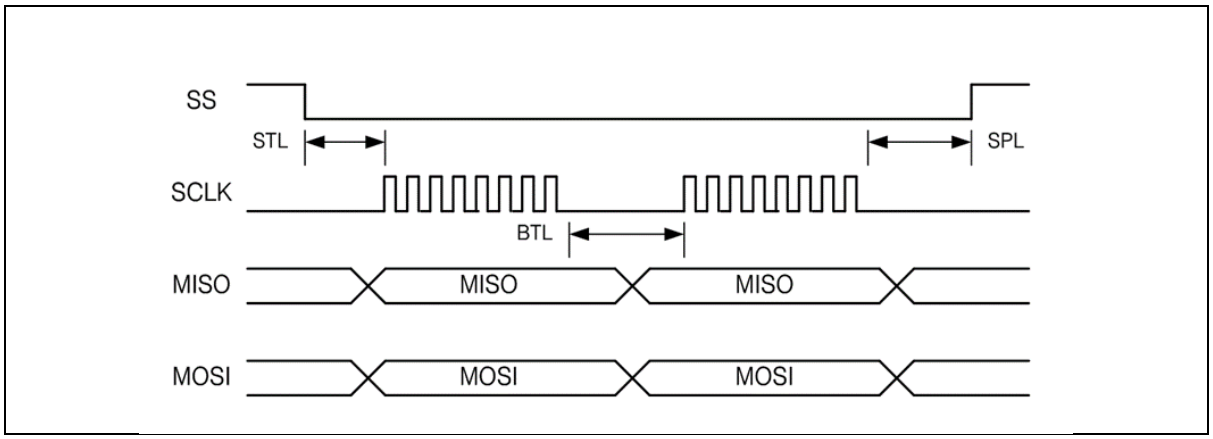


Figure 98. SPI Wave Form (STL, BTL and SPL)

17.3 Functional description

SPI Transmit block and Receive block share Clock Gen Block but they are independent each other. Transmit block and Receive block have double buffers and SPI is available for back to back transfer operation.

17.3.1 SPI timing

SPI has four modes of operation. These modes essentially control the way data is clocked in or out of an SPI device. The configuration is done by two bits in the SPI control register (SPInCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock. The clock phase (CPHA) control bit selects one of the two fundamentally different transfer formats.

To ensure a proper communication between master and slave both devices have to run in the same mode. This can require a reconfiguration of the master to match the requirements of different peripheral slaves.

The clock polarity has no significant effect on the transfer format. Switching this bit causes the clock signal to be inverted (active high becomes active low and idle low becomes idle high). The settings of the clock phase, however, selects one of the two different transfer timings, which are described closer in the next two chapters. Since the MOSI and MISO lines of the master and the slave are directly connected to each other, the diagrams show the timing of both device, master and slave.

The nSS line is the slave select input of the slave. The nSS pin of the master is not shown in the diagrams. It has to be inactive by a high level on this pin (if configured as input pin) or by configuring it as an output pin.

The timing of a SPI transfer where CPHA is zero is shown in figures 98 and 99. Two wave forms are shown for the SCK signal -one for CPOL equals zero and another for CPOL equals one.

When the SPI is configured as a slave, the transmission starts with the falling edge of the /SS line. This activates the SPI of the slave and the MSB of the byte stored in its data register (SPInTDR) is output on the MISO line. The actual transfer is started by a software write to the SPInTDR of the master. This causes the clock signal to be generated. In cases where the CPHA equals zero, the SCLK signal remains zero for the first half of the first SCLK cycle. This ensures that the data is stable on the input lines of both the master and the slave.

The data on the input lines is read with the edge of the SCLK line from its inactive to its active. The edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one) causes the data to be shifted one bit further so that the next bit is output on the MOSI and MISO lines.

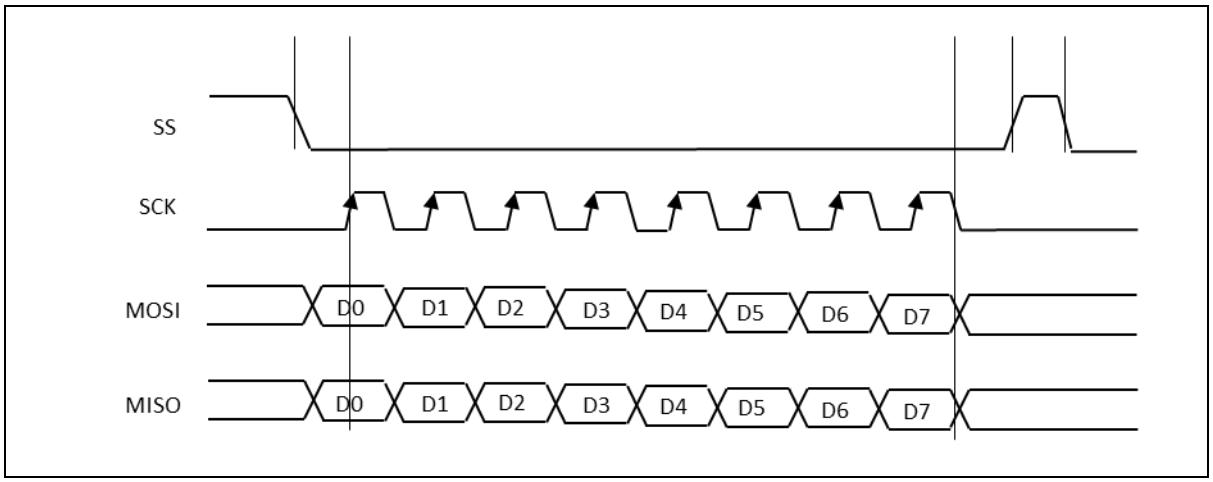


Figure 99. SPI Transfer Timing 1/4 (CPHA=0, CPOL=0, MSBF=0)

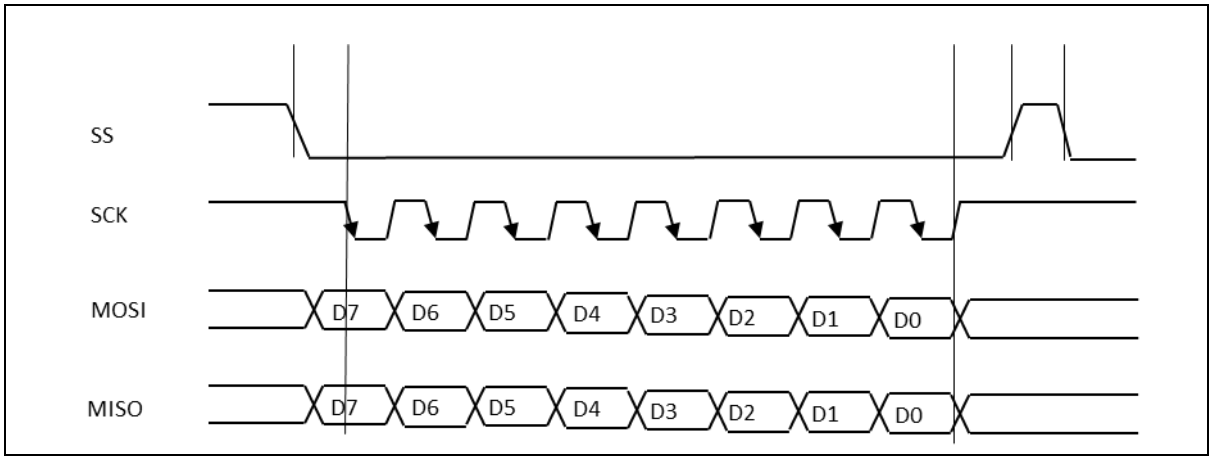


Figure 100. SPI Transfer Timing 2/4 (CPHA=0, CPOL=1, MSBF=1)

The timing of a SPI transfer where CPHA is one is shown in figures 5 and 6. Two wave forms are shown for the SCLK signal -one for CPOL equals zero and another for CPOL equals one.

Like in the previous cases the falling edge of the nSS lines selects and activates the slave. Compared to the previous cases, where CPHA equals zero, the transmission is not started and the MSB is not output by the slave at this stage. The actual transfer is started by a software write to the SPInTDR of the master what causes the clock signal to be generated. The first edge of the SCLK signal from its inactive to its active state (rising edge if CPOL equals zero and falling edge if CPOL equals one) causes both the master and the slave to output the MSB of the byte in the SPInTDR.

As shown in figures 3 and 4, there is no delay of half a SCLK-cycle. The SCLK line changes its level immediately at the beginning of the first SCLK-cycle. The data on the input lines is read with the edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one). After eight clock pulses the transmission is completed.

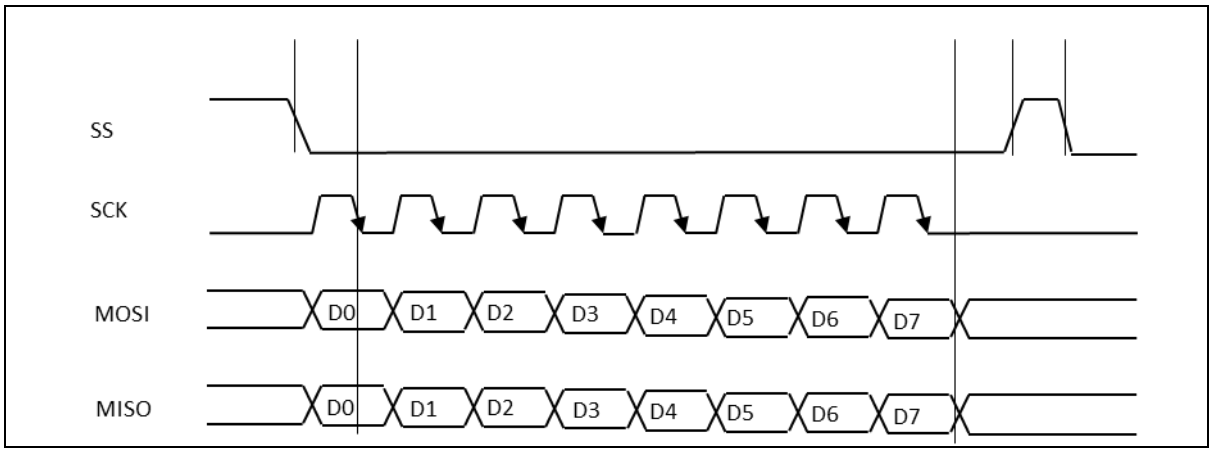


Figure 101. SPI Transfer Timing 3/4 (CPHA=1, CPOL=0, MSBF=0)

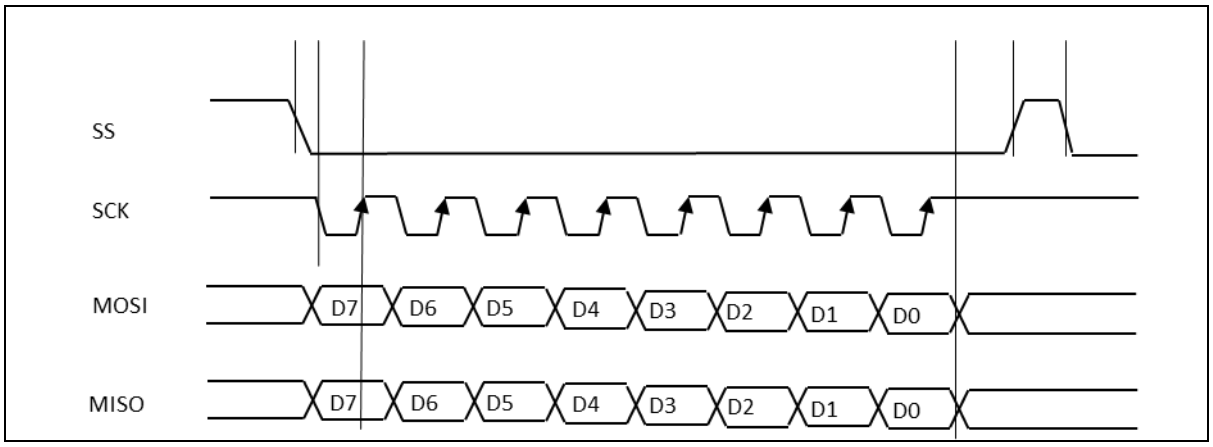


Figure 102. SPI Transfer Timing 4/4 (CPHA=1, CPOL=1, MSBF=1)

17.3.2 DMA handshake

SPI supports DMA handshaking operation. In order to operate DMA handshake, DMA registers should be set first (see chapter 7. DMAC). As Transmitter and Receiver are independent each other, SPI can operate the two channels at the same time.

After DMA channel for receiver is enabled and receive buffer is filled, SPI sends Rx request to DMA to empty the buffer and waits ACK signal from DMA. If Receive buffer is filled again after ACK signal, SPI sends Rx request. If DMA Rx DONE becomes high, RXDMAF (SPInSR[8]) goes "1" and an interrupt is serviced when RXDIE (SPInCR[17]) is set.

Likewise, if transmit buffer is empty after DMA channel for transmitter is enabled, SPI sends Tx request to DMA to fill the buffer and waits ACK signal from DMA. If transmit buffer is empty again after ACK signal, SPI sends Tx request. If DMA Tx DONE becomes high, TXDMAF (SPInSR[9]) goes "1" and an interrupt is serviced when TXDIE (SPInCR[18]) is set.

Slave transmitter sends dummy data at the first transfer (8 to 17 SCLKs) in DMA handshake mode.

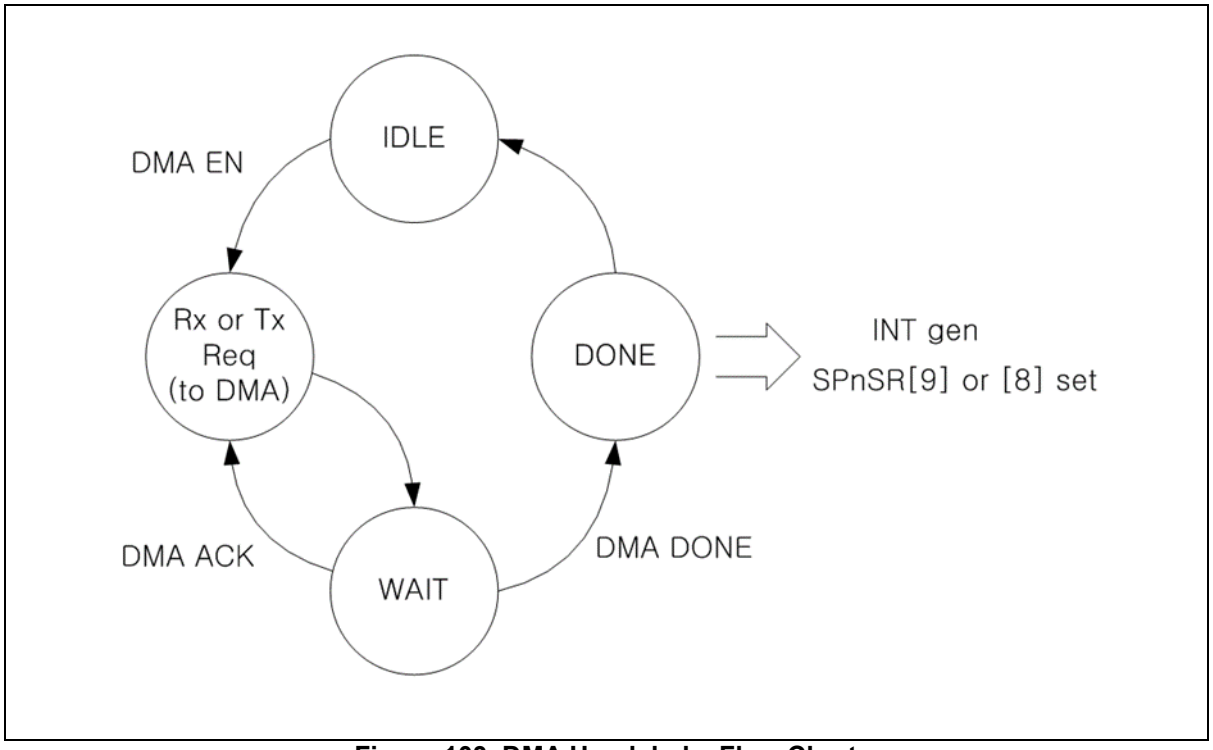


Figure 103. DMA Handshake Flow Chart

18 12-bit ADC

Analog-to-digital converter (A/D) allows conversion of an analog input signal to corresponding 12-bit digital value. The A/D module has 14 analog inputs. The output of the multiplexer is the input into the converter which generates the result through successive approximation. The A/D module has three registers which are the A/D converter control register (ADC_CR), A/D converter data register (ADC_DR) and A/D converter prescaler data register (ADC_PREDR). The channels to be converted are selected by setting ADSEL[3:0]. The register ADC_DRH and ADC_DRL contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADC_DR, the A/D conversion status bit AFLAG is set to '1', and the A/D interrupt is set. During A/D conversion, AFLAG bit is read as '0'.

ADC block of A31G21x series consists of an independent ADC unit featuring the followings:

- 14 channels of analog inputs
- S/W (ADST) and Timer trigger: TIMER10/11/12 A match and ADC trigger signal from TIMER30 support
- Maximum 4.5MHz conversion rate (Max. 150Ksps)
- Conversion time : 30 clock
- 6-bit Prescaler

Table 67 introduces pins assigned for ADC.

Table 67. Pin Assignment of ADC: External Signal

Pin name	Type	Description
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

18.1 12-bit ADC block diagram

In this section, 12-bit ADC is described in a block diagram in Figure 104.

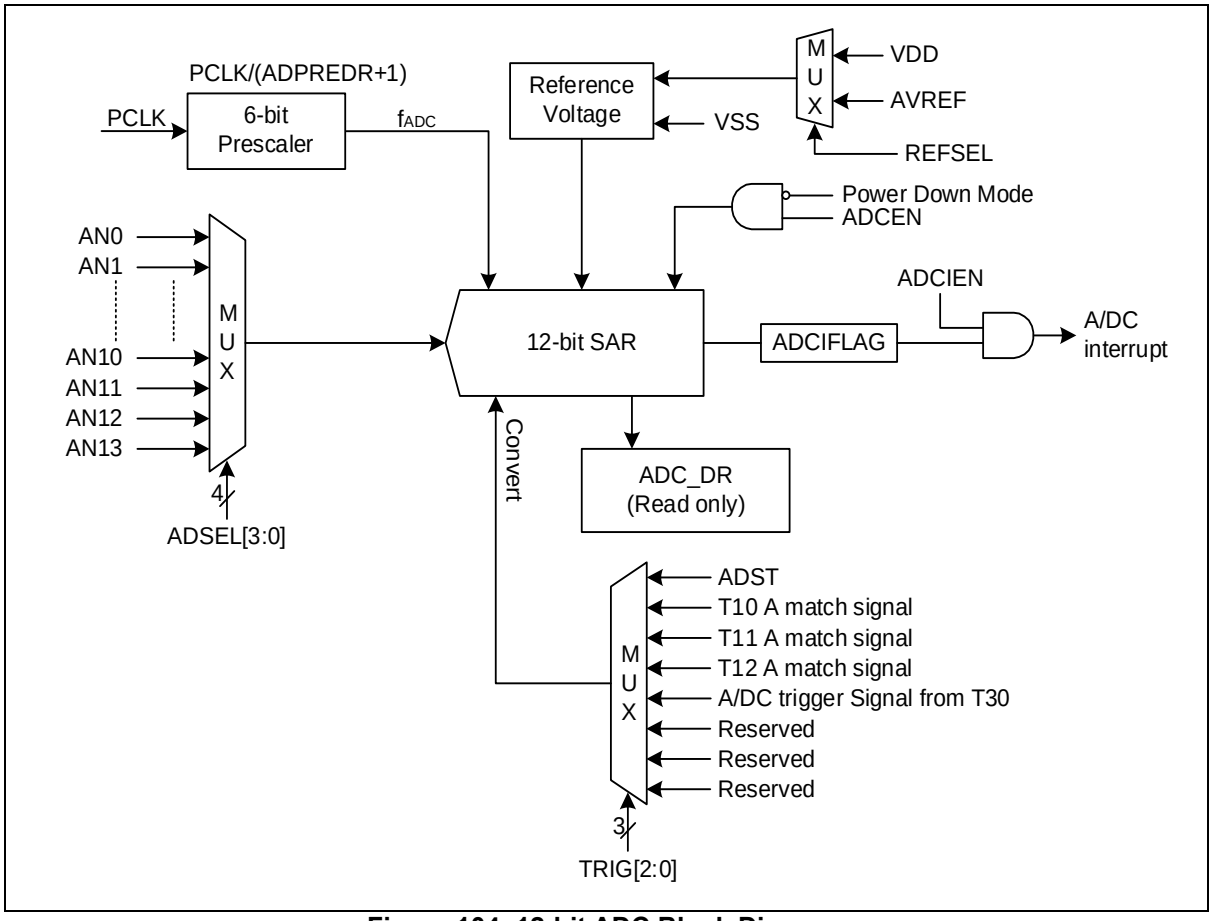


Figure 104. 12-bit ADC Block Diagram

18.2 Registers

Base address of ADC unit is introduced in the followings:

Table 68. Base Address of 12-bit ADC

Name	Base address
ADC	0x4000_3000

Table 69. 12-bit ADC Register Map

Name	Offset	Type	Description	Reset value	Ref.
ADC_CR	0x00	RW	A/D Converter Control Register	0x0000_0000	18.2.1
ADC_DR	0x04	RO	A/D Converter Data Register	0x0000_XXXX	18.2.2
ADC_PREDR	0x08	RW	A/D Converter Prescaler Data Register	0x0000_000F	18.2.3

18.2.1 ADC_CR: A/D converter control register

A/D Converter module should be configured properly before running. A/D Converter Control Register is 32-bit register. This Register is able to 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	REFSEL	Reserved	ADST	Reserved	ADCIEN	ADCIFLAG	ADSEL
-		0	-	000	0	-	0	-	0	0	0000
-		RW	-	RW	RW	-	RW	-	RW	RW	RW

15	ADCEN	A/DC Module Enable bit. (The A/DC is automatically disabled at POWER-DOWN mode)	
		0	Disable A/DC module operation.
		1	Enable A/DC module operation.
13 11	TRIG	A/DC Trigger Signal Selection bits.	
		000	ADST.
		001	Timer 10 A-match signal.
		010	Timer 11 A-match signal.
		011	Timer 12 A-match signal.
		100	A/DC trigger signal from timer 30
		Others	Reserved
10	REFSEL	A/DC Reference Selection bit.	
		0	Select analog power. (VDD)
		1	Select external reference. (AVREF)
8	ADST	A/DC Conversion Start bit. This bit is automatically cleared to "0b" after operation.	
		0	No effect.
		1	Trigger signal generation for conversion start.
5	ADCIEN	A/DC Interrupt Flag bit.	
		0	Disable A/DC interrupt.
		1	Enable A/DC interrupt.
4	ADCIFLAG	A/DC Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred, This bit is cleared to '0' when write '1'.
3 0	ADSEL	A/D Converter Channel Selection bits.	
		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

18.2.2 ADC_DR: A/D converter data register

A/D Converter Data Register is 32-bit register. This Register is able to 32/16/8-bit access.

