32-bit Cortex-M3 based General Purpose Microcontroller

FlashROM 384 · 256 · 128KB / DataFlashROM 32KB / SRAM 24KB

A33G52x

User's Manual

Version 1.19

2023.05.08

Histoy of changes

Date	Person	Version	Changes
2017/12/07	S.H.Kim	Ver1.0	Published standard version
2017/12/23	S.H.Kim	Ver1.0	Modified the description of PMU_PER, PMU_PCCR registers Changed block diagram of PMU
2018/01/04	S.H.Kim	Ver1.0.1	Modified descriptions of Table 1.2 Replaced TRACEDIV with Reserved FMC_AR description and register
2018/01/18	S.H.Kim	Ver1.0.2	Modified block diagrams of PMU, Timer, I2C, SPI, UART, ADC, etc. Added Nomenclature of A33G52x (Figure 1.2) Modified the description of PMU_CMR, PMU_MCMR
2018/01/31	S.H.Kim	Ver1.0.3	Added VDD Fall/Rise time rate to recommendation for operating.
2018/02/21	S.H.Kim	Ver1.0.4	Added Reset Block diagram (3.3.2) Added the description of Cold-Reset and Warm-Reset (3.6.4) Added the description of IRQ interrupt (2.2)
2018/03/23	S.H.Kim	Ver1.0.5	Modified VDD-CORE Modified the description of PMU_PLLCON register Modified the description of FMC_CRC register
2018/05/11	S.H.Kim	Ver1.0.6	Modified overall contents Modified Table 16.2 Modified the timing diagram of 16-bit timer (Figure 8.2 ~ 8.7)
2018/07/09	S.H.Kim	Ver1.0.7	Modified Figure 1.2. Device Nomenclature of A33G52x
2018/10/18	S.H.Kim	Ver1.0.8	Modified the descriptions of 16.3.7
2018/11/22	S.H.Kim	Ver1.0.9	Removed Figure 1.2 Device Nomenclature of A33G52x according to internal policy. Updated the List of Figures
2019/01/03	S.H.Kim	Ver1.0.10	The 'Read cycle time (tAAD)' specification was changed from '50ns' to '40ns' in Table 16.15, Table 16.17 and Table 16.19. The flash access timing was changed from '20MHz' to '25MHz' in 14.4.1 FMC_CFG Description and Table 14.5.
2019/03/20	S.H.Kim	Ver1.0.11	Modified the description of 14.4.1 Modified Table 14.5
2019/09/30	S.H.Kim	Ver1.0.12	The description of the 6th bit offset in the SPI_CR register was modified.
2019/11/22	H.G.Song	Ver1.0.13	Added the description of system parameter 1 (Figure 14.2) (11.5.12)
2020/01/30	S.H.Kim	Ver1.0.14	Table 16.4 I_{OL} min. value is modified. The copyright statement has been modified.
2020/02/28	S.H.Kim	Ver1.0.15	Modified Figure 10.1 I2C block diagram Added the notification of Table 10.2 Added the chapter 10.5.6 Debounce function for I2C
2020/03/09	S.H.Kim	Ver1.0.16	Modified Table 16.3, Table 16.4 Min. Input High Voltage. Added Table 16.7 DC chacracteristics of IOSC temperature specification. Modified Table 16.6
2020/07/10	S.H.Kim	Ver1.0.17	Modified pin layout of 100MQFP, 100LQFP, 80LQFP packages Modifed pin configuration of PA0, PC2, PC3, PC4 Modified Table 16.6 Added the notification of Figure 14.2 Modified the description of PMU_PLLCON[13] Modified 80-pin package’s ADC channel number of Ordering Information Modified the descriptions of PMU_RSER and PMU_RSSR registers Modified ‘1-word byte’ to ‘1-word’ Removed ‘AVREF’ of PA0 pin Added SWDIO, SWCLK, SWO pin name
2020/09/18	S.H.Kim	Ver1.0.18	Added Maximum Operating Junction Temperature (T_J) Specification
2021/10/26	S.H.Kim	Ver1.0.19	Added the note to 9.6.4 Modified the description of SYNC in PWMn_CTRL register.
2023/05/08	T.W.Lim	Ver1.19	Changed the format of the revision number to “X.YZ” according to internal policy.