ADC_DR=0x4000_3004

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			ADDATA
-			0xXXX
-			RO

11 0	ADDATA	A/D Converter Result Data bits.

18.2.3 ADC_PREDR: A/D converter prescaler data register

A/D Converter Prescaler Data Register is 32-bit register. This Register is able to 32/16/8-bit access.

ADC_PREDR=0x4000_3008

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
-			0x0F
-			RW

5 0	PRED	A/D Converter Prescaler Data bits.

18.3 Functional description

18.3.1 Conversion timing

The A/D conversion process requires 2 steps to convert each bit and 30 clocks

the conversion rate is calculated as follows:

4 Clocks sample time + 26 Clocks Conversion Time = 30 clock

Conversion Time : MCLK * 30 clock = 4.5MHz*30=6.67us

(Maximum 4.5MHz conversion rate)

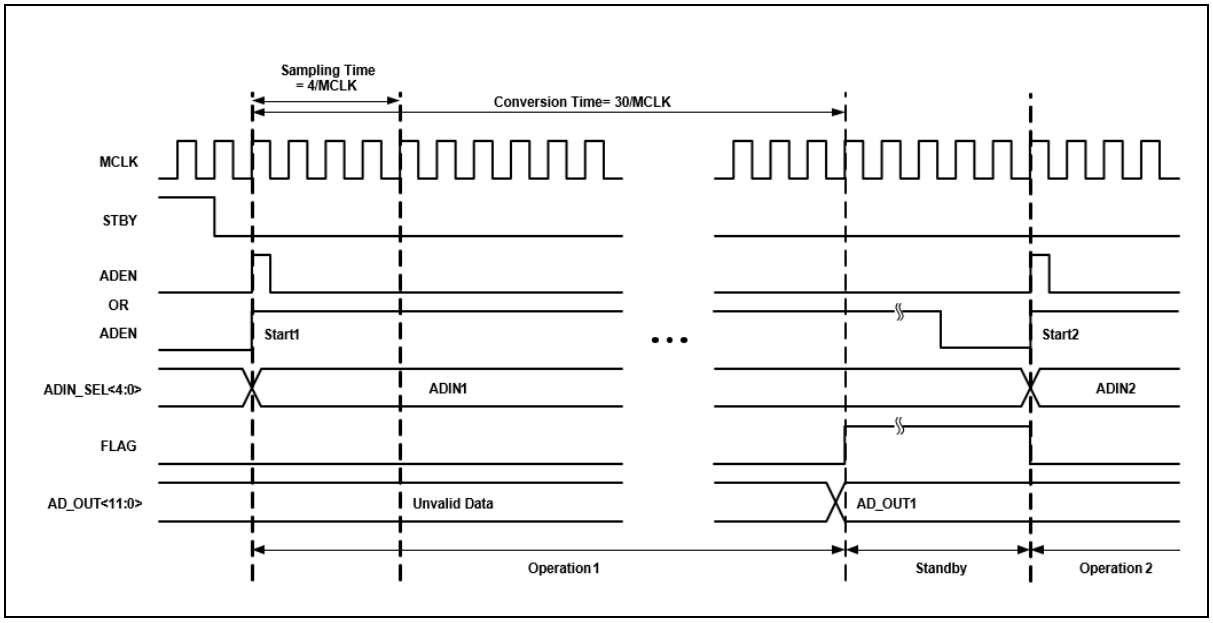


Figure 105. ADC Single Mode Timing (When ADC.MR.AMOD = '0')

19 5-bit DAC

Digital-to-analog (D/A) converter uses successive approximation logic to convert 5-bit digital value to an analog output level.

Table 70 introduces pins assigned for ADC.

Table 70. Pin Assignment of ADC: External Signal

Pin name	Type	Description
DAO	A	D/A converter Output

19.1 5-bit DAC block diagram

In this section, 5-bit DAC is described in a block diagram in Figure 106.

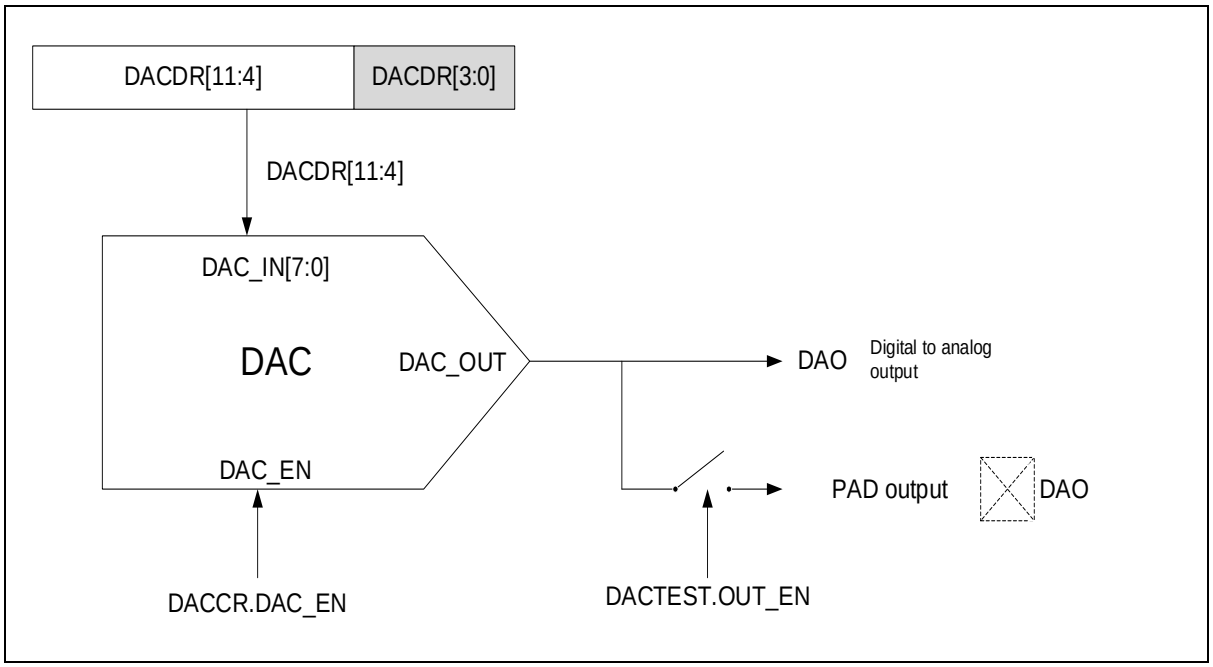


Figure 106. 5-bit DAC Block Diagram

19.2 Registers

Base address of DAC unit is introduced in the followings:

Table 71. Base Address of DAC

Name	Base address
DAC	0x4000_3450

Table 72. 10-bit DAC Register Map

Name	Offset	Type	Description	Reset value	Ref.
DAC_CR	0x00	RW	D/A Converter Control Register	0x0000_0000	19.2.1
DAC_DR	0x04	RW	D/A Converter Data Register	0x0000_0000	19.2.2
DAC_DACEN	0x08	RW	D/A Converter Port Control Register	0x0000_0000	19.2.3

19.2.1 DAC_CR: D/A converter control register

D/A Converter module should be configured properly before running. D/A Converter Control Register is 32-bit register. This Register is able to 32/16/8-bit access.

DAC_CR=0x4000_3450

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																															DACEN
-																															0
-																															RW

0	DACEN	D/A Module Enable bit	
		0	Disable D/AC module operation.
		1	Enable D/AC module operation.

19.2.2 DAC_DR: D/A converter data register

D/A Converter Data Register is 32-bit register. This Register is able to 32/16/8-bit access.

DAC_DR=0x4000_3454

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		DADATA	Reserved
-		0x00	-
-		RW	

11 4	DADATA[7:0]	D/A Converter Input Data bits.

19.2.3 DAC_DACEN: D/A converter PORT control register

D/A Converter Test Register is 32-bit register. This Register is able to 32/16/8-bit access.

DAC_OUTEN =0x4000_3458

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			DACPORT
-			0
-			RW

0	DACPORT	DAC output on PORT pin is disabled. DAC output on PORT pin is enabled.

20 Capacitive touch sensor system

Capacitive touch sensor systems are typical human machine interfaces (HMI) which operate by detecting changes in electrostatic capacitance produced by the touch of a finger or other conductor.

The use of capacitive touch technology can easily improve reliability in product design, and enhance the end-user experience. It also enables manufacturing costs to be lowered in a wide range of fields such as household appliances (white goods), healthcare devices, and other electric and electronic equipment.

The comparator features the followings:

- Self-Capacitive Touch Key Sensor.
- Total 24-channel Touch Key Support.
- 16-bits Sensing Resolutions.
- Fast Initial Self Calibration.
- Key Detection Mode: Single/Multi-Mode.
- The Improvement of the SNR by Bias-Calibration in Analog Sensing Block

Table 73 introduces pins assigned for TOUCH

Table 73. Pin Assignment of Touch: External Pins

Pin name	Type	Description
CS0 to CS23	IA	Capacitive touch switch input

20.1 Touch block diagram

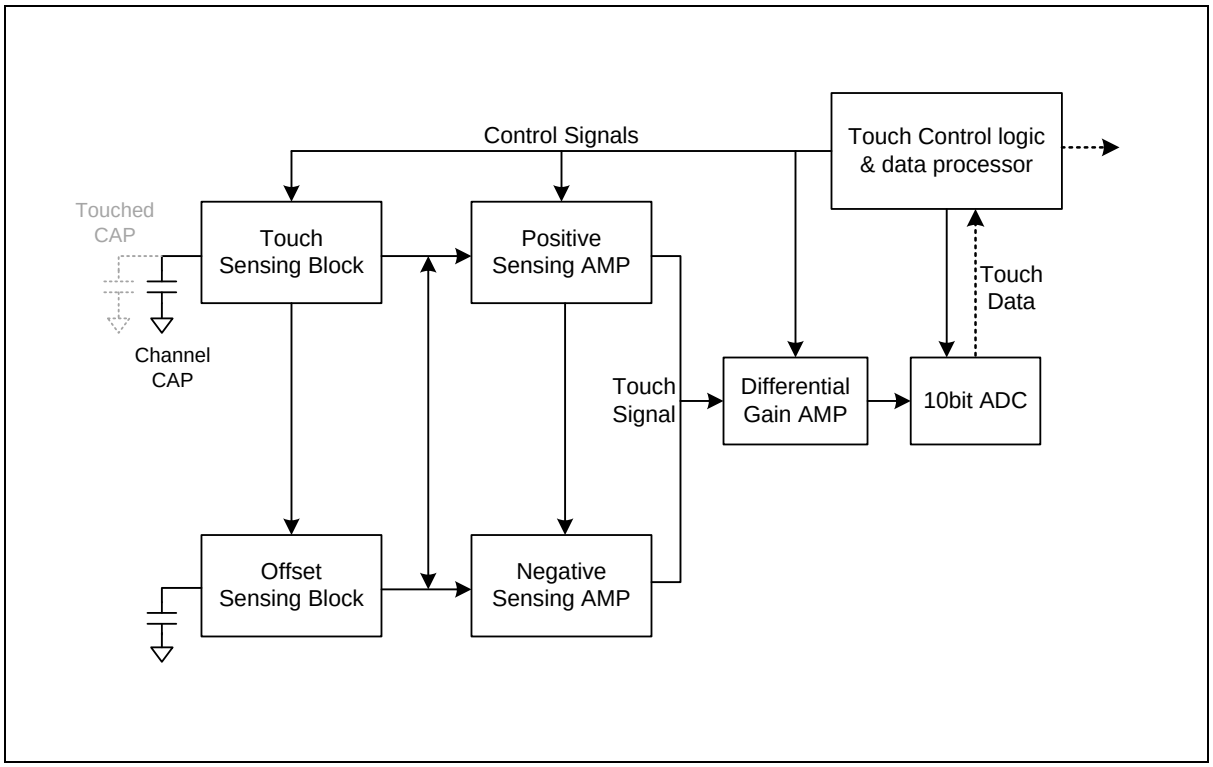


Figure 107. Touch Block Diagram

20.2 Registers

Base address of the comparator unit is introduced in the followings:

Table 74. Base Address of Touch

Name	Base address
TOUCH	0x4000_3600

Table 75. Comparator Register Map

Name	Offset	Type	Description	Reset value	Ref.
TSn_SUM_CH	0x000 -0x05C	RO	Touch Sensor Channel n(0~23) Sum Register	0x0000_0000	20.2.1
TSn_SCO	0x060 -0x0BC	RW	Touch Sensor Offset Capacitor Selection Register for Channel n(0~23)	0x0000_0000	20.2.2
TS_CON	0x100	RW	Touch Sensor Control Register	0x0000_0000	20.2.3
TS_MODE	0x104	RW	Touch Sensor Mode Register	0x0000_0010	20.2.4
TS_SUM_CNT	0x108	RW	Touch Sensor Sum Repeat Count Register	0x0000_0001	20.2.5
TS_CH_SEL	0x10C	RW	Touch Sensor Channel Selection Register	0x0000_0000	20.2.6
TS_SLP_CR	0x114	RW	Touch Sensor Low Pass Filter Control Register	0x0000_0001	20.2.7
TS_ADC_CH_SEL	0x118	RW	ADC Channel Selection Register	0x0000_0000	20.2.8
TS_INTEG_CNT	0x11C	RW	Touch Sensor Sensing Integration Count Register	0x0000_0032	20.2.9
TS_FREQ_NUM	0x120	RW	Touch Sensor Frequency Number Register	0x0000_00FF	20.2.10
TS_FREQ_DEL	0x124	RW	Touch Sensor Frequency Delta Register	0x0000_0000	20.2.11
TS_CLK_CFG	0x128	RW	Touch Sensor Clock Configuration Register	0x0000_0010	20.2.12
TS_TRIM_OSC	0x12C	RW	Touch Sensor RING Oscillator Trimming Selection Register	0x0000_00B8	20.2.13

Table 75. Comparator Register Map (continued)

Name	Offset	Type	Description	Reset value	Ref.
TS_TRIM_A_OSC	0x130	RW	Touch Sensor RING Oscillator Trimming for ADC Register	0x0000_00FF	20.2.14
TS_SCI	0x134	RW	Touch Sensor Input Capacitor Selection Register	0x0000_0034	20.2.15
TS_SCC	0x138	RW	Touch Sensor Conversion Capacitor Selection Register	0x0000_0004	20.2.16
TS_SVREF	0x13C	RW	Touch Sensor VREF Resistor Selection Register	0x0000_0004	20.2.17
TS_TAR	0x140	RW	Touch Sensor Integration AMP Reset Register	0x0000_0020	20.2.18
TS_TRST	0x144	RW	Touch Sensor Reset time of Sensing Register	0x0000_0003	20.2.19
TS_TDRV	0x148	RW	Touch Sensor Sample time of Sensing Register	0x0000_0003	20.2.20
TS_TINT	0x14C	RW	Touch Sensor Integration time of Sensing Register	0x0000_0014	20.2.21
TS_TD	0x150	RW	Touch Sensor Differential AMP Sampling Register	0x0000_0020	20.2.22
TS_TWR	0x154	RW	Touch Sensor Wait time Register	0x0000_0010	20.2.23
TS_TLED	0x158	RW	LED stable time Register	0x0000_0300	20.2.24

20.2.1 TSn_SUM_CH: touch sensor channel n sum register

Touch Sensor Channel n Sum or ADC Register (n = 0 to 23)

TS00_SUM_CH=0x4000_3600, TS01_SUM_CH=0x4000_3604, TS02_SUM_CH=0x4000_3608, TS03_SUM_CH=0x4000_360C
TS04_SUM_CH=0x4000_3610, TS05_SUM_CH=0x4000_3614, TS06_SUM_CH=0x4000_3618, TS07_SUM_CH=0x4000_361C
TS08_SUM_CH=0x4000_3620, TS09_SUM_CH=0x4000_3624, TS10_SUM_CH=0x4000_3628, TS11_SUM_CH=0x4000_362C
TS12_SUM_CH=0x4000_3630, TS13_SUM_CH=0x4000_3634, TS14_SUM_CH=0x4000_3638, TS15_SUM_CH=0x4000_363C
TS16_SUM_CH=0x4000_3640, TS17_SUM_CH=0x4000_3644, TS18_SUM_CH=0x4000_3648, TS19_SUM_CH=0x4000_364C
TS20_SUM_CH=0x4000_3650, TS21_SUM_CH=0x4000_3654, TS22_SUM_CH=0x4000_3658, TS23_SUM_CH=0x4000_365C

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		SUM_CH_DATA	
-		0x0000	
-		RO	

15 0	SUM_CH_DATA	Touch Sensor Channel n Sum or ADC Data

20.2.2 TSn_SCO: touch sensor offset capacitor selection register for TSn

Touch Sensor Offset Capacitor Selection Register for TSn (n = 0 to 23)

TS00_SCO=0x4000_3660, TS01_SCO=0x4000_3664, TS02_SCO=0x4000_3668, TS03_SCO=0x4000_366C
TS04_SCO=0x4000_3670, TS05_SCO=0x4000_3674, TS06_SCO=0x4000_3678, TS07_SCO=0x4000_367C
TS08_SCO=0x4000_3680, TS09_SCO=0x4000_3684, TS10_SCO=0x4000_3688, TS11_SCO=0x4000_368C
TS12_SCO=0x4000_3690, TS13_SCO=0x4000_3694, TS14_SCO=0x4000_3698, TS15_SCO=0x4000_369C
TS16_SCO=0x4000_36A0, TS17_SCO=0x4000_36A4, TS18_SCO=0x4000_36A8, TS19_SCO=0x4000_36AC
TS20_SCO=0x4000_36B0, TS21_SCO=0x4000_36B4, TS22_SCO=0x4000_36B8, TS23_SCO=0x4000_36BC

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			SCO
-			0x00
-			RW

8 0	SCO	Touch Sensor Offset Capacitor Selection	
		0_0000_0000b	0pF
		0_0000_0001b	0.1pF
		0_0000_0010b	0.2pF
		0_0000_0100b	0.4pF
		0_0000_1000b	0.8pF
		0_0001_0000b	1.6pF
		0_0010_0000b	3.2pF
		0_0100_0000b	6.4pF
		0_1000_0000b	12.8pF
		1_0000_0000b	25.6pF
		-	-
		1_1111_1111b	51.1pF

20.2.3 TS_CON: touch sensor control register

Touch Sensor Control Register. This register is 8-bit register.

TS_CON=0x4000_3700

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				OSC_EN	BGR_EN	TS_IF	Reserved	TS_RUN
-				0	0	0	-	0
-				RW	RW	RW	-	RW

4	OSC_EN	Oscillator Enable 0 Oscillator Disable (Default) 1 Oscillator Enable
3	BGR_EN	Band Gap Reference Enable 0 BGR Disable (Default) 1 BGR Enable
2	TS_IF	Touch Sensor Interrupt Flag 0 No new sensing results 1 In normal mode, this flag indicates that the new sensing results are generated. For next sensing, this flag must be cleared.
0	TS_RUN	Touch Sensor Enable 0 Touch Sensor Disable (Default) 1 Touch Sensor Enable

20.2.4 TS_MODE: touch sensor mode register

Touch Sensor Mode Register. This register is 8-bit register.

TS_MODE=0x4000_3704

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			SREF	SC_GAIN	SAP1	SAP0	Reserved	PORT1	PORT0
-			0	0	0	1	-	0	0
-			RW	RW	RW	RW	-	RW	RW

7	SREF	External Reference Offset Enable 0 Disable 1 Enable
6	SC_GAIN	Gain Calibration Capacitor Enable 0 Gain Calibration Capacitor Disable (Default) 1 Gain Calibration Capacitor Enable
5 4	SAP[1:0]	Touch Sensor Selection 01 Touch Sensor mode Select (Default) 10 ADC. The result of ADC is stored only at SUM_CH0 register
1 0	PORT[1:0]	Port Configuration During Inactive Status 00 Input Floating 01 Output Low 10 Output High

20.2.5 TS_SUM_CNT: touch sensor sum repeat count register

Touch Sensor Sum Repeat Count Register. This register is 8-bit register.

TS_SUM_CNT=0x4000_3708

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			TS_SUM_CNT
-			0x01
-			RW

7 0	TS_SUM_CNT	Touch Sensor Sum Repeat Count

20.2.6 TS_CH_SEL: touch sensor channel selection register

Touch Sensor Channel Selection Register

TS_CH_SEL=0x4000_370C

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	CH23_SEL	CH22_SEL	CH21_SEL	CH20_SEL	CH19_SEL	CH18_SEL	CH17_SEL	CH16_SEL	CH15_SEL	CH14_SEL	CH13_SEL	CH12_SEL	CH11_SEL	CH10_SEL	CH09_SEL	CH08_SEL	CH07_SEL	CH06_SEL	CH05_SEL	CH04_SEL	CH03_SEL	CH02_SEL	CH01_SEL	CH00_SEL
-	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

23 0	CH[23:0]_SEL	Touch Sensor Channel Selection Register	
		0	Disable (Default)
		1	Enable Touch Key

20.2.7 TS_SLP_CR: touch sensor low pass filter control register

Touch Sensor Low Pass Filter Control Register This register is 8-bit register.

TS_SLP_CR=0x4000_3714

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				SLP_C2	SLP_C1	SLP_C0	SLP_R3	SLP_R2	SLP_R1	SLP_R0
-				0	0	0	0	0	0	1
-				RW	RW	RW	RW	RW	RW	RW

6	SLP_C[2:0]	Capacitor Trimming for Input Low Pass Filter	
4		000	0pF
		001	4pF
		010	8pF
		011	12pF
		100	16pF
		101	20pF
		110	24pF
		111	28pF
3	SLP_R[3:0]	Resistor Trimming for Input Low Pass Filter	
0		0000	Channel open
		0001	0k
		0010	5k
		0100	10k
		1000	20k
		1110	2.8k
		0110	3.3k
		1010	4.0k
		1100	6.7k

20.2.8 TS_ADC_CH_SEL: ADC channel selection register

ADC Channel Selection Register

TS_ADC_CH_SEL=0x4000_3718

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	CH23_SEL	CH22_SEL	CH21_SEL	CH20_SEL	CH19_SEL	CH18_SEL	CH17_SEL	CH16_SEL	CH15_SEL	CH14_SEL	CH13_SEL	CH12_SEL	CH11_SEL	CH10_SEL	CH09_SEL	CH08_SEL	CH07_SEL	CH06_SEL	CH05_SEL	CH04_SEL	CH03_SEL	CH02_SEL	CH01_SEL	CH00_SEL
-	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

23	CH[23:0]_SEL	ADC Channel Selection
0		0 Disable (Default)
		1 Enable

20.2.9 TS_INTEG_CNT: touch sensor sensing integration count register

Touch Sensor Integration Count Register. This register is 8-bit register.

TS_INTEG_CNT=0x4000_371C

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			TS_INTEG_CNT
-			0x32
-			RW

7	TS_INTEG_CNT	Touch Sensor Sensing Integration Count
0		

20.2.10 TS_FREQ_NUM: touch sensor frequency number register

Touch Sensor Frequency Number Register. This register is 8-bit register.

TS_FREQ_NUM=0x4000_3720

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			TS_FREQ_NUM
-			0xFF
-			RW

7	TS_FREQ_NUM	Touch Sensor Frequency Number
0		
		NOTE: This register indicates the number of steps for frequency delta

20.2.11 TS_FREQ_DEL: touch sensor frequency delta register

Touch Sensor Frequency Delta Register. This register is 8-bit register.

TS_FREQ_DEL=0x4000_3724

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			TS_FREQ_DEL
-			0x00
-			RW

7 0	TS_FREQ_DEL	Touch Sensor Frequency Delta Register **NOTE:** This register indicates the frequency difference in every integration.

20.2.12 TS_CLK_CFG: touch sensor clock configuration register

Touch Sensor Clock Configuration Register. This register is 8-bit register.

TS_CLK_CFG=0x4000_3728

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			ACLKSEL	ACLKDIV2	ACLKDIV1	ACLKDIV0	TSCLKOE	TSCLKDIV2	TSCLKDIV1	TSCLKDIV0
-			0	0	0	1	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RW	RW

7	ACLKSEL	ADC Clock Source Select 0 Touch Sensor Clock 1 System MCU Clock
6 4	ACLKDIV[2:0]	ADC Clock Divider 000 OSC_{sys} / 1 (48MHz) 001 OSC_{sys} / 2 (24MHz) 010 OSC_{sys} / 4 (12MHz) 011 OSC_{sys} / 8 (6MHz) 100 OSC_{sys} / 16 101 OSC_{sys} / 32 110 OSC_{sys} / 64 111 OSC_{sys} / 128
3	TSCLKOE	Divided Touch Sensor Clock Output Enable 0 Clock Output Disable (Default) 1 Clock Output Enable
2 0	TSCLKDIV[2:0]	Touch Sensor Clock Divider 000 OSC_{ts} / 1 (27MHz) 001 OSC_{ts} / 2 010 OSC_{ts} / 4 011 OSC_{ts} / 8 100 OSC_{ts} / 16 101 OSC_{ts} / 32 110 OSC_{ts} / 64 111 OSC_{ts} / 128

After TS_RUN is cleared, please change the clock divider

20.2.13 TS_TRIM_OSC: touch sensor RING oscillator trimming selection register

Touch Sensor RING Oscillator Trimming Selection Register. This register is 8-bit register.

TS_TRIM_OSC=0x4000_372C

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			TRIM_OSC
-			0xB8
-			RW

7	TRIM_OSC	Touch Sensor RING Oscillator Trimming Selection
0		0x00 35MHz (maximum)
		0xB8 20MHz (default)
		0xFF 3.5MHz (minimum)

20.2.14 TS_TRIM_A_OSC: touch sensor RING oscillator trimming for ADC register

Touch Sensor RING Oscillator Trimming for ADC Register. This register is 8-bit register.

TS_TRIM_A_OSC=0x4000_3730

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			TRIM_A_OSC
-			0xFF
-			RW

7	TRIM_A_OSC	Touch Sensor RING Oscillator Trimming for ADC
0		0x00 35MHz (maximum)
		0xB8 29MHz (default)
		0xFF 3.5MHz (minimum)

20.2.15 TS_SCI: touch sensor clock configuration register

Touch Sensor Input Capacitor Selection Register. This register is 8-bit register.

TS_SCI=0x4000_3734

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			IBIAS_TRIM3	IBIAS_TRIM2	IBIAS_TRIM1	IBIAS_TRIM0	Reserved	SCI2	SCI1	SCI0
			0	0	1	1	-	1	0	0
			RW	RW	RW	RW	-	RW	RW	RW

7 4	IBIAS_TRIM[3:0]	BGR Current Bias Control
2 0	SCI[2:0]	Touch Sensor Input Capacitor Selection 000 1.2pF 001 2.4pF 010 3.6pF 011 4.8pF 100 6.0pF 101 7.2pF 110 8.4pF 111 9.6pF

20.2.16 TS_SCC: touch sensor conversion capacitor selection register

Touch Sensor Conversion Capacitor Selection Register. This register is 8-bit register.

TS_SCC=0x4000_3738

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				SCC2	SCC1	SCC0
-				1	0	0
-				RW	RW	RW

2 0	SCC[2:0]	Touch Sensor Conversion Capacitor Selection 000 2.4pF 001 4.8pF 010 7.2pF 011 9.6pF 100 12.0pF 101 14.4pF 110 16.8pF 111 19.2pF

20.2.17 TS_SVREF: touch sensor VREF resistor selection register

Touch Sensor VREF Resistor Selection Register. This register is 8-bit register.

TS_SVREF=0x4000_373C

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				SVREF3	SVREF2	SVREF1	SVREF0
-				0	1	0	0
-				RW	RW	RW	RW

3	SVREF[3:0]	Touch Sensor VREF Resistor Selection
0		0000 Open
		0001 2.5k
		0010 5k
		0100 10k
		1000 20k
		1111 1.3k

20.2.18 TS_TAR: touch sensor integration AMP reset register

Touch Sensor Integration AMP Reset Register. This register is 8-bit register.

TS_TAR=0x4000_3740

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			TAR
-			0x20
-			RW

7	TAR	Touch Sensor Integration AMP Reset Register
0		

20.2.19 TS_TRST: touch sensor reset time of sensing register

Touch Sensor Reset time of Sensing Register. This register is 8-bit register.

TS_TRSTP=0x4000_3744

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			TRST
-			0x03
-			RW

7	TRST	Touch Sensor Reset time of Sensing
0		

20.2.20 TS_TDRV: touch sensor driving time of sensing register

Touch Sensor Driving time of Sensing Register. This register is 8-bit register.

TS_TDRVP=0x4000_3748

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			TDRV
-			0x03
-			RW

7 0	TDRV	Touch Sensor Driving time of Sensing

20.2.21 TS_TINT: touch sensor integration time of sensing register

Touch Sensor Integration time of Sensing Register. This register is 8-bit register.

TS_TINTP=0x4000_374C

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			TINT
-			0x14
-			RW

7 0	TINT	Touch Sensor Integration time of Sensing

20.2.22 TS_TD: touch sensor differential AMP sampling register

Touch Sensor Differential AMP Sampling Register. This register is 8-bit register.

TS_TD=0x4000_3750

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			TD
-			0x20
-			RW

7 0	TD	Touch Sensor Differential AMP Sampling

20.2.23 TS_TWR: touch sensor wait time register

Touch Sensor Wait Time Register. This register is 8-bit register.

TS_TWR=0x4000_3754

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			TWR
-			0x10
-			RW

7 0	TWR	Touch Sensor Wait Time

20.2.24 TS_TLED: LED stable time register

LED stable Time Register. This register is 8-bit register.

TS_TLED=0x4000_3758

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			TLED
0X000003			0x00
-			RW

7 0	TLED	LED stable Time

21 LED driver

LED drive contains 16 COM X 10 SEG output pins. Its controller consists of display data RAM memory, COM and SEG generator.

The sixteen COMs and the ten SEGs can also be used as I / O pins. COMOE1, COMOE2 and SEGOE1, SEGOE2 registers are used to select one from SEG0 to SEG9 and COM0 to COM15.

During reset process when turning on the power, reset-input, low voltage or watchdog timer, LEDs are turned off.

The comparator features the followings:

- 16 COM X 10 SEG
- COM0 to COM15 and SEG0 to SEG9

Table 76 introduces pins assigned for ADC.

Table 76. Pin Assignment of LED: External Signal

Pin name	Type	Description
COM0 to COM15	O	LED common signal outputs
SEG0 to SEG9	O	LED segment signal outputs

21.1 LED block diagram

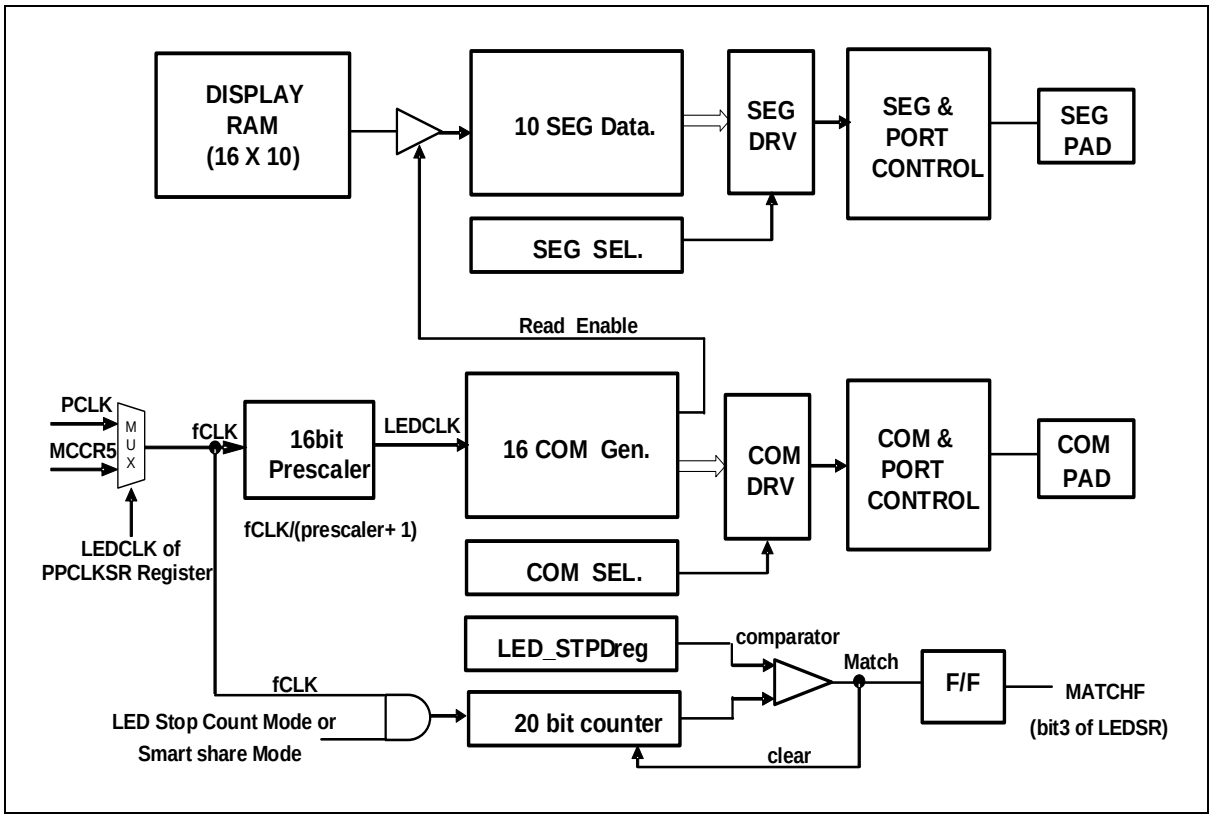


Figure 108. LED Block Diagram

21.2 Registers

Base address of the comparator unit is introduced in the followings:

Table 77. Base Address of Comparator

Name	Base address
LED	0x4000_6000

Table 78. Comparator Register Map

Name	Offset	Type	Description	Reset value	Ref.
LED_COMOE	0x00	RW	COM Ouput Enable Register	0x0000_0000	21.2.1
LED_SEGOE	0x04	RW	SEG Output Enable Register	0x0000_0000	21.2.2
LED_PRESD	0x08	RW	LED Prescaler Data Register	0x0000_0000	21.2.3
LED_COMER	0x0C	RW	COM Enable Register	0x0000_0000	21.2.4
LED_COMPWID	0x10	RW	COM Pulse Width Control Register	0x0000_0000	21.2.5
LED_DIMM1	0x14	RW	COM Dimming Control Register1	0x0000_0000	21.2.6
LED_DIMM2	0x18	RW	COM Dimming Control Register2	0x0000_0000	21.2.7
LED_DIMM3	0x1C	RW	COM Dimming Control Register3	0x0000_0000	21.2.8
LED_DIMM4	0x20	RW	COM Dimming Control Register4	0x0000_0000	21.2.9
LED_STPD	0x30	RW	LED STOP Duration Register	0x0000_0000	21.2.10
LED_SR	0x34	RW	LED STATUS Register	0x0000_0001	21.2.11
LED_CON2	0x38	RW	LED Control Register2	0x0000_0000	21.2.12
LED_CON1	0x3C	RW	LED Control Register1	0x0000_0000	21.2.13

21.2.1 LED_COMOE: COM output enable register

COM Output Enable Register is 27-bit register. This Register is able to 32/16/8-bit access.

LED_COMOE=0x4000_6000

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		COMOE2	COMOE1
-		0x00	0x00
-		RW	RW

15	COMOE2	Port Mode Select2	
8		00	Normal I/O PORT Select
		FF	COM[15:8] Select
7	COMOE1	Port Mode Select1	
0		00	Normal I/O PORT Select
		FF	COM[7:0] Select

21.2.2 LED_SEGOE: SEG output enable register

SEG Output enable Register is 11-bit register. This Register is able to 32/16/8-bit access.

LED_SEGOE=0x4000_6004

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		SEGOE2	SEGOE1
-		0	0x00
-		RW	RW

9	SEGOE2	Port Mode Select2	
8		0	Normal I/O PORT Select
		7	SEG[9:8] Select
7	SEGOE1	Port Mode Select1	
0		0	Normal I/O PORT Select
		FF	SEG[7:0] Select

21.2.3 LED_PRESD: LED prescaler data register

LED Prescaler Data Register is 16-bit register. This Register is able to 32/16/8-bit access.

LED_PRESD=0x4000_6008

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		PRESD2	PRESD1
-		0x00	0x00
-		RW	RW

15 0	PRESD	Pre-scale value of LED clock(PRESD = {PRESD2,PRESD1}) LED Clock = fCLK/(PRESD + 1) (fCLK is a selected input clock)

21.2.4 LED_COMER: COM enable register

COM Enable Register is 27-bit register. This Register is able to 32/16/8-bit access.

LED_COMER=0x4000_600C

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		COM15	COM14	COM13	COM12	COM11	COM10	COM9	COM8	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
-		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

15 0	LED_COMER[15:0]	Only Selected COM of COM0 and COM15 is active and displayed. **NOTE**: Only COM to be used, should be written to "1". Only One COM is allowed.

21.2.5 LED_COMPWID: COM pulse width control register

COM Pulse Width Control Register is 8-bit register. This Register is able to 32/16/8-bit access.

LED_COMPWID=0x4000_6010

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			COMPWID
-			0x00
-			RW

7 0	COMPWID	COM Pulse Width Control bits COM Width= LED CLK/(COMPWID + 1) (LED CLK is Prescaler output clock)

21.2.6 LED_DIMM1: COM dimming control register 1

LED_DIMM1 Dimming Control Register is 32-bit register. This Register is able to 32/16/8-bit access.

LED_DIMM1=0x4000_6014

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
COMDIMM3	COMDIMM2	COMDIMM1	COMDIMM0
0x00	0x00	0x00	0x00
RW	RW	RW	RW

31 24	COMDIMM3	COM3 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK* COMDIMM3
23 16	COMDIMM2	COM2 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM2
15 8	COMDIMM1	COM1 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM1
7 0	COMDIMM0	COM0 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM0
All COMDIMM should be less than (COMPWID to Overlaptime). If not, COM is not displayed.		