Table of Contents

List of Figures

List of Tables

CHAPTER 1. Overview

1.1 Overview

A33G52x processor is designed for the main controller of various household appliances.In particular, accordance with the tendency that the microcontroller is becoming more complicated and high performance in consumer electronics, ARM's high-speed 32-bit Cortex-M3 Core is used. In addition, for handling more features, this microcontroller has a variety of peripheral devices and large amounts of flash memory.

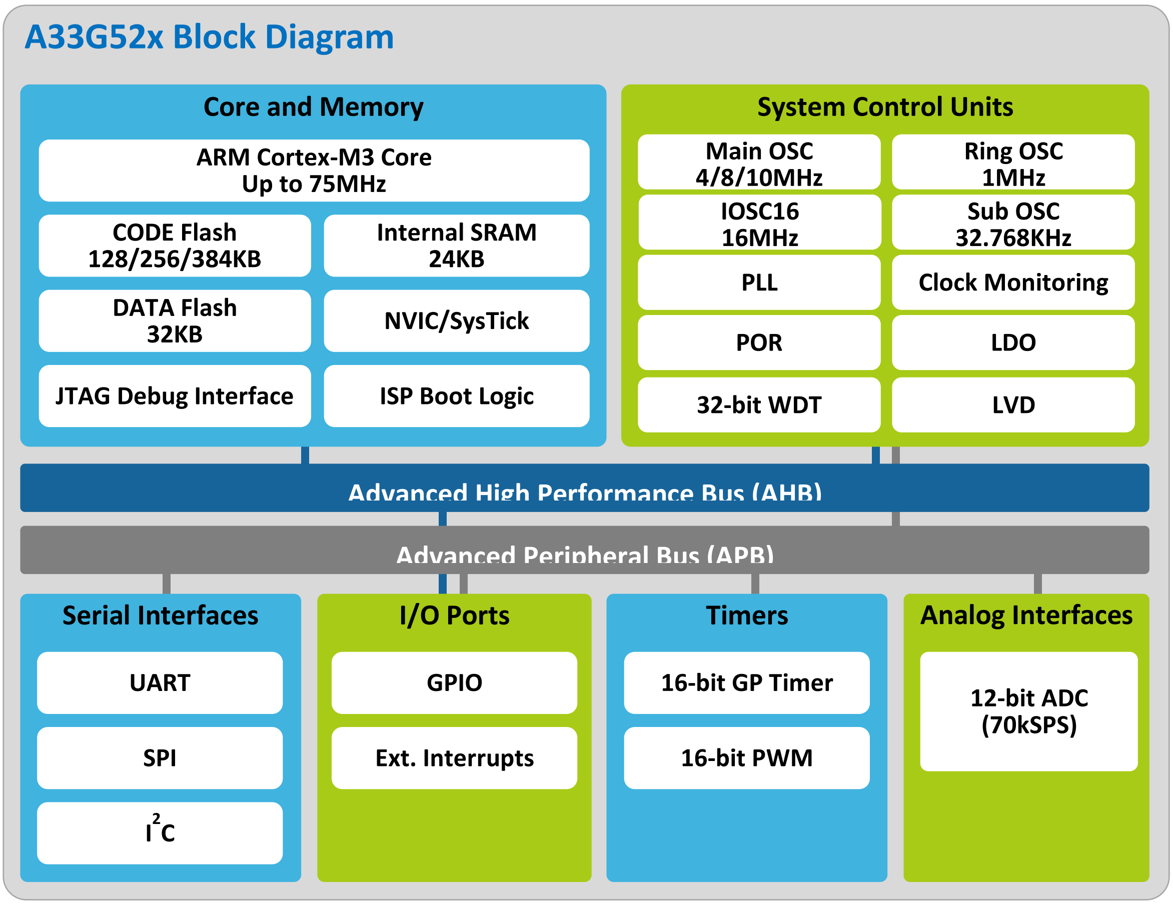


Figure 1.1. Block diagram of A33G52x Series MCUs

1.2 Ordering Information

Table 1.1. Ordering Information

Product Name	CODE Flash [KB]	DATA Flash [KB]	SRAM [KB]	UART [ch]	SPI [ch]	I2C [ch]	PWM [ch]	12-bit ADC [ch]	I/O Ports [ch]	Package
A33G527VQ	384	32	24	4	2	2	8	16	90	100 MQFP
A33G527VL	384	32	24	4	2	2	8	16	90	100 LQFP14
A33G527RL*	384	32	24	4	2	2	8	10	60	64 LQFP12
A33G526VQ*	256	32	24	4	2	2	8	16	90	100 MQFP
A33G526VL*	256	32	24	4	2	2	8	16	90	100 LQFP14
A33G526MM	256	32	24	4	2	2	8	15	71	80 LQFP14
A33G526ML*	256	32	24	4	2	2	8	15	71	80 LQFP12
A33G526RL	256	32	24	4	2	2	8	10	60	64 LQFP12
A33G526RM*	256	32	24	4	2	2	8	10	60	64 LQFP10
A33G524MM*	128	32	24	4	2	2	8	15	71	80 LQFP14
A33G524ML*	128	32	24	4	2	2	8	15	71	80 LQFP12
A33G524RL	128	32	24	4	2	2	8	10	60	64 LQFP12
A33G524RM*	128	32	24	4	2	2	8	10	60	64 LQFP10

: For available options for further information on the device with a '' mark, please contact the ABOV Sales Offices.

1.3 Main Features

A33G52x has a Cortex-M3 32-bit embedded core, and has a variety of peripherals. A brief description of each part is as follows:

Table 1.2. Outline of Specifications

Category	**Module**	**Description**
Core	CPU	○ Max. Speed : 75 MHz ○ 32-bit ARM Cortex-M3 CPU ○ CPU Register Set General Purpose Register : 32-bit Thumb-2 Instruction Main Stack Pointer (MSP) & Process Stack Pointer (PSP) : R13 Link Register (LR) : R14 Program Counter (PC) : R15 ○ Data Alignment: Little endian ○ Harvard Architecture ○ AHB/APB
	Interrupt	○ NVIC (Nested-Vectored Interrupt Controller) ○ 64 Peripheral Interrupts ○ 3-bit width of Group Priority: 8-step priority.