21.2.7 LED_DIMM2: COM dimming control register 2

LED_DIMM1 Dimming Control Register are 32-bit register. This Register is able to 32/16/8-bit access.

LED_DIMM2=0x4000_6018

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
COMDIMM7	COMDIMM6	COMDIMM5	COMDIMM4
0x00	0x00	0x00	0x00
RW	RW	RW	RW

31 24	COMDIMM7	COM7 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM7
23 16	COMDIMM6	COM6 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM6
15 8	COMDIMM5	COM5 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM5
7 0	COMDIMM4	COM4 Dimming Control bits 00~FF Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM4
All COMDIMM should be less than (COMPWID to Overlaptime). If not, COM is not displayed.		

21.2.8 LED_DIMM3: COM dimming control register 3

LED_DIMM2 Dimming Control Register is 32-bit register. This Register is able to 32/16/8-bit access.

LED_DIMM3=0x4000_601C

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
COMDIMM11	COMDIMM10	COMDIMM9	COMDIMM8
0x00	0x00	0x00	0x00
RW	RW	RW	RW

31 24	COMDIMM11	COM11 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM11
23 16	COMDIMM10	COM10 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM10
15 8	COMDIMM9	COM9 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM9
7 0	COMDIMM8	COM8 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM8
All COMDIMM should be less than (COMPWID – Overlaptime). If not, COM is not displayed.		

21.2.9 LED_DIMM4: COM dimming control register 4

LED_DIMM3 Dimming Control Register is 32-bit register. This Register is able to 32/16/8-bit access.

LED_DIMM4=0x4000_6020

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
COMDIMM15	COMDIMM14	COMDIMM13	COMDIMM12
0x00	0x00	0x00	0x00
RW	RW	RW	RW

31 24	COMDIMM15	COM15 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM15
23 16	COMDIMM14	COM14 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM14
15 8	COMDIMM13	COM13 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM13
7 0	COMDIMM12	COM12 Dimming Control bits
	00~FF	Dimmed COM Width = COMPWID - Overlaptime – LEDCLK*COMDIMM12
All COMDIMM should be less than (COMPWID – Overlaptime). If not, COM is not displayed.		

21.2.10 LED_STPD: LED stop duration register

LED STOP Duration Register is 20-bit register.

LED_STPD=0x4000_6030

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	LEDSTPD		
-	0x000000		
-	RW		

19 0	LEDSTPD	LED STOP Duration Register (since LED Start) LED STOP Count mode or Smart share mode 1) In LED STOP Count mode, this register is used as LED STOP duration value. The counter will automatically start to count after COM display - When the counter value is matched with this value, LED can re-start. It mean that touch key scan operation should be stop 2) In Smart Share mode, this register is used as LED STOP duration value. The counter will automatically start to count after COM display - Although the counter value is matched with this value LED can't re-start until pending touch key ends. 3)This register value is applicable only in LED STOP Count Mode or Smart Share modes -.LED STOP Duration Width= fCLK/(LEDSTPD + 1) (fCLK is selected mux output clock) For example If fCLK = 24MHZ and LEDSTPD =3a8c0h, STOP duration is about 10ms (24MHZ/(239,808 +1))

21.2.11 LED_SR: LED status register

LED STATUS Register is 4-bit register. This Register is able to 32/16/8-bit access.

LED_SR=0x4000_6034

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				MATCHF	LED_INT	LED_INTE	LED_ENDF
-				0	0	0	1
-				RW	RW	RW	RW

3	MATCHF	Flag to occur when LEDSTPD reg match with counter	
		0	Not Matched
		1	Matched
2	LED_INT	LED Interrupt Flag(in LED_INTE=1)	
		0	LED Interrupt not Generated
		1	LED Interrupt Generated
1	LED_INTE	LED Interrupt Enable	
		0	Disable
		1	Enable
0	LED_ENDF	LED Operation End Flag	
		1	Under LED no Operation
		0	Under LED Operation

NOTES:

1. MATCHF flag is clear as "0" after matched not only by write "0" but also by LED operation end
2. LED_INT isn't clear as "0" after interrupt generated as long as doesn't write "0"
3. Under Following Condition, LED_ENDF bit is set as "1"
   A. after reset release
   B. when write "1" LED_ENDF
   C. when LED operation end since LED start
   D. write "0" LEDEN(bit1 of LEDCON1) or LEDST(bit0 of LEDCON1)
4. .Under Following Condition, LED_ENDF bit is set as "0"
   A. when write "0" LED_ENDF
   B. when LED is operating since LED start

21.2.12 LED_CON2: LED control register2

LED Control Register2 is 6-bit register. This Register is able to 32/16/8-bit access.

LED_CON2=0x4000_6038

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				OVERLAP	OVERTS2	OVERTS1	OVERTS0
-				0	0	0	0
-				RW	RW	RW	RW

3	OVERLAP	OVERLAP TIME Select	
		0	Select (OVERLAP Time = fCLK/8 ~ fCLK/1024)
		1	Non-Select (OVERLAP Time = fCLK/2)
2	OVERTS	OVERLAP TIME Select	
1		000	fCLK/1024
0		001	fCLK/512
		010	fCLK/256
		011	fCLK/128
		100	fCLK/64
		101	fCLK/32
		110	fCLK/16
		111	fCLK/8
NOTE: Overlap time is recommended to be within range of 5us~60us.			

21.2.13 LED_CON1: LED control register1

LED Control Register1 is 4-bit register. This Register is able to 32/16/8-bit access.

LED_CON1=0x4000_603C

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				MD1	MD0	LEDEN	LEDST
				0	0	0	0
				RW	RW	RW	RW

3	MD[1:0]	Mode Select	
2		00	LED Alone Mode
		01	Hand Shake(with touch) Mode
		10	LED Stop Count Mode(since LED Start)
		11	Smart Share(with touch) Mode
1	LEDEN	LED Enable	
		0	LED Disable
		1	LED Enable
0	LEDST	LED START, STOP Operation	
		0	Stop LED Operation
		1	Start LED Operation

NOTES:

1. LED Operation is possible only after LEDEN(bit1 of LEDCON1 reg) is set "1"
2. Under Following Condition, LEDST bit is set as "1"
   A. when write "1" LEDST
   B. when TS_END from touch is generated in hand shake mode or smart share mode
   C. when LEDSTPD reg is matched with counter in LED Stop count Mode or smart share mode
3. .Under Following Condition, LEDST bit is clear as "0"
   A. when write "0" LEDEN
   B. when write "0" LEDST
   C. when LED operation is terminated
4. MD is changed to "00" only when write "0" LEDEN. Otherwise MD is holding previous written value.

21.3 Functional description

21.3.1 LED display RAM organization

Display data is stored in display data area of external data memory. The display data stored in the external display data area (address between 0x4000_6044H and 0x4000_6080H) is read automatically and sent to LED driver by hardware.

The LED driver generates SEG signals and COM signals in accordance with the display data and drive method. Display patterns can be changed only by memory overwrite of the display external data area.

Table 79 shows the correspondence between the external display data area and the COM/SEG pins. LED is turned on when the display data is "1", while it is turned off with "0" of the display data.

31 to 9 bits are not implemented.

Table 79. LED Display RAM

Name	Address	Bit						
		31	---	9	---	2	1	0
DISPRAM0	0x4000_6044	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM1	0x4000_6048	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM2	0x4000_604C	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM3	0x4000_6050	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM4	0x4000_6054	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM5	0x4000_6058	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM6	0x4000_605C	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM7	0x4000_6060	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM8	0x4000_6064	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM9	0x4000_6068	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM10	0x4000_606C	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM11	0x4000_6070	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM12	0x4000_6074	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM13	0x4000_6078	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM14	0x4000_607C	---	---	SEG9	---	SEG2	SEG1	SEG0
DISPRAM15	0x4000_6080	---	---	SEG9	---	SEG2	SEG1	SEG0

21.3.2 LED display mode function

Depending on how to set MD values (bit3, 2 of LED CONTROL REGISTER1), one of four modes can be selected for the use.

LED alone mode

First mode is LED alone mode which is unrelated to touch function. This can re-start by writing "1" in LEDST by programming.

All modes introduced below can be available only after LEDEN (bit1 of LEDCON1) is set to "1". Figure 109 shows an example of the LED alone mode with COMER reg = 000_0000_0000_0000_0000_0100_0001.

In Figure 109, (k) shows overlaptime between the COMs. The overlaptime can be controlled by programming. The larger the value of the COMDIMM is, the smaller the COM width in the direction of arrow is as shown in (j).

An interrupt occurs at point (A) where the LED operation is terminated when LED_INTE is set.

When enabling the LED by rewriting "1" to LEDST as shown in (B), LED starts again.

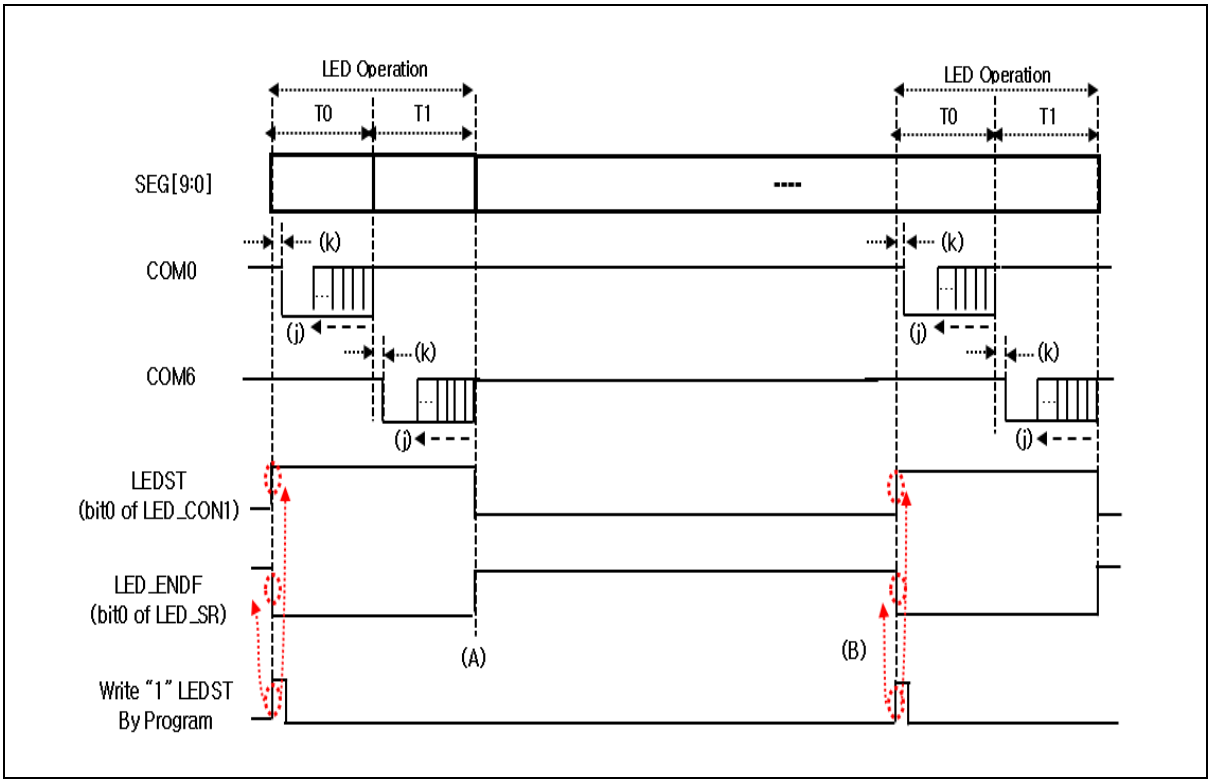


Figure 109. LED Alone Mode

Handshake mode

The second mode is Handshake mode. This informs the touch when the LED is finished and makes LED start again by receiving TS_END signal from the touch.

Figure 110 shows an example of the Handshake mode with COMER reg = 000_0000_0000_0000_0000_0000_0001.

As shown in Figure 110, frame cycle is defined by sum of LED operation (1) and touch key scan (2). It is the minimum cycle without flicker. An interrupt occurs at point (A) where the LED operation is terminated when LED_INTE is set. Touch continues to send "1" until LED_ENDF changes its value to "0" at (B). Then LED starts again.

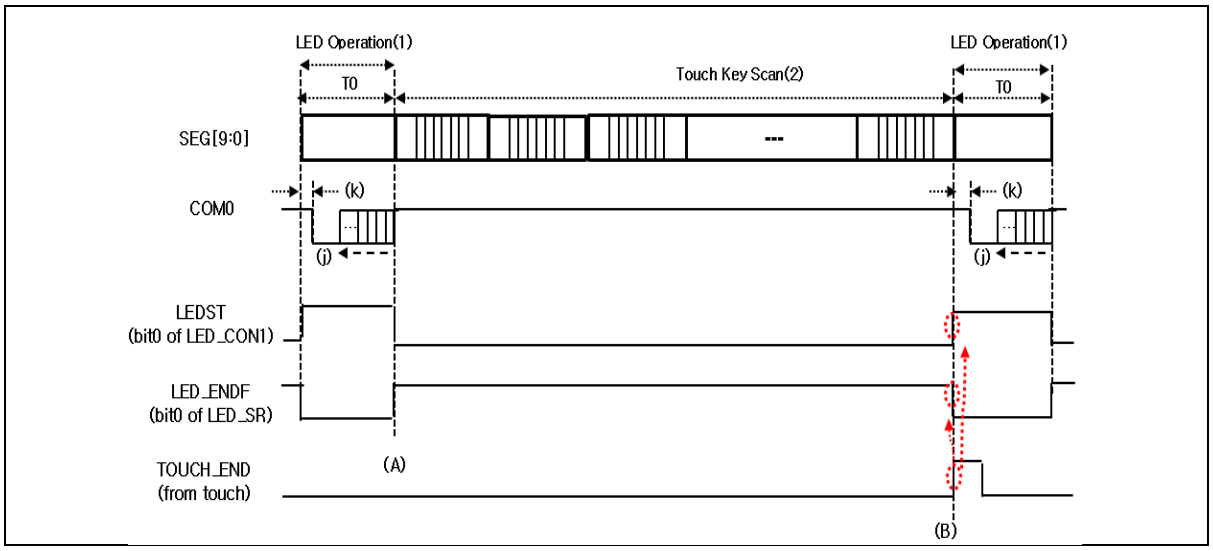


Figure 110. Handshake Mode

LED stop count mode

The third mode is LED stop count mode. Beginning of the next LED operation is determined by value of LED stop duration register when the previous LED operation ends. This is required when deciding the period of frame in which LED can operate without flickers.

Figure 111 shows an example of the LED stop count mode with COMER reg = 000_0000_0000_0000_0100_0000_0000.

As shown in Figure 111, an interrupt occurs at point (A) where the LED operation is terminated when LED_INTE is set. At the same time, 20-bit counter starts to increase by the internal enable bit. If the 20-bit sized LEDSTPD register is matched with the counter, MATCHF (bit3 of LED_SR) is set to "1", LED_ENDF is cleared to "0", 20-bit counter is cleared to "0", and LED starts again as shown at (B).

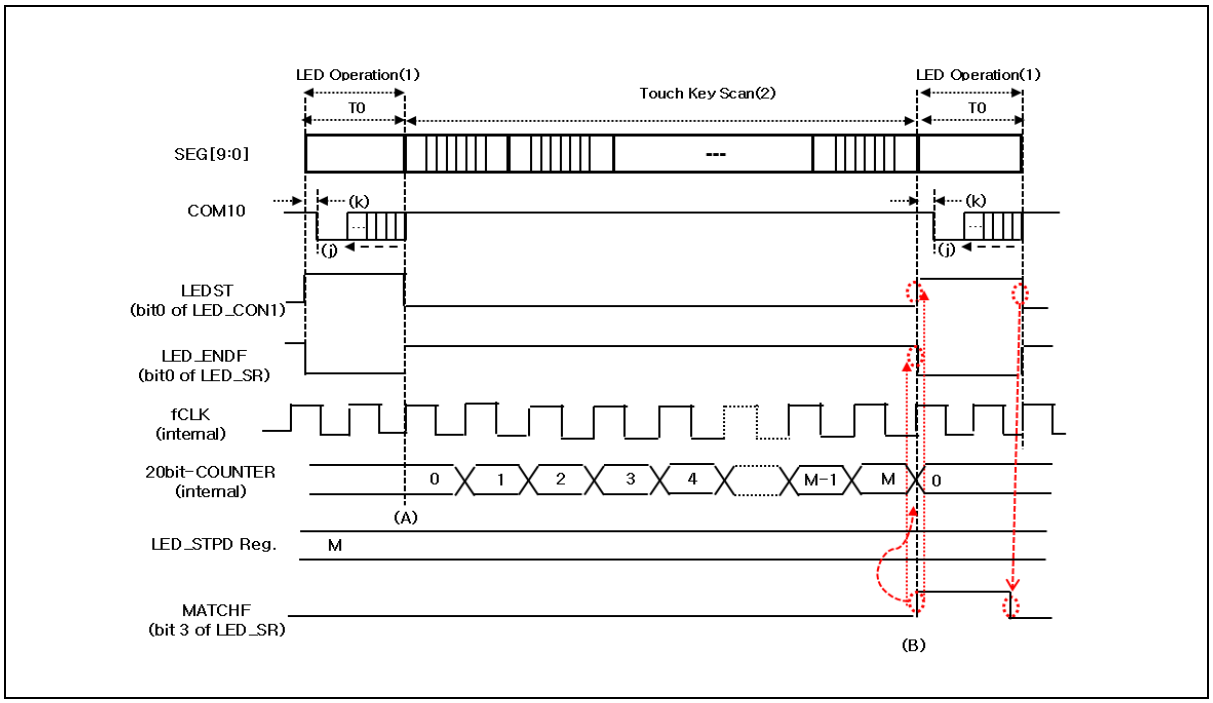


Figure 111. LED STOP Count Mode

Smart share mode

The fourth mode is Smart share mode. In this mode, the next LED operation can't start until current pending touch key scan ends although MATCHF (bit3 of LEDSR) is set after ending the previous LED operation. This mode is necessary when as many as touch key scan want to be executed during touch key scan time.

Figure 112 shows an example of the Smart share mode with COMER reg = 000_0000_0000_0000_0100_0000_0000.

As shown in Figure 112, an interrupt occurs at point (A) where the LED operation is terminated when LED_INTE is set. At the same time, 20-bit counter starts to increase by the internal enable bit although 20-bit sized LEDSTPD register is matched with the counter. Then MATCHF (bit3 of LEDSR) is set to "1".

LED_ENDF can't clear to "0" and LED can't start again until the TS_END signal from the touch becomes active. If LED display is not necessary any longer, it is highly recommended to set LEDEN (bit1 of LED_CON1) to "0" to execute between the end of LED operation (A) and LED restart (B) to prevent flickers.

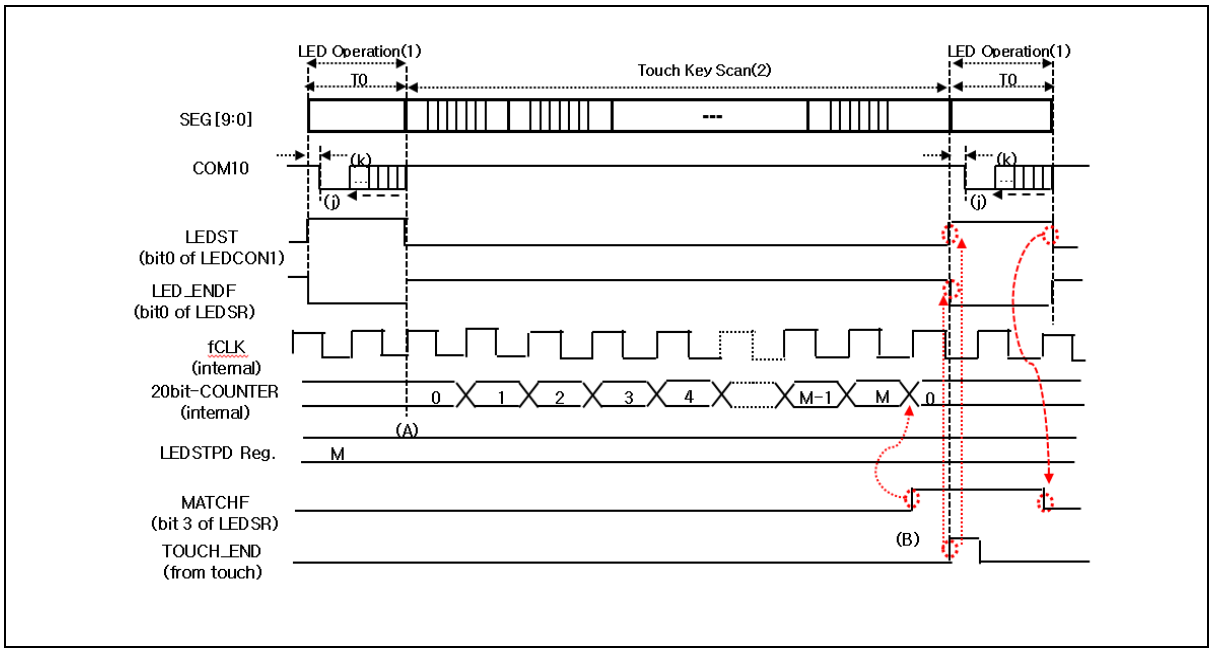


Figure 112. Smart Share Mode

22 Cyclic redundancy check and checksum (CRC checksum)

Cyclic redundancy check (CRC) generator is used to get a 16-bit CRC code from Flash ROM and a generator polynomial. Among other applications, CRC-based techniques are used to verify data transmission or storage integrity. In the scope of the functional safety standards, they offer a means of verifying the Flash memory integrity. CRC generator helps compute a signature of the software during runtime, to be compared with a reference signature.

CRC generator of A31G21x series features the followings:

- Auto CRC (DMA) and User CRC Mode.
- Polynomial:
  - CRC-CCITT ($G_1(x) = x^{16} + x^{12} + x^5 + 1$)
  - CRC-16 ($G_2(x) = x^{16} + x^{15} + x^2 + 1$)
- CRC Mode and Checksum Mode.

22.1 CRC and checksum block diagram

Figure 113 describes the CRC and checksum in a block diagram.

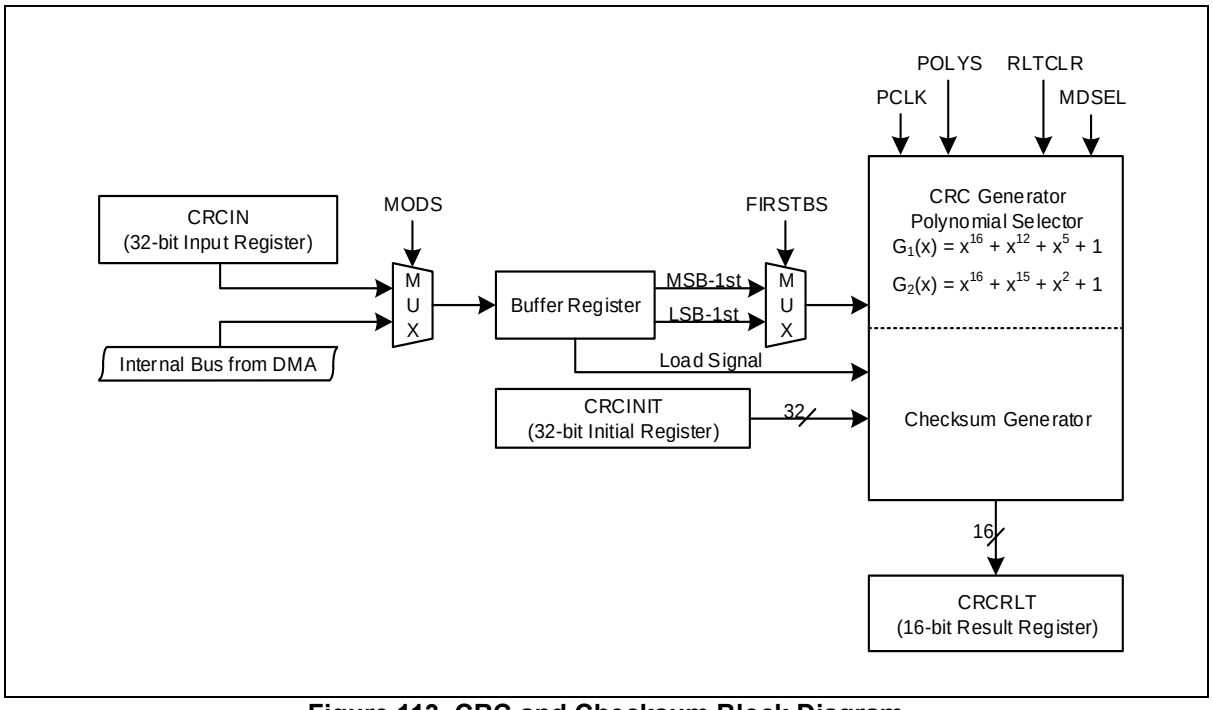


Figure 113. CRC and Checksum Block Diagram

22.2 Registers

Base address of the CRC and checksum block is introduced in the followings:

Table 80. Base Address of CRC

Name	Base address
CRC	0x4000_0300

Table 81. CRC Register Map

Name	Offset	Type	Description	Reset value	Ref.
CRC_CR	0x00	RW	CRC/Checksum Control Register	0x0000_0000	22.2.1
CRC_IN	0x04	RW	CRC/Checksum Input Data Register	0x0000_0000	22.2.2
CRC_RLT	0x08	RO	CRC/Checksum Result Data Register	0x0000_FFFF	22.2.3
CRC_INIT	0x0C	RW	CRC/Checksum Initial Data Register	0x0000_0000	22.2.4

22.2.1 CRC_CR: CRC control register

CRC/Checksum Control Register is a 32-bit register. This Register is able to 32/16/8-bit access.

CRC_CR=0x4000_0300

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			CRCINTEN	CRCINTF	MODS	RLTCLR	MDSEL	POLYS	Reserved	FIRSTBS	CRCRUN
-			0	0	0	0	0	0	-	0	0
-			RW	RW	RW	RW	RW	RW	-	RW	RW

9	CRCINTEN	CRC interrupt enable bit
		0 Disable
		1 Enable
8	CRCINTF	CRC interrupt flag bit
		0 No request occurred
		1 Request occurred, This bit is cleared to '0' when write '1'.
7	MODS	User/DMA Mode Selection bit.
		0 User mode. (Calculation every writing data to the CRC_IN register)
		1 Auto mode. (Calculation till CRCSADR == CRCEADR)
6	RLTCLR	CRC/Checksum Result Data Register (CRC_RLT) Initialization bit.
		0 No effect.
		1 Initialize the CRC_RLT register with the value of CRCINIT (This bit is automatically cleared to "0b" after operation)
5	MDSEL	CRC/Checksum Selection bit.
		0 Select CRC.
		1 Select checksum.
4	POLYS	Polynomial Selection bit. (CRC only)
		0 Select CRC-CCITT ($G_1(x) = x^{16} + x^{12} + x^5 + 1$)
		1 Select CRC-16 ($G_2(x) = x^{16} + x^{15} + x^2 + 1$)
1	FIRSTBS	First Shifted-in Selection bit. (CRC only)
		0 MSB-1st.
		1 LSB-1st.
0	CRCRUN	CRC/Checksum Start Control and Busy bit.
		0 Not busy. The CRC operation can be finished by writing "0b" to this bit on running.
		1 Start CRC operation.
		NOTE: The 4 "NOP instruction" should be executed immediately after this bit is set to "1b".

NOTES:

1. The CRC_RLT register and the CRC/Checksum block should be initialized by writing "1b" to the RLTCLR bit before a new CRC/Checksum calculation.
2. The CRCRUN bit should be set to "1b" 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. It is prohibited writing any data to the CRC_IN register during CRCRUN=0.
5. The checksum is calculated by byte unit. Ex) On 0x34A7E991, CRC_RLT = 0x34 + 0xA7 + 0xE9 + 0x91.
6. The 4 "NOP Instruction" should follow immediately after CRCRUN bit is set to "1b".

22.2.2 CRC_IN: CRC input data register

CRC Input Data Register is 32-bit register.

CRC_IN=0x4000_0304

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			
0x00000000			
RW			

31 0	INDATA	CRC Input Data bits.

NOTE: The CRC_IN register should be written by 1-word (32-bits).

22.2.3 CRC_RLT: CRC result data register

CRC Result Data Register is 32-bit register. This Register is able to 32/16/8-bit access.

CRC_RLT=0x4000_0308

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	
-		0XFFFF	
-		RO	

15 0	RLTDATA	CRC Result Data bits.

22.2.4 CRC_INIT: CRC initial data register

CRC Initial Data Register is 32-bit register. This Register is able to 32/16/8-bit access.

CRC_INIT=0x4000_030C

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		INIDATA	
-		0x0000	
-		RW	

15 0	INIDATA	CRC Initial Data bits.

22.3 Functional description

22.3.1 CRC polynomial structure

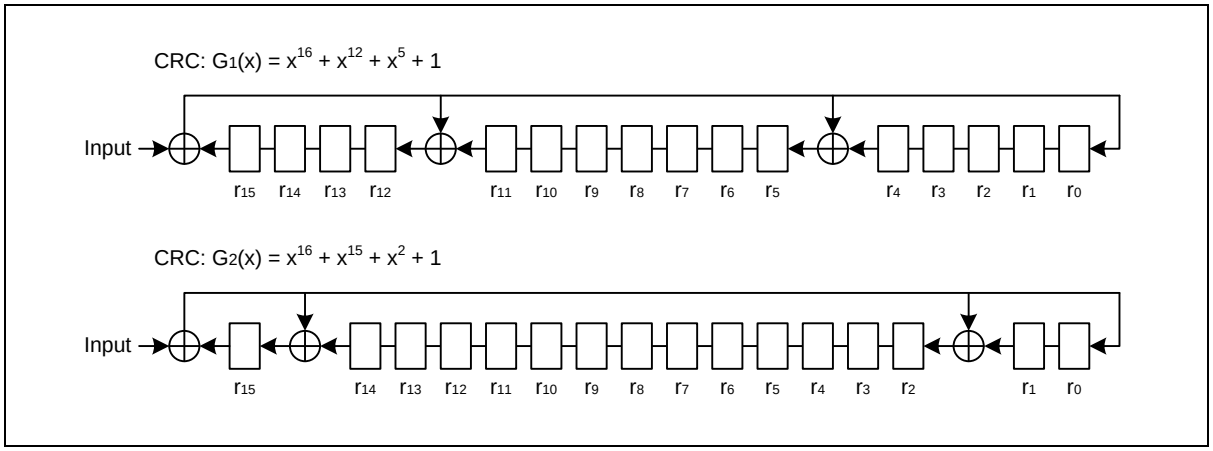


Figure 114. CRC Polynomial Structure

22.3.2 CRC operation procedure in DMA mode

1. CRC/Checksum Enable and Clock Enable. (SCU_PCER, SCU_PER)
2. Set CRC initial data register. (CRC_INIT)
3. Set CRC control register.
4. CRC operation enable (CRCRUN = 1)
5. DMA configuration & operation
6. Read the CRC result.

22.3.3 CRC operation procedure in user CRC and checksum mode

1. CRC/Checksum Enable and Clock Enable (SCU_PCER, SCU_PER)
2. Set CRC initial data register. (CRC_INIT)
3. Set CRC control register.
4. CRC operation enable. (CRC_RUN = 1)
5. Input CRC Data at CRC_IN.
6. CRC stop and read CRC result.

23 Electrical characteristics

23.1 Absolute maximum ratings

Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions.

Table 82. Absolute Maximum Rating

Parameter	Symbol	Ratings	Unit	Remark
Supply Voltage	VDD	-0.3 – +6.5	V	—
Normal Pin	V_I	-0.3 – VDD+0.3	V	Voltage on any pin with respect to VSS
	V_O	-0.3 – VDD+0.3	V	
	I_{OH}	-20	mA	Maximum current output sourced by (I_{OH} per I/O pin)
	ΣI_{OH}	-100	mA	Maximum current (ΣI_{OH})
	I_{OL}	25	mA	Maximum current sunk by (I_{OL} per I/O pin)
	ΣI_{OL}	210	mA	Maximum current (ΣI_{OL})
Total Power Dissipation	T_P	300	mW	—
Storage Temperature	T_{STG}	-45 – +125	℃	—

23.2 Recommended operating conditions

Table 83. Recommended Operating Condition

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Supply Voltage	VDD	Logic block	1.8	—	5.5	V
		Touch	2.7	—	5.5	V
		LED	3.3		5.5	
Operating Frequency	FREQ	HSE	2	—	16	MHz
		LSE	—	32.768	—	KHz
		HSI	31.52	32	32.48	MHz
		LSI500KHz	400	500	600	KHz
		LSI40KHz	22.8	40	62.6	KHz
Operating Temperature	Top	VDD = 1.8 – 5.5V (Commercial grade)	-40	—	+85	℃
		VDD = 1.8 – 5.5V (Industrial grade)	-40		+105	℃

23.3 ADC characteristics

Table 84. ADC Electrical Characteristics

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating voltage	AVDD		2.4	5	5.5	V
Resolution				12		Bit
Operating current	IDDA	AVDD = 5.0VA		1	2	mA
Analog input range	V_{AN}		VSS		AVREF	V
Conversion rate	F_{CONV}			—	150	Ksps
Operating frequency	ACLK				4.5	MHz
Integral Non-Linearity	INL	AVDD=2.4V < AVDD < 5.5V,T_A = 25 °C		±4	±10	LSB
Differential Non-Linearity	DNL			±1	±4	LSB
Top Offset Error(FSE)	TOE			±6	±12	LSB
Zero Offset Error	ZOE			±4	±8	LSB

23.4 Power-on reset characteristics

Table 85. POR Electrical Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating Current	IDD	—	-	60	-	uA
POR set level	V_{set}	—	1.05	1.2	1.35	V
POR reset level	V_{reset}	—	1.0	1.1	1.2	V

23.5 Low voltage reset/indicator characteristics

Table 86. Low Voltage Reset/Indicator Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Detection Level	V_{LVR}	Falling Voltage (Error rate 5%)		1.60		V
				1.69		
				1.78		
				1.90		
				1.99		
				2.12		
				2.30		
				2.47		
				2.67		
				3.04		
				3.18		
				3.59		
				3.72		
				4.03		
				4.20		
				4.48		
Hysteresis	-		-	100	200	mV
Noise cancelling time	-		-	2	-	us
Operation Current	I_{DD}		-	3.5	5	uA
Operation Current (STOP)	$I_{DD, STOP}$		-	2.5	3	nA

23.6 High frequency internal RC oscillator characteristics

Table 87. High Frequency Internal RC Oscillator Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Frequency	f_{HSI}	VDD = 1.8V to 5.5V	31.52	32	32.48	MHz
Tolerance		T_A = – 40 °C to + 85 °C (Commercial grade)	–	–	±1.0	%
		T_A = – 40 °C to + 105 °C (Industrial grade)	–	–	±1.5	
Clock Duty Ratio	T_{OD}	–	–	50	–	%
Stabilization Time	t_{HFS}	–	–	–	–	us
IRC Current	I_{HSI}	Enable	–	190	–	uA
		Disable	–	1	--	uA

23.7 Low frequency internal RC oscillator characteristics

Table 88. Low Frequency (500KHz) Internal RC Oscillator Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating voltage	VDD		1.8	5	5.5	V
Operating current	I_{LIRC}	Enable	—	1.5	2	uA
		Disable	—	1	20	nA
Frequency	f_{LIRC}	VDD = 1.8V to 5.5V	400	500	600	KHz
Stabilization time	t_{LFS}	—	—	100		us

23.8 Touch switch characteristics

Table 89. Touch Switch Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating Voltage	V_{DD}	–	2.7	-	5.5	V
	V_{DDA}	–	2.7	-	5.5	V
VDC Voltage	V_{CCL}	From MCU	–	1.9	–	V
SNR (Signal-to-Noise Ratio)	SNR	–	–	20	–	dB
Self-Calibration Time	T_{CAL}	–	–	10	–	ms
Scan Speed	T_{SCAN}	–	–	10	–	ms
Supply Current	I_{DD}	–	–	1	–	mA
Operation Temperature	T_{OPER}	VDD = 2.7 – 5.5V (Commercial grade)	-40	–	+85	℃
		VDD = 2.7 – 5.5 V(Industrial grade)	-40	–	+105	℃

23.9 LED driver characteristics

Table 90. LED Driver Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Operating Voltage	V_{Ddext}		3.3	–	5.5	V
Operating Temperature	T_A	Commercial grade	-40	–	85	℃
		Industrial grade	-40	–	105	℃
COM output leakage	I_{CLKG}		-1		1	uA
SEG output leakage	I_{SLKG}		-1		1	uA
SEG Current	I_{SEG}	V_{Ddext} = 3.3V ,V_{OL_LED} = 0.3V	23	–	–	mA

23.10 DC electrical characteristics

Table 91. DC Electrical Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Input High Voltage	V_{IH1}	All input pins, nRESET	0.8VDD	-	VDD	V
	V_{IH2}	PF5,PF6,PF7 are input 1.8V level	0.9	-	VDD	
Input Low Voltage	V_{IL1}	All input pins, nRESET	0	-	0.2VDD	V
	V_{IL2}	PF5,PF6,PF7 are input 1.8V level	0	-	0.6	
Output High Voltage	V_{OH1}	VDD=5V, I_{OH1} = – 2.36mA	0.8VDD	-	VDD	V
Output Low Voltage	V_{OL1}	VDD=5V, I_{OL1} = 4.86mA	0	-	0.2VDD	V
Input high leakage current	I_{IH}	All Input ports	-4	-	-	uA
Input low leakage current	I_{IL}	All Input ports	-	-	+4	uA
Pull-up resistor	R_{PU}	V_I=0V, T_A=25°C, All Input ports	40	-	70	KΩ
		V_I=0V, T_A=25°C, nRESET PIN		250		
Pull-down resistor	R_{PD}	V_I=VDD, T_A=25°C, All Input ports	40	-	70	KΩ
OSC feedback resistor	R_{X1}	XIN=VDD, XOUT=VSS, T_A=25°C, VDD=5V		1		MΩ

23.11 Supply current characteristics

Table 92. Supply Current Characteristics

Parameter	Symbol	Conditions		Typ	Max	Units
Supply Current	I_{DD1} (Run)	f_{XIN} = 8MHz	VDD=5V±10%	4.0	12.0	mA
		f_{HSI} = 32MHz		10.0	30.0	
		f_{HSI} = 8MHz		3.5	10.0	
		f_{LSI} = 500KHz		200	600	uA
		F_{LSE} = 32.768KHz		140	300	
	I_{DD2} (Sleep)	f_{XIN} = 8MHz		5	15	mA
		f_{HSI} = 32MHz		6	18	
		f_{HSI} = 8MHz		2	6	
		f_{LSI} = 500KHz		180	500	uA
		F_{LSE} = 32.768KHz		130	400	
	I_{DD3} (Deep Sleep)	WDT block(WDTRC) ON, LVD ON T_A = 25 °C		8		uA
	I_{DD4} (Deep Sleep)	WDT block(WDTRC) ON, LVD OFF T_A = 25 °C		6		
	I_{DD5} (Deep Sleep)	WDT block(WDTRC) OFF, LVD ON T_A = 25 °C		4		uA
	I_{DD6} (Deep Sleep)	WDT block(WDTRC) OFF, LVD OFF T_A = 25 °C		2		
		WDT block(WDTRC) OFF, LVD OFF T_A = 85 °C		10		
		WDT block(WDTRC) OFF, LVD OFF T_A = 105 °C		30		

NOTES:

1. All supply current items don't include the current of a low frequency internal RC oscillator and a peripheral block.
2. All supply current items include the current of the power-on reset (POR) block.

23.12 AC characteristics

Table 93. AC Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
RESETB input low width	t_{RST}	VDD = 5V	10	—	—	us
Interrupt input high low width	t_{IWH}, t_{IWL}	All interrupts, VDD = 5V	100	—	—	ns
External counter input high low pulse width	t_{ECWH}, t_{ECWL}	VDD = 5V All external counter input	100	—	—	
External counter transition time	t_{REC}, t_{FEC}	Ecn, VDD = 5V All external counter input	—	—	20	

NOTE: Data based on characterization results, not tested in production.