Memory	Code Flash	○ Capacity : 384KB (A33G527) 256KB (A33G526) 128KB (A33G524) ○ Built-in high capacity code flash memory ○ Max. 25 MHz Flash Access Timing ○ 1Byte unit PROGRAM ○ 512Bytes or 1KB Sector ERASE ○ Self-PROGRAM 1-word (4 Bytes) PROGRAM Supports to update data in some Code Flash memory region during execution of user program in Code area. ○ CRC16 generation and verification of data in code flash. ○ Endurance : 10,000 cycles ○ Lifetime : 10 years
	Data Flash	○ Capacity: 32KB ○ Max. 25 MHz Flash Access Timing ○ 1Byte unit PROGRAM, ○ 1KB unit Sector ERASE ○ CRC16 generation and verification of data in data flash. ○ Endurance : 100,000 cycles ○ Lifetime : 10 years

	Boot Rom	○ Enter the boot mode of the processor according to the external BOOT pin input ○ SPI and UART Interfaces in BOOT mode ○ In System Programming User data can be programmed in the internal flash memory on the application board in BOOT mode.
	SRAM	○ Capacity: 24KB ○ Available as a program region ○ Suitable high-speed operation for time-critical code execution. ○ The SRAM area is partially remapped to the interrupt vector area.
PMU (Power Management Unit)	Operation Modes	○ Run Mode (Run) ○ Sleep Mode (Idle) ○ Deep-Sleep Mode (Power Down)
	Clock	○ 1MHz Internal Ring Oscillator (RINGOSC) ○ 16MHz Internal Oscillator (IOSC16) ○ 4~10MHz External Main Oscillator (MXOSC) ○ 32.768kHz External Auxiliary Oscillator (SXOSC) ○ PLL Frequency synthesizer (PLL) High frequency operation (Max. 75MHz Output Frequency) 1MHz-unit fine-tuning of output frequency. ○ Main system clock monitoring. ○ External oscillation clock monitoring and non-oscillation error handling function. ○ Frequency dividing function for main system clock (HCLK) and peripheral device module clock (PCLK).
	Reset	Reset events are occurred by below reset sources : ○ Main Clock Fail ○ External nRESET Pin ○ Core reset ○ Software Reset ○ POR (Power-On Reset) ○ LVD (Low Voltage-Monitoring) Reset ○ External main oscillation error
	LDO (Low Drop Out)	○ Integrated LDO (Low Drop Out) for low-power operation
	POR (Power On Reset)	○ Internal core voltage monitoring and reset signal generation
	LVD (Low Voltage Detector)	○ 8-step voltage detection level ○ LVD (Low Voltage Detector) reset ○ LVD Interrupt ○ Wake-up by LVD function after sleep or deep-sleep mode.

	Low-Power Consumption	○ Low-power operation mode Sleep mode (Idle) Deep Sleep Mode (Power Down)
	Wake-up Event	Wake-up events are occurred by below wake-up sources : ○ GPIOA~GPIOF ○ FRT ○ Failure of external main oscillation ○ WDT ○ LVD ○ Rapid wake-up operation with internal oscillator and external clock source
GPIO	PCU GPIO	○ General purpose I/O Ports ○ 100-pin LQFP/MQFP I/O pins: 90 ○ 80-pin MQFP I/O pins: 71 ○ 64-pin LQFP I/O pins: 60 ○ Configuration of pin mode Push-Pull Output Open-Drain Logic Input Analog Input ○ Setting pin function using MUX ○ High/Low Level detection and Interrupt ○ Rising/Falling edge detection and interrupt ○ Setting Pull-up/Pull-down/Debounce ○ Large-current Port (Port D) ○ Separate bit set/reset function ○ Wake-up event by external asynchronous input

Timer	16-bit Timer	○ 16-bit general purpose up-count timer ○ 10 channels TnC: Timer input 10-ch TnO: Timer output 10-ch ○ Timer operation modes Periodic timer mode Counter mode PWM mode Capture mode ○ Interrupt Events Timer/Counter match interrupt Timer overflow interrupt ○ Timer Input clock MXOSC/IOSC16/SXOSC/RINGOSC Timer input by external TnC pin in capture mode ○ Output timer clock signal by external TnO pin ○ 10-bit prescaler
	WDT	○ 32-bit down-count timer ○ Reset event and peridic Interrupt ○ MXOSC/IOSC16/SXOSX/RINGOSC clock source selection ○ 8-step prescaler
	FRT	○ 32-bit Free-run Timer system internal's time calculation 32-bit up-count timer ○ Periodic interrupt mode Occurred timer interrupt according to the time interval set by the user. Integrated comparator for match interrupt ○ Overflow interrupt ○ 8-step prescaler
PWM (Pulse-Width Modulation)	PWM	○ PWM generator with 8 channels ○ PWM signal with 16-bit independent counter ○ consists of 1-unit per 4 channels. ○ 8-bit prescaler per 1 unit ○ Configuration of duty and period of PWM output signal ○ 1/2, 1/4, 1/8, 1/16 clock divider ○ 16-bit period and count value set ○ built-in channels that outputs inverted PWM signal

Serial Interfaces	I2C	○ Standard I2C communication specification ○ 2 channels ○ Supports master/slave per channel ○ 7-bit slave address ○ Byte-by-byte data communication by interrupt and polling type ○ I2C Max. transfer rate: 400kbps ○ Configuring I2C clock and data signal latency
	SPI	○ 2 channels synchronous serial communication port ○ Double buffer structure for high-speed transmission ○ Master / slave selection function of communication channel ○ Setting function for transmission data Number of bits (8/9/16/17 bit variable) SPI clock speed LSB-first or MSB first transmission ○ Supports SPI0 channel when entering BOOT mode.
	UART	○ 16550/16450 compatible asynchronous serial communication port ○ 4 channels 16550- compliant device with 2-channel FIFO 16450-compliant device with two-channel double buffer ○ Built-in fractional point divider to improve baud rate accuracy. ○ Supports UART0 channel when entering BOOT mode. ○ Single/Multi-Sampling of receiving data
12-bit A/D converter	ADC	○ 12-bit resolution ○ Single SAR A/D converter ○ Built-in Analog MUX for multiple input ○ Input voltage for ADC conversion GNDV ~ AVDDV ○ 70kSPS A/D conversion time Max. 15us/channel (AVDD=5.0V, A/D conversion clock=4MHz) ○ Interrupt of end of A/D conversion ○ Trigger source of A/D conversion: Timer 7-channels ○ A/D conversion by internal START bit or external trigger source ○ Channels 100-pin LQFP/MQFP : 16-ch x 1-unit 80/64-pin LQFP : 10-ch x 1-unit
Debug	Debug Interfaces	○ 100-pin MQFP/LQFP Trace / JTAG / SWD ○ 80-pin LQFP JTAG / SWD ○ 64-pin LQFP JTAG / SWD

1.4 Block Diagram

A33G52x is shown as below consists of a variety of peripheral devices.