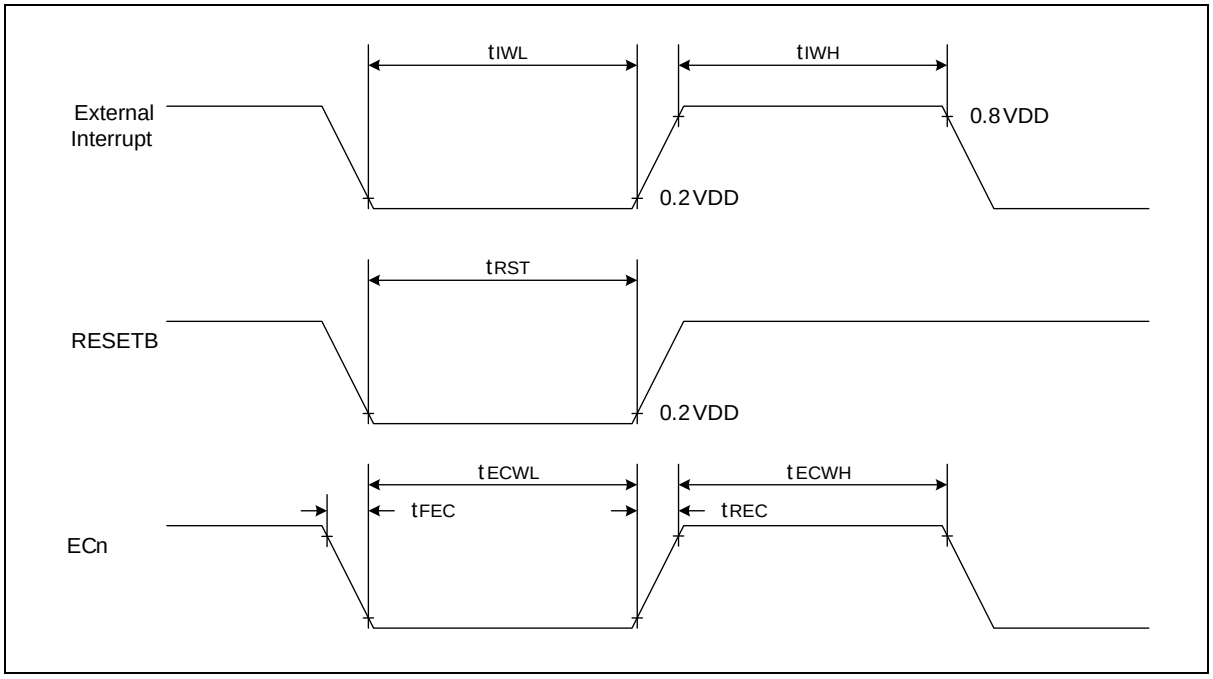


Figure 115. AC Timing

23.13 SPI characteristics

Table 94. SPI Characteristics (2.7 -5.5V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Output Clock Pulse Period	t_{SCK}	Internal SCK source	400	–	–	ns
Input Clock Pulse Period		External SCK source	400	–	–	
Output Clock High, Low Pulse Width	t_{SCKH}, t_{SCKL}	Internal SCK source	180	–	–	
Input Clock High, Low Pulse Width		External SCK source	180	–	–	
First Output Clock Delay Time	t_{FOD}	Internal/External SCK source	200	–	–	
Output Clock Delay Time	t_{DS}	–	–	–	100	
Input Setup Time	t_{DIS}	–	180	–	–	
Input Hold Time	t_{DIH}	–	180	–	–	

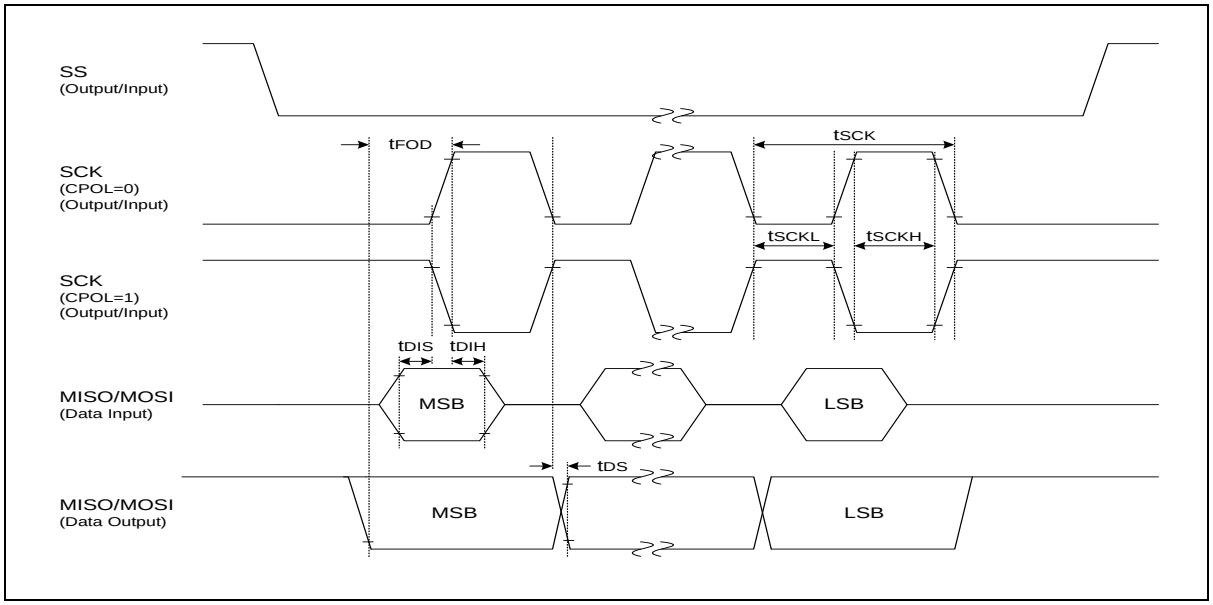


Figure 116. SPI Timing

23.14 I2C characteristics

Table 95. I2C Characteristics

(VDD = 1.8 to 5.5V)

Parameter	Symbol	Min	Max	Units
Clock frequency	t_{SCL}	0	400	KHz
Clock high pulse width	t_{SCLH}	0.6	—	us
clock low pulse width	t_{SCLL}	1.3	—	
Bus free time	t_{BF}	1.3	—	
Start condition setup time	t_{STSU}	0.6	—	
Start condition hold time	t_{STHD}	0.6	—	
Stop condition setup time	t_{SPSU}	0.6	—	
Stop condition hold time	t_{SPHD}	0.6	—	
Output valid from clock	t_{VD}	0	—	
Data input hold time	t_{DIH}	0	1.0	
Data input setup time	t_{DIS}	100	—	ns

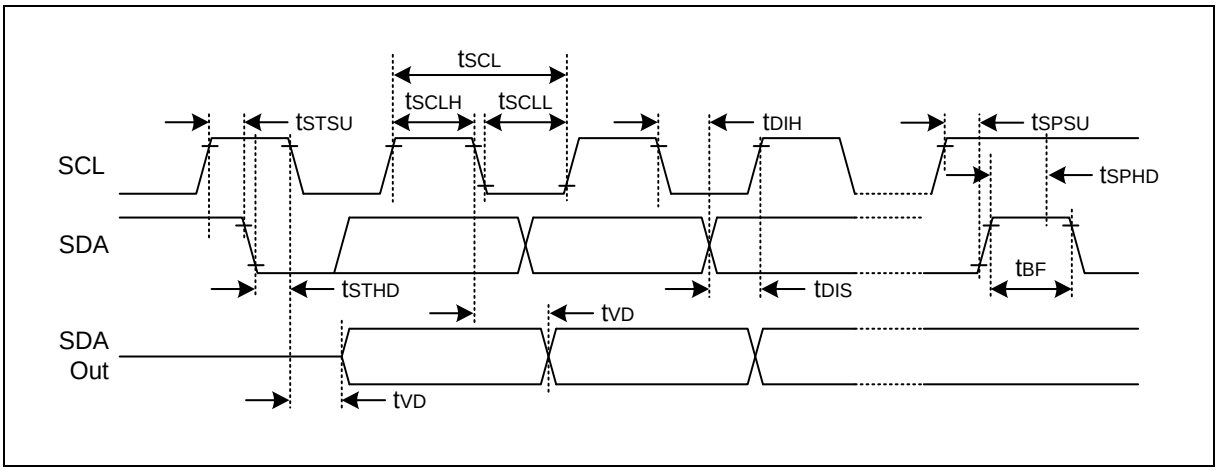


Figure 117. I2C Timing

23.15 USART UART timing characteristics

Table 96. UART Timing Characteristics

Parameter	Symbol	Typ	Units
Serial port clock cycle time	tSCK	t_{CPU} x 16	us

NOTES:

1. tCPU : BDR * 2 * (1/PCLK)

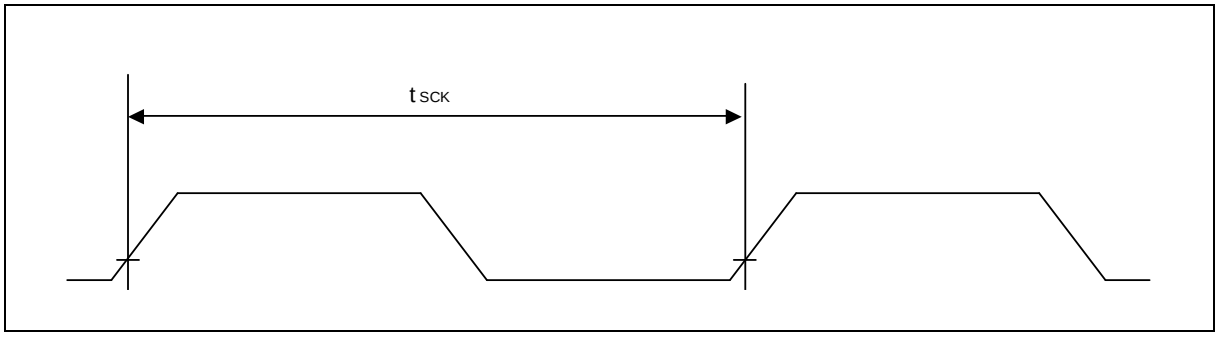


Figure 118. Waveform of UART Timing Characteristics

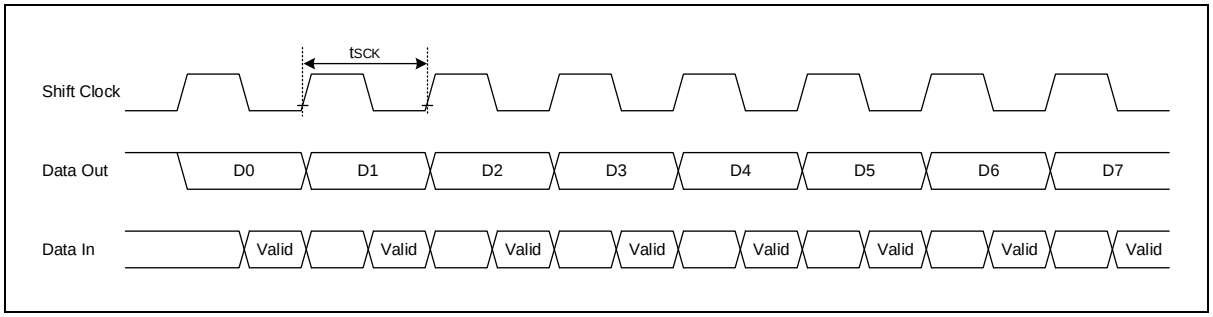


Figure 119. UART Module Timing

NOTES:

1. tSCK : Board Rate

23.16 Data retention voltage in STOP mode

Table 97. Data Retention Voltage in Stop mode

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Data retention supply voltage	V_{DDDR}	—	1.8	—	5.5	V
Data retention supply current	I_{DDDR}	V_{DDDR} = 1.8V, (T_A = 25°C), Deep sleep mode	—	—	1	uA

23.17 Internal flash ROM characteristics

Table 98. Internal Flash ROM Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Reset Cycle Time	fRSTBUSY	–	5.6	8	10.4	us
Fuse Program Cycle Time	fFRDBUSY		4.2	6	7.8	
Normal Program Cycle Time	tPGMBUSY		21	30	42	
Normal Page Erase Cycle Time	tPERSBUSY		2.8	4	5.2	ms
Sector Erase Cycle Time	tSERSBUSY		2.8	4	5.2	
Chip Erase Cycle Time	tMERSBUSY	–	5.6	8	10.4	
Flash Program Voltage	V_{PGM}	On erase/write	1.8	–	5.5	V
Endurance of Write/Erase	N_{FWE}	T_A=25 °C, Page unit	10,000	–	–	Times
Retention Time	t_{FRT}		10	–	–	Years

23.18 Input/output capacitance

Table 99. Input/ Output Capacitance

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Input Capacitance	C_{IN}	f=500KHz Unmeasured pins are connected VSS	–	–	10	pF
Output Capacitance	C_{OUT}					
I/O Capacitance	C_{IO}					

23.19 Main oscillator characteristics

Table 100. Main Oscillator Characteristics

Oscillator	Parameter	Conditions	Min	Typ	Max	Units
Crystal	Main oscillation frequency	1.8 V – 5.5 V	2.0	–	16.0	MHz
Ceramic Oscillator	Main oscillation frequency	1.8 V – 5.5 V	2.0	–	16.0	

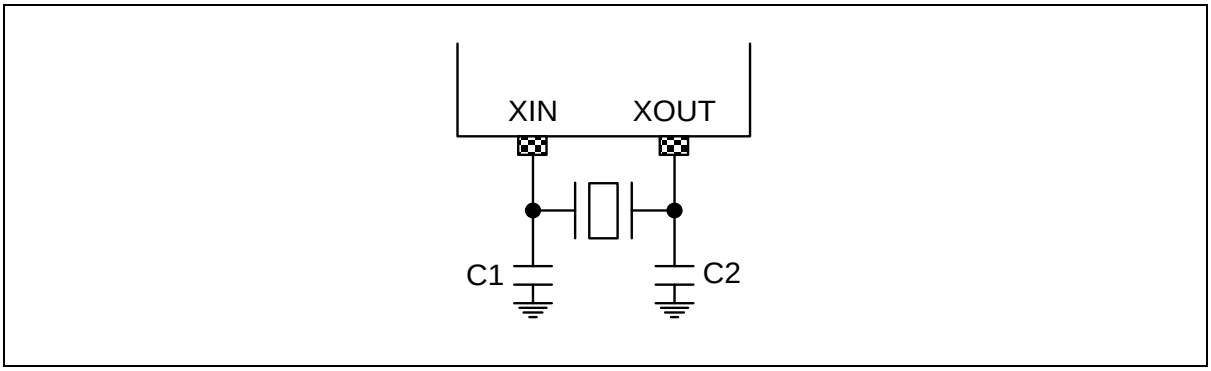


Figure 120. Crystal/Ceramic Oscillator

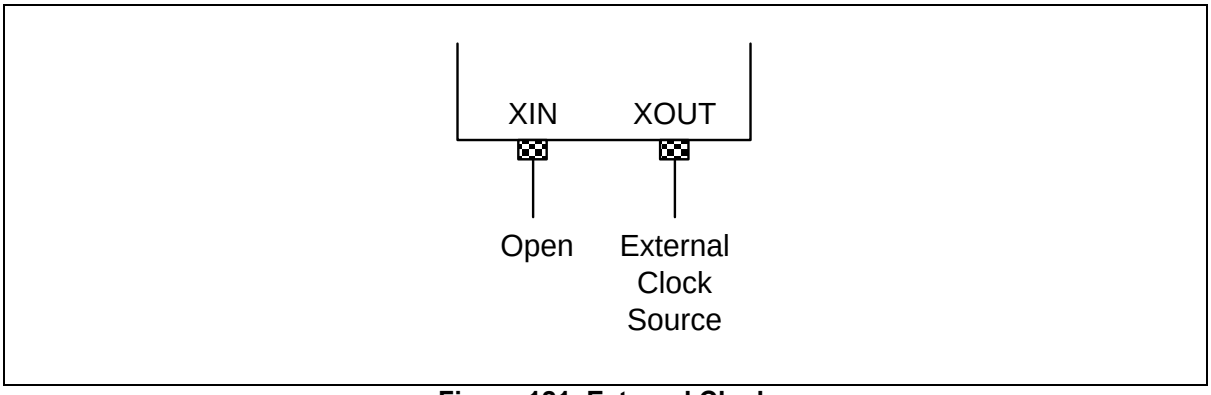


Figure 121. External Clock

23.20 Sub oscillator characteristics

Table 101. Sub Oscillator Characteristics

Oscillator	Parameter	Conditions	Min	Typ	Max	Units
Crystal	Sub oscillation frequency	1.8 V – 5.5 V	32	32.768	38	KHz

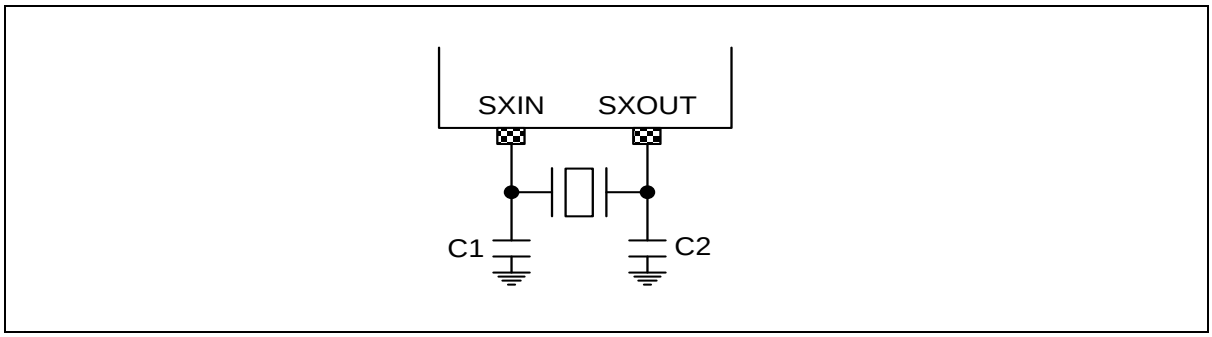


Figure 122. Crystal Oscillator

23.21 Main oscillator stabilization time

Table 102. Main Oscillation Stabilization Time

Oscillator	Conditions	Min	Typ	Max	Units
Crystal	f_{XIN} ≥ 2 MHz Oscillation stabilization occurs when VDD is equal to the minimum oscillator voltage range.	–	–	60	ms
Ceramic		–	–	10	
External Clock	f_{XIN} = 2.0 to 40 MHz XIN input high and low width (t_{XL}, t_{XH})	12.5	–	250	ns

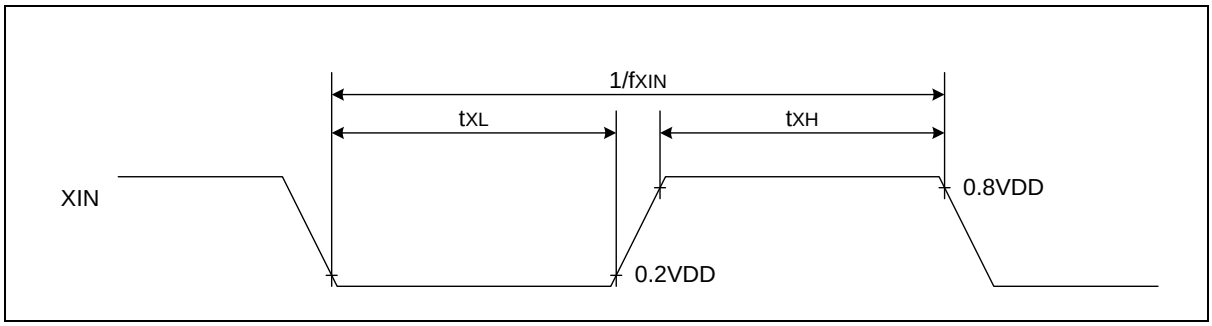


Figure 123. Clock Timing Measurement at XIN

23.22 Sub oscillator stabilization time

Table 103. Sub Oscillation Stabilization Time

Oscillator	Conditions	Min	Typ.	Max	Units
Crystal	–	–	–	10	sec
	VDD=3V, T_A=25°C	–	0.7	1.5	

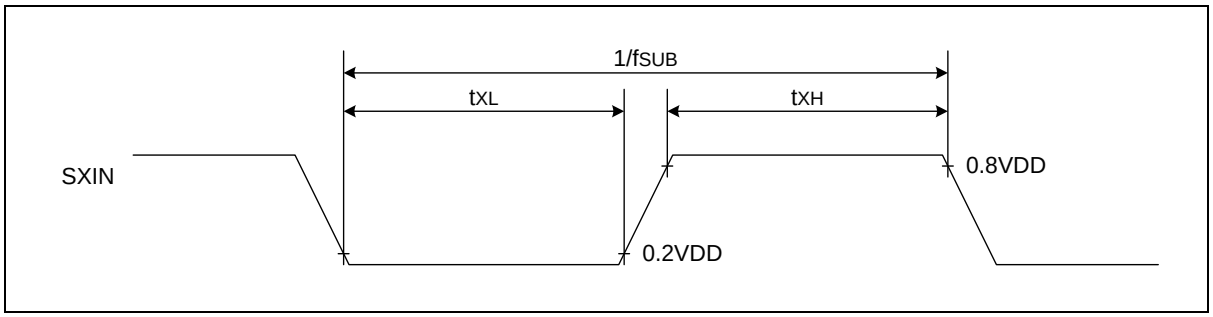


Figure 124. Clock Timing Measurement at SXIN

23.23 Operating voltage range

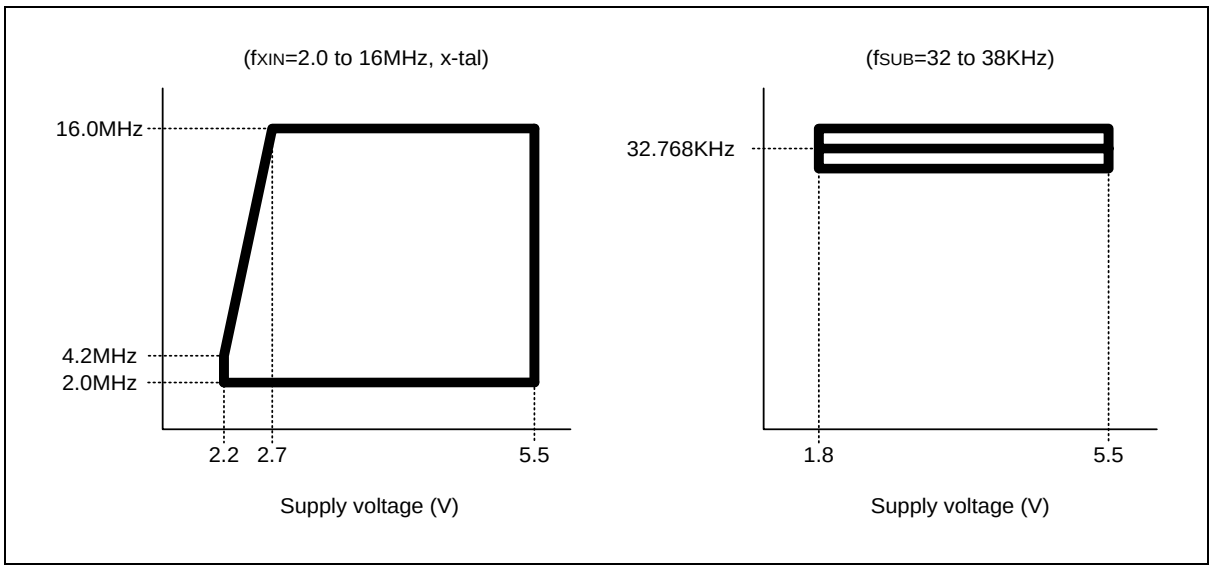


Figure 125. Operating Voltage Range

23.24 PLL electrical characteristics

Table 104. PLL Electrical Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Output Frequency	f_{OUT}	–	-	-	48	MHz
Operating Current	I_{DD}		–	-	1	mA
Duty	f_{DUTY}		40	–	60	%
VCO	f_{VCO}		0.8	–	192	MHz
Input Frequency	f_{IN}		2	8	16	MHz
Locking Time	t_{LOCK}		190			us

23.25 D/A converter characteristics

Table 105. D/A Converter Characteristics

(V_{DD}=1.8-5.5V, V_{DD}=DAVREF, V_{ss}=0)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Resolution	-	-	-	-	5	Bit
Analog Output Voltage	D_{AOUT}		GND+0.02	-	AV_{DD}-0.02	V
Reference Input Voltage	EXTREF		2.7	-	AV_{DD}	V
Integral Nonlinearity	INL	@AVDD=5V or EXTREF_A=5V	-	-±1	±4	LSB
Differential Nonlinearity	DNL		-	±1	±2	LSB
D/AC Current	I_{DAC}		-	0.6	1.0	mA
Conversion Time	-		-	1	2	Us

24 Development tools

This chapter introduces wide range of development tools for A31G21x. ABOV offers software tools, debuggers, and programmers to help a user in generating right results to match target applications. ABOV supports entire development ecosystem of the customers.

24.1 Compiler

ABOV semiconductor does not provide any compiler for A31G21x. However, since A31G21x have ARM's high-speed 32-bit Cortex-M0+ Cores for their CPU, you can use all kinds of third party's standard compiler such as Keil C Compiler. These compilers' output debug information can be integrated with our A-Link and A-Link Pro. Please visit our website www.abovsemi.com for more information regarding the A-Link and A-Link Pro.

24.2 Debugger

The A-Link and A-Link Pro support ABOV Semiconductor's A31G21x MCU emulation in SWD Interface. The A-Link and A-Link Pro use two wires interfacing between PC and MCU, which is attached to user's system. The A-Link and A-Link Pro can read or change the value of MCU's internal memory and I/O peripherals. In addition, the A-Link and A-Link Pro control MCU's internal debugging logic. This means A-Link and A-Link Pro control emulation, step run, monitoring and many more functions regarding debugging.

The A-Link and A-Link Pro run underneath MS operating system such as MS-Windows NT/ 2000/ XP/ Vista/ 7/ 8/ 8.1/ 10 (32-bit, 64-bit).

Programming information using the A-Link and A-Link Pro are provided in Figure 126. More detailed information about the A-Link and A-Link Pro, please visit our website www.abovsemi.com and download the debugger S/W and documents.

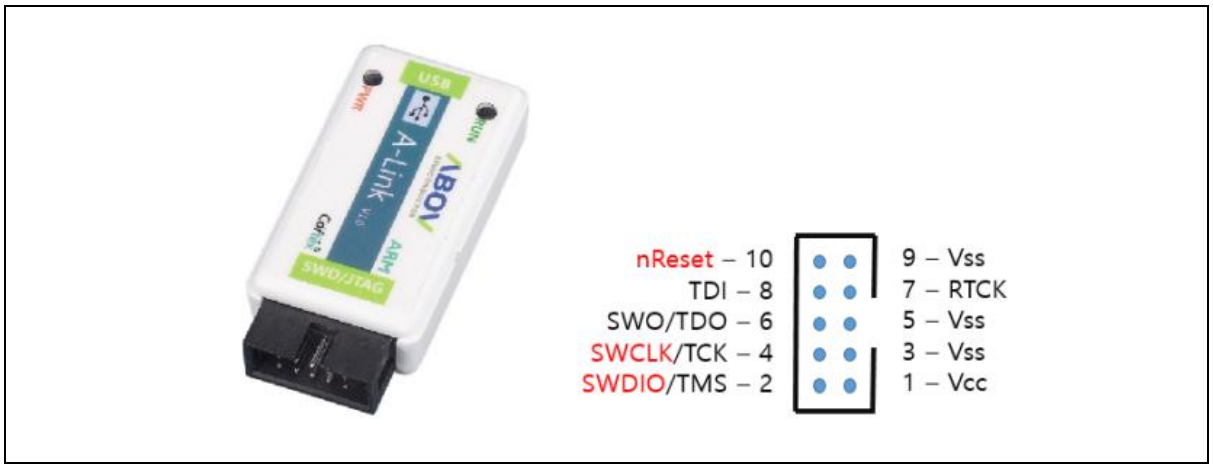


Figure 126. A-Link and Pin Descriptions

24.3 Programmer

24.3.1 E-PGM+

E-PGM+ is a single programmer, and allows a user to program on the device directly.

- Support ABOV devices
- 2~5 times faster than S-PGM+
- Main controller: 32-bit MCU @ 72MHz
- Buffer memory: 1MB

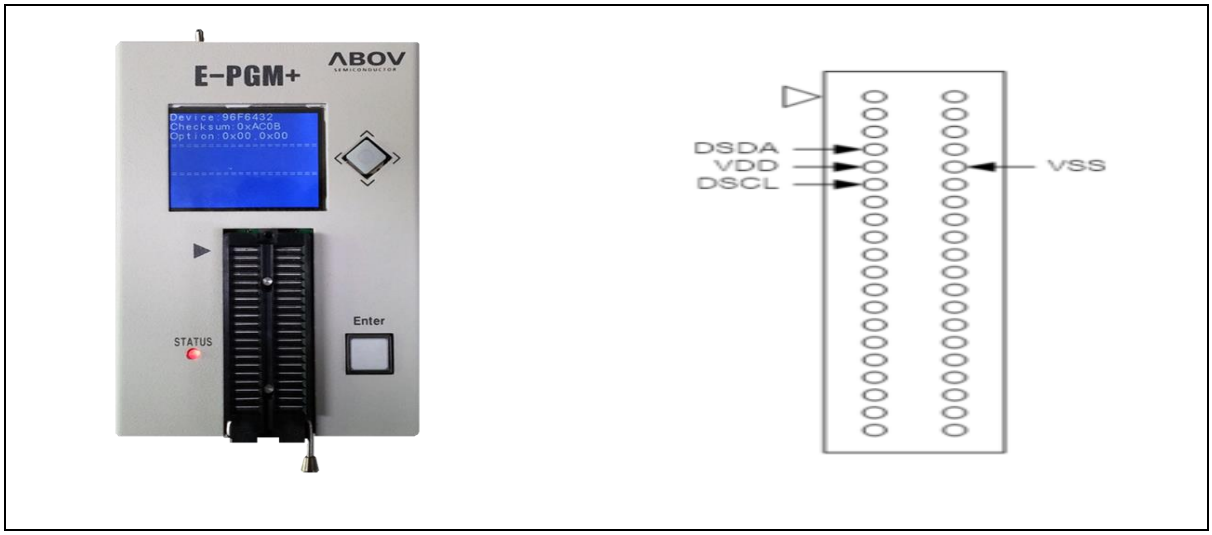


Figure 127. E-PGM+ (Single Writer) and Pin Descriptions

24.3.2 Gang programmer

E-Gang4 and E-Gang6 allows a user to program on multiple devices at a time. They run not only in PC controlled mode but also in standalone mode without PC control. USB interface is available and it is easy to connect to the handler.

Figure 128. E-Gang4 and E-Gang6 (for Mass Production)

25 Circuit design guide

Refer to the Recommended Circuit and Layout Design as shown below.

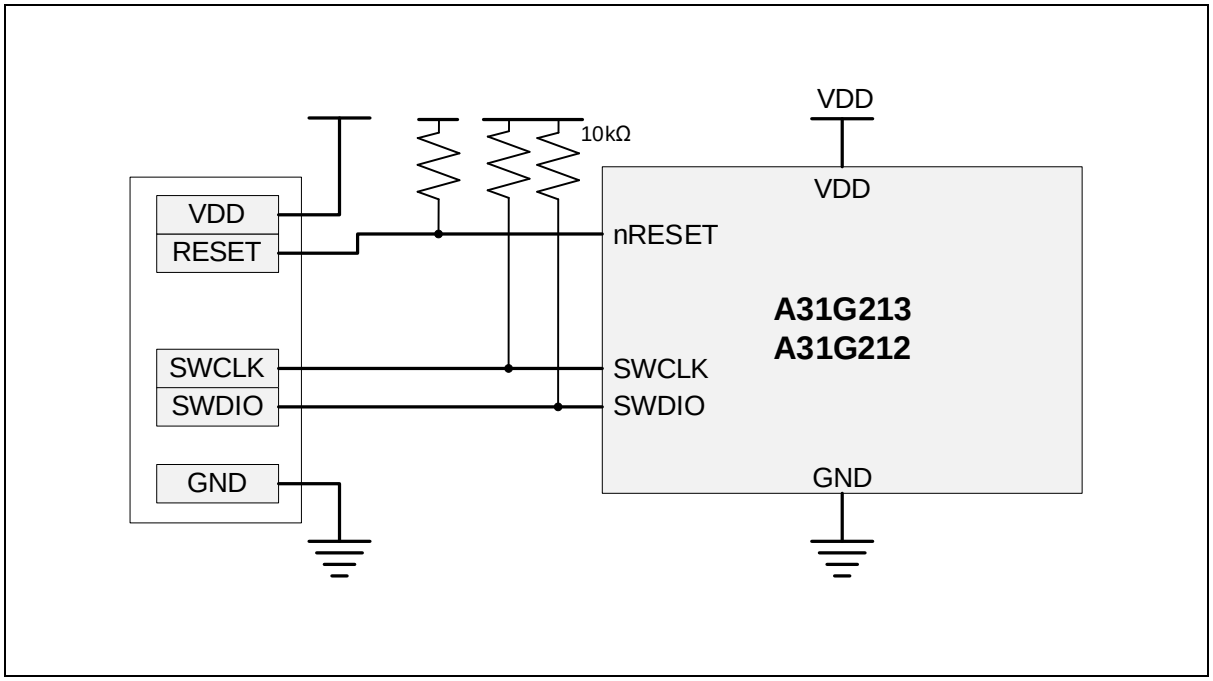


Figure 129. PCB Design Guide for On-Board Programming

26 Package information

This chapter provides A31G21x series package information.

26.1 48 LQFP package information

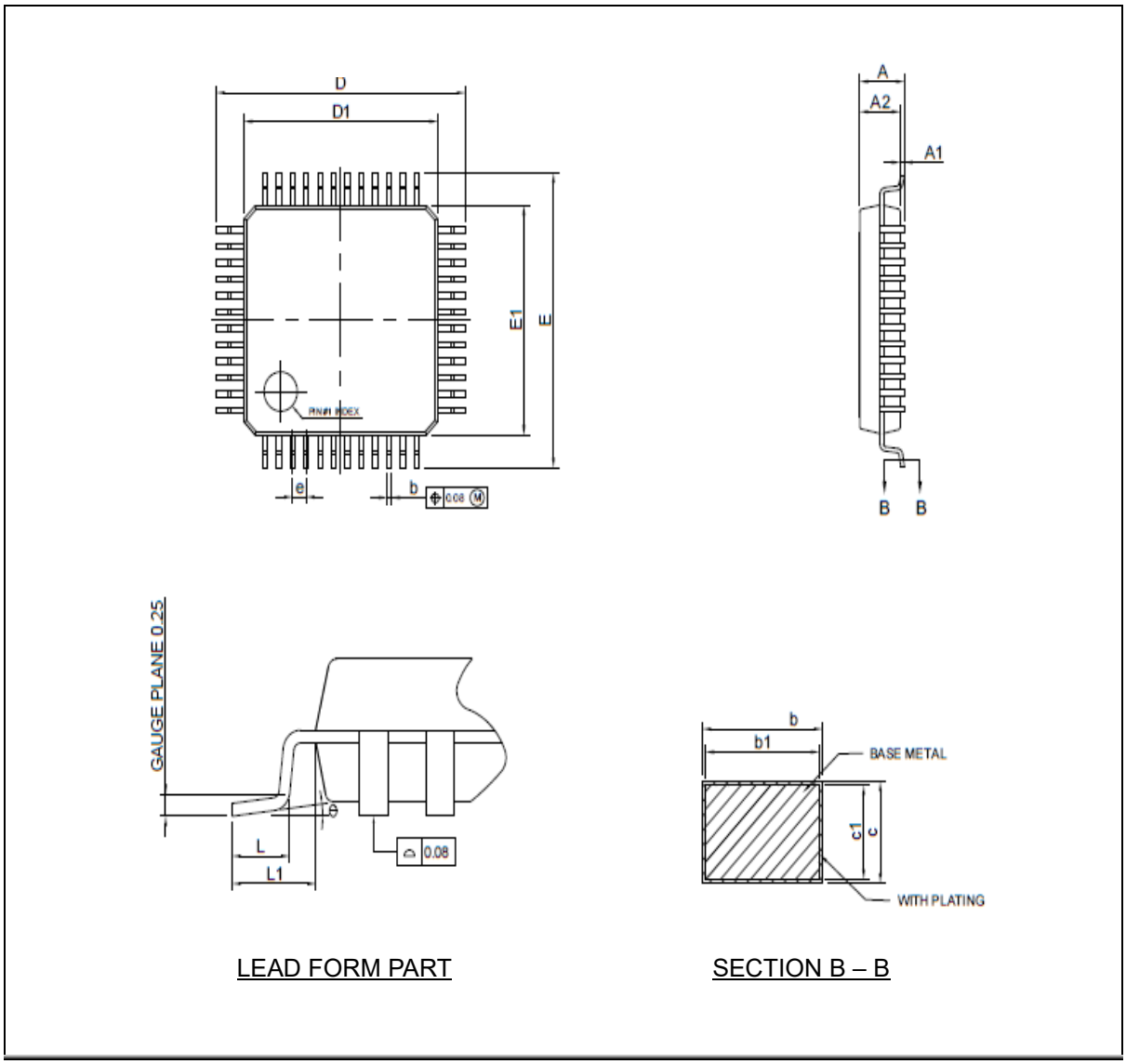


Figure 130. 48 LQFP Package Outline

Table 106. 48LQFP 7 x 7 Package Mechanical Data

Symbol	Dimension (mm)		
	Min.	Nom.	Max.
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
b1	0.17	0.20	0.23
c	0.09	—	0.20
c1	0.09	—	0.16
D	8.80	9.00	9.20
D1	6.80	7.00	7.20
E	8.80	9.00	9.20
E1	6.80	7.00	7.20
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
Θ	0°	3.5°	7°

NOTES:

1. All dimensions refer to JEDEC standard MS-026-BBC.
2. Dimensions 'D1' and 'E1' do not include mold protrusions. Allowable protrusion is 0.25mm per side. 'D1 and 'E1' are maximum plastic body dimensions including mold mismatch.
3. Dimension 'b' does not include dambar protrusion. Allowable dambar protrusion shall not cause the lead width to exceed the maximum 'b' dimension by more than 0.08mm.
4. 'A1' is defined as the distance from the seating plane to the lowest point on the package body.

26.2 44 MQFP package information

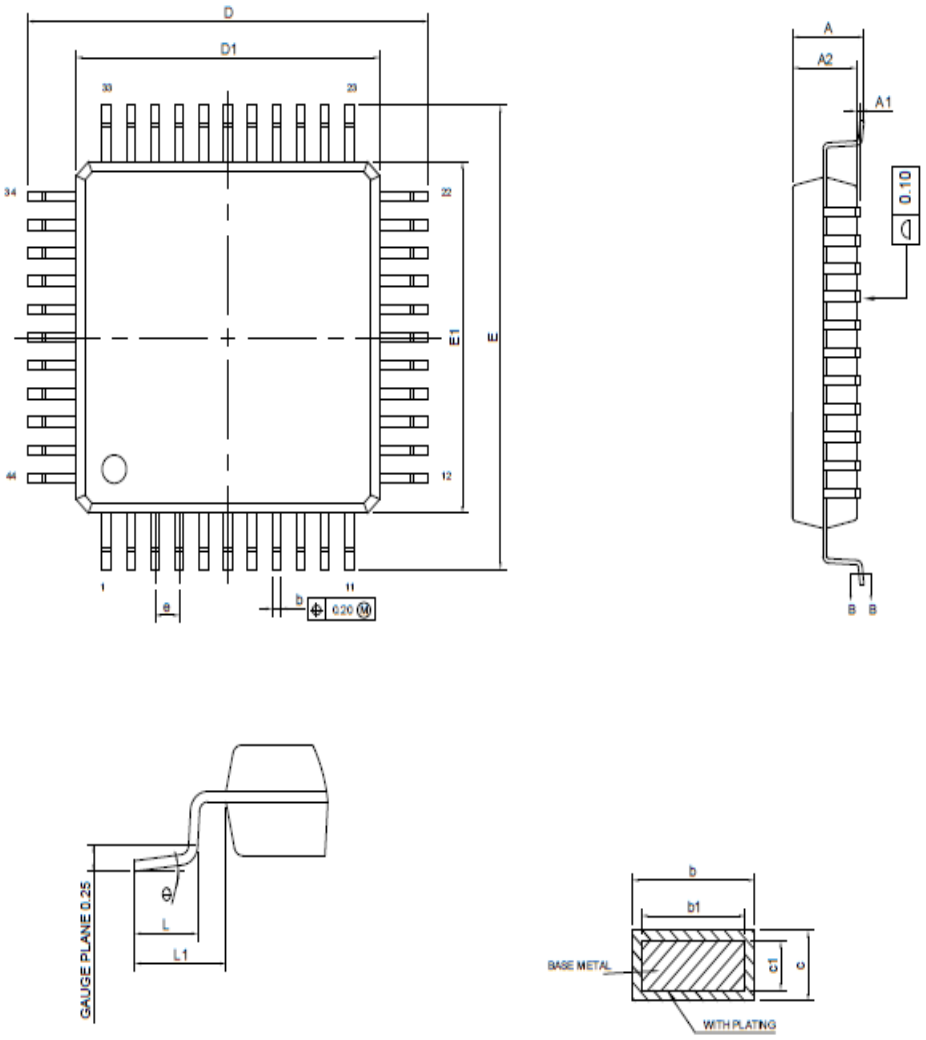


Figure 131. 44 MQFP Package Outline

Table 107. 44MQFP 10 x 10 Package Mechanical Data

Symbol	Dimension (mm)		
	Min.	Nom.	Max.
A	—	—	2.45
A1	0.00	—	0.25
A2	1.80	2.10	2.20
b	0.28	—	0.45
b1	0.28	0.35	0.41
c	0.11	—	0.23
c1	0.11	0.15	0.19
D	12.90	13.20	13.50
D1	9.80	10.00	10.20
E	12.90	13.20	13.50
E1	9.80	10.00	10.20
e	0.80 BSC		
L	0.60	—	1.03
L1	1.60 REF		
Θ	0°	—	8°

NOTES:

1. All dimension refer to JEDEC standard MS-022-AB.
2. Dimensions 'D1' and 'E1' do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions 'D1 and 'E1' do not include mold mismatch.
3. Dimension 'b' does not include dambar protrusion. The dambar protrusion shall not cause the lead width to exceed the maximum 'b' dimension by more than 0.08mm.
4. 'A1' is defined as the distance from the seating plane to the lowest point on the package body.

26.3 32 LQFP package information

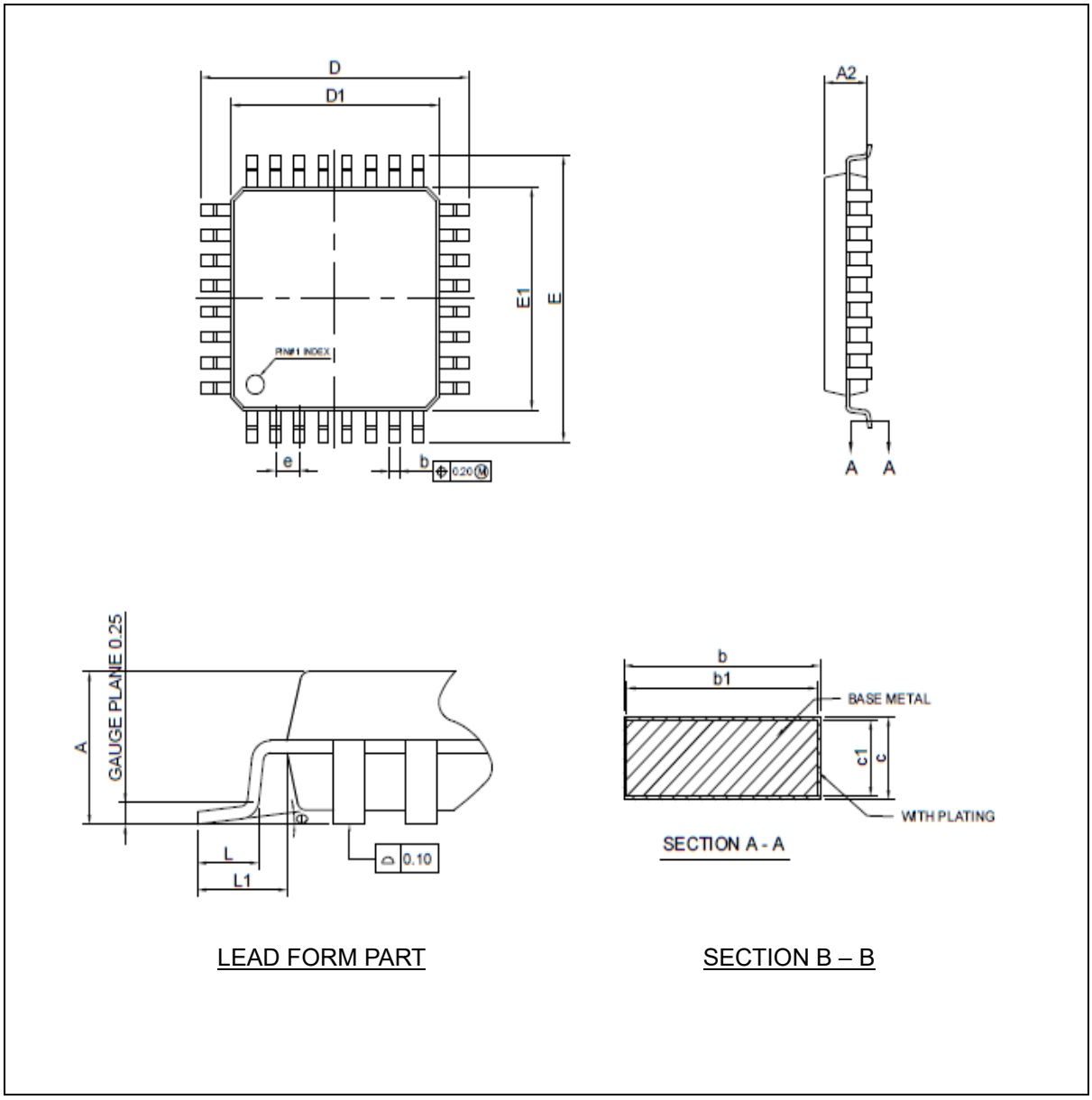


Figure 132. 32 LQFP Package Outline

Table 108. 32LQFP 07 x 07 Package Mechanical Data

Symbol	Dimension (mm)		
	Min.	Nom.	Max.
A	—	—	1.60
A1	0.05	—	0.20
A2	1.35	1.40	1.45
b	0.30	0.37	0.42
b1	0.30	0.35	0.40
c	0.09	—	0.20
c1	0.09	—	0.16
D	8.80	9.00	9.20
D1	6.80	7.00	7.20
E	8.80	9.00	9.20
E1	6.80	7.00	7.20
e		0.80 BSC	
L	0.45	0.60	0.75
L1	1.00 REF		
Θ	0°	3.5°	7°

NOTES:

1. All dimension refer to JEDEC standard MS-026-BBA.
2. Dimensions 'D1' and 'E1' do not include mold protrusion. Allowable protrusion is 0.25mm per side. 'D1 and 'E1' are maximum plastic body size dimensions including mold mismatch.
3. Dimension 'b' does not include dambar protrusion. Allowable dambar protrusion shall not cause the lead width to exceed the maximum 'b' dimension by more than 0.08mm.
4. 'A1' is defined as the distance from the seating plane to the lowest point on the package body.

26.4 32 QFN package information

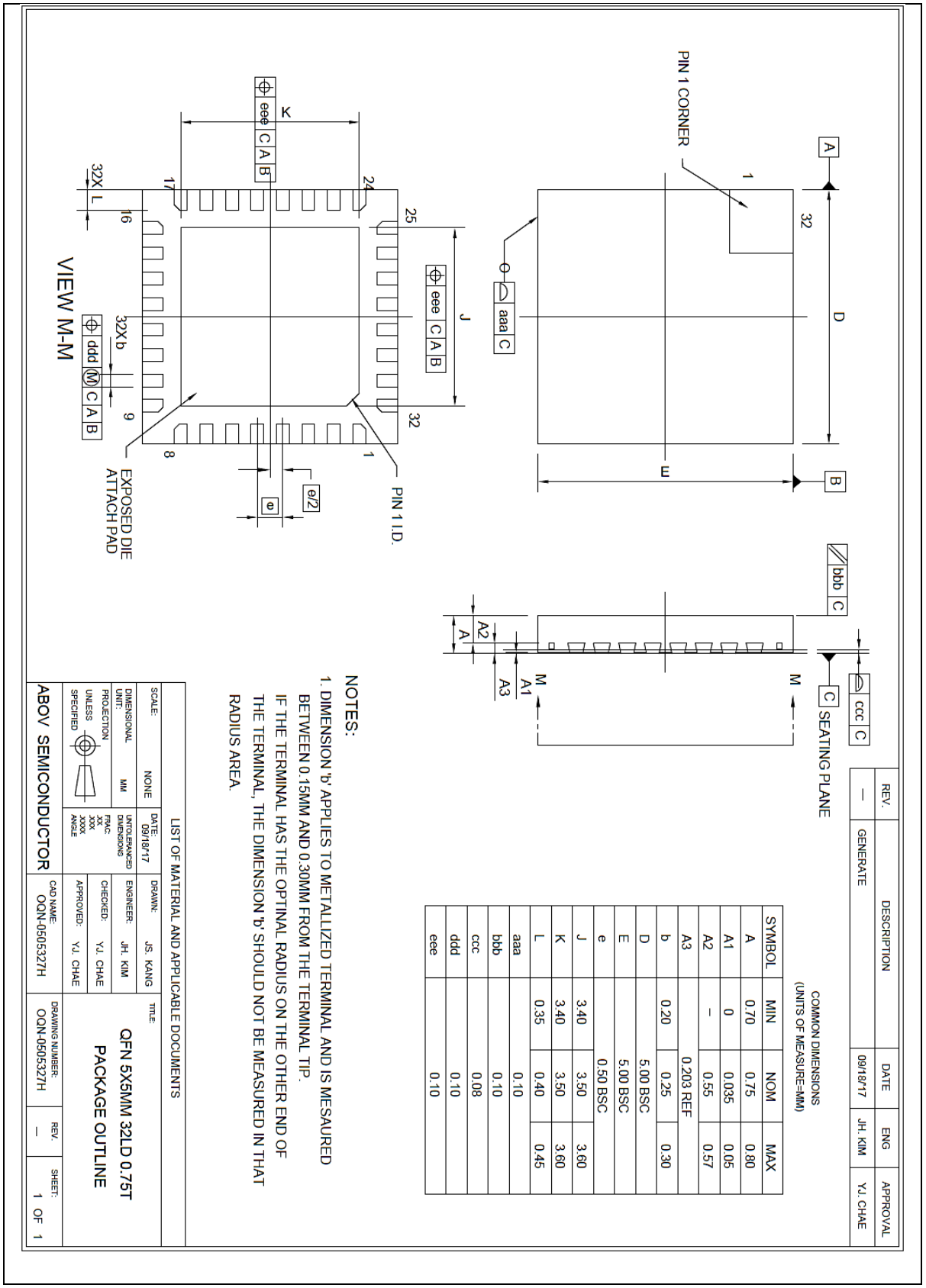
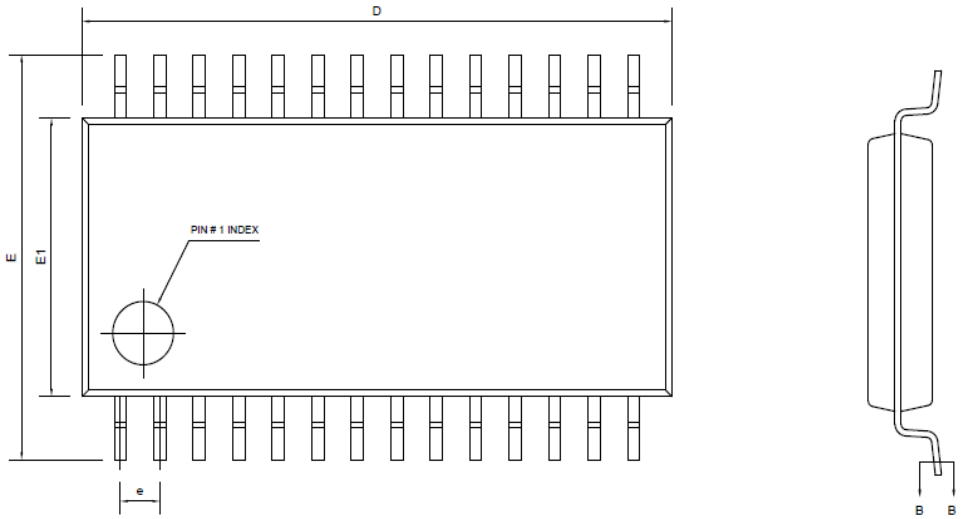
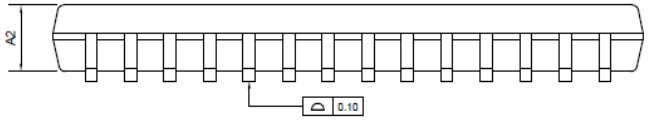
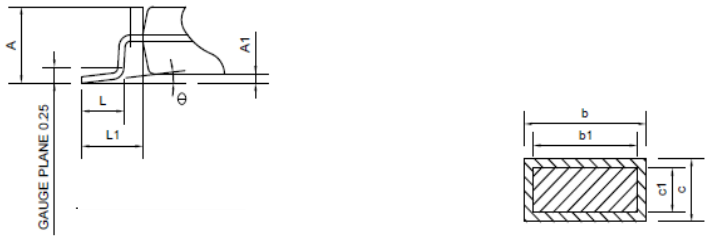


Figure 133. 32 QFN 05 x 05 Package Outline

26.5 28 TSSOP package information







LEAD FORM PART SECTION B – B

Figure 134. 28 TSSOP Package Outline

Table 109. 28TSSOP 4.4MM Package Mechanical Data

Symbol	Dimension (mm)		
	Min.	Nom.	Max.
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
b	0.19	—	0.30
b1	0.19	0.22	0.25
c	0.09	—	0.20
c1	0.09	—	0.16
D	9.60	9.70	9.80
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
Θ	0°	—	8°

NOTES:

1. All dimensions refer to JEDEC standard MO-153-AE.
2. Dimension 'D' does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burr shall not exceed 0.15mm per side.
3. Dimension 'E1' does not include interlead flash or protrusion. Interlead flash or protrusion shall not exceed 0.25mm per side.
4. Dimension 'b' does not include dambar protrusion. Allowable dambar protrusion shall be 0.08mm total in excess of the 'b' dimension at maximum material condition.
5. 'A1' is defined as the vertical distance from the seating plane to the lowest point on the package body.

27 Ordering information

Table 110. A31G21x Device Ordering Information

Device name	Flash	SRAM	UART	USART	SPI	I2C	TIMER	PWM	ADC	I/O Ports	Package
A31G213CL	64KB	6KB	2	2	2	2	4(16bit)/2(32bit)	6	14	44	LQFP-48
A31G213SQ*	64KB	6KB	2	2	2	2	4(16bit)/2(32bit)	6	12	40	MQFP-44
A31G213KN*	64KB	6KB	2	2	1	2	4(16bit)/2(32bit)	6	8	28	LQFP-32
A31G213KU*	64KB	6KB	2	2	1	2	4(16bit)/2(32bit)	6	8	28	QFN-32
A31G213GR*	64KB	6KB	2	1	1	2	4(16bit)/2(32bit)	6	6	24	TSSOP-28
A31G212CL*	32KB	6KB	2	2	2	2	4(16bit)/2(32bit)	6	14	44	LQFP-48
A31G212SQ*	32KB	6KB	2	2	2	2	4(16bit)/2(32bit)	6	12	40	MQFP-44
A31G212KN*	32KB	6KB	2	2	1	2	4(16bit)/2(32bit)	6	8	28	LQFP-32
A31G212KU*	32KB	6KB	2	2	1	2	4(16bit)/2(32bit)	6	8	28	QFN-32
A31G212GR*	32KB	6KB	2	1	1	2	4(16bit)/2(32bit)	6	6	24	TSSOP-28

* For available options or further information on the devices with "*" marks, please contact the ABOV Sales Office.

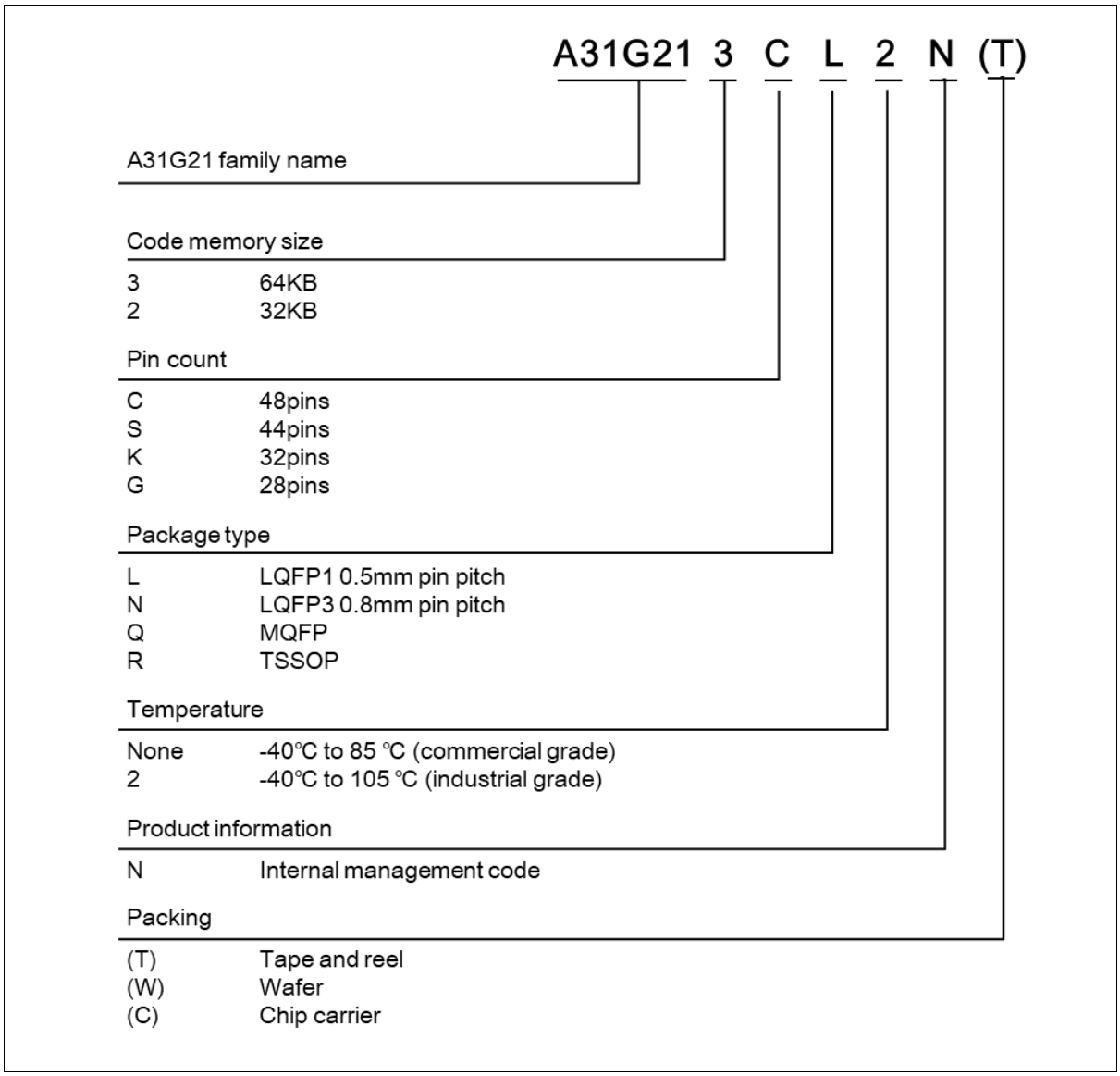


Figure 135. A31G21x Device Numbering Nomenclature

Revision history

Date	Revision	Description
Aug.22, 2019	1.00	1st creation
Dec.30, 2019	1.01	Change to new format
Jul. 10. 2020	1.02	Table 1 : ARM Cortex-M0+ Core modified.
Aug.10. 2020	1.03	TIMERn_CR Registers, Table 3, Table 11, Table 37 modified.
Sep.09. 2020	1.04	WDT_WINDR , Figure 136, Figure 137, Table 88. Absolute maximum rating modified.
Feb.05. 2021	1.05	ADC Conversion timing modified , SCU_PLLCON note add. Flash Program Voltage modified.
Aug.11. 2021	1.06	TS_TRIM_OSC, TS_TRIM_A_OSC modified
Dec.22. 2021	1.07	Table 20. Flash Memory Controller Features modified
Nov.09. 2022	1.08	1) Add a package type, "A31G213KU*, A31G212KU (32 QFN)". Add a Figure, "Figure 5. QFN 32 Pinouts" Add a Figure, "Figure 133. 32 QFN 05 x 05 Package Outline" "27 Ordering information" 2) LED_SEGOE Registers modified
Dec.08. 2022	1.09	Revised the font of this document
Jan.12. 2023	1.10	Change pull-up min/max value in Table 91 DC Electrical Characteristics
Feb.15.2023	1.11	Change UART Spec

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
93, 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