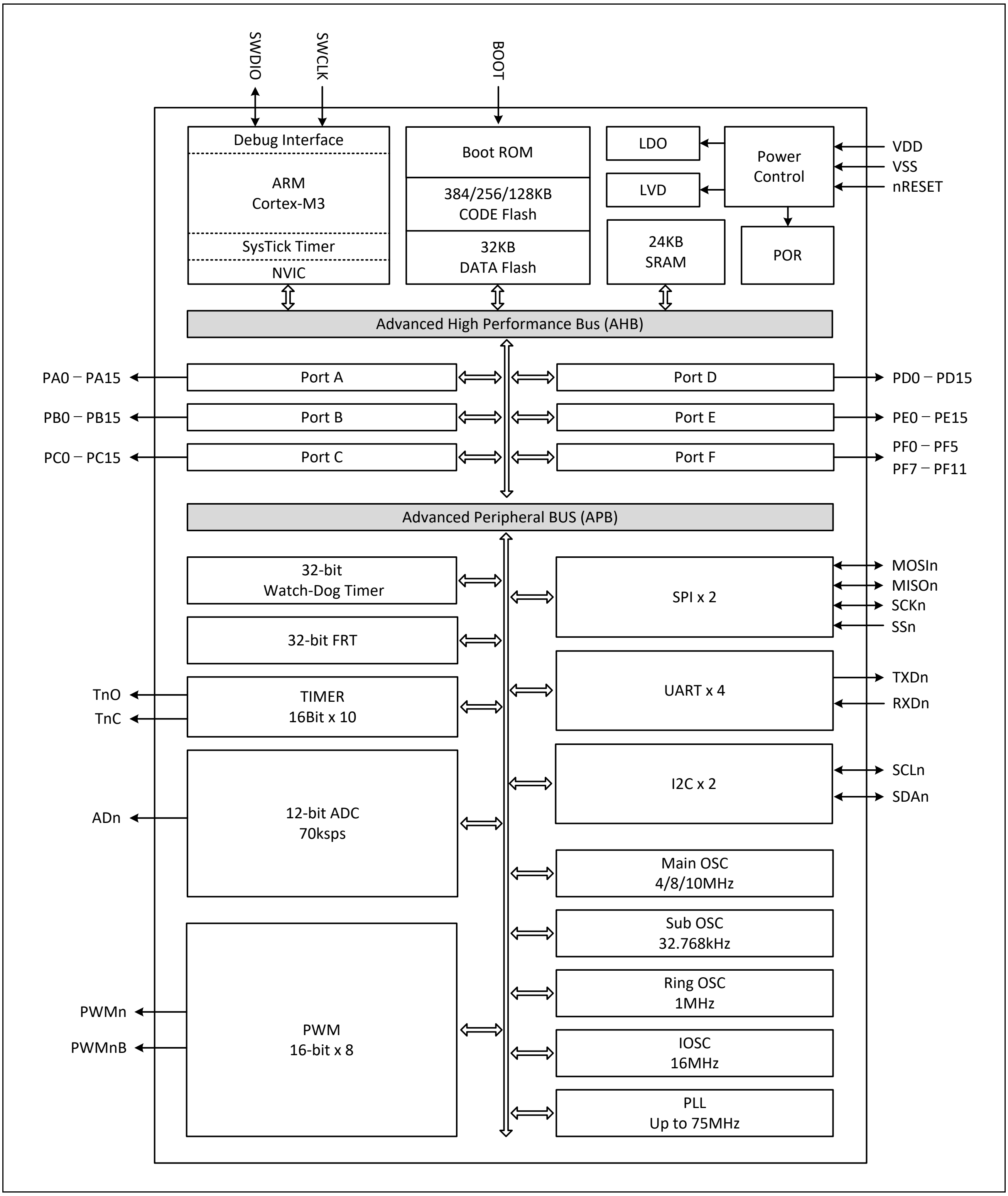


Figure 1.2. Internal block diagram of A33G52x MCUs

1.5 Pin Layout of Packages

1.5.1 A33G527VQ / A33G526VQ (100MQFP)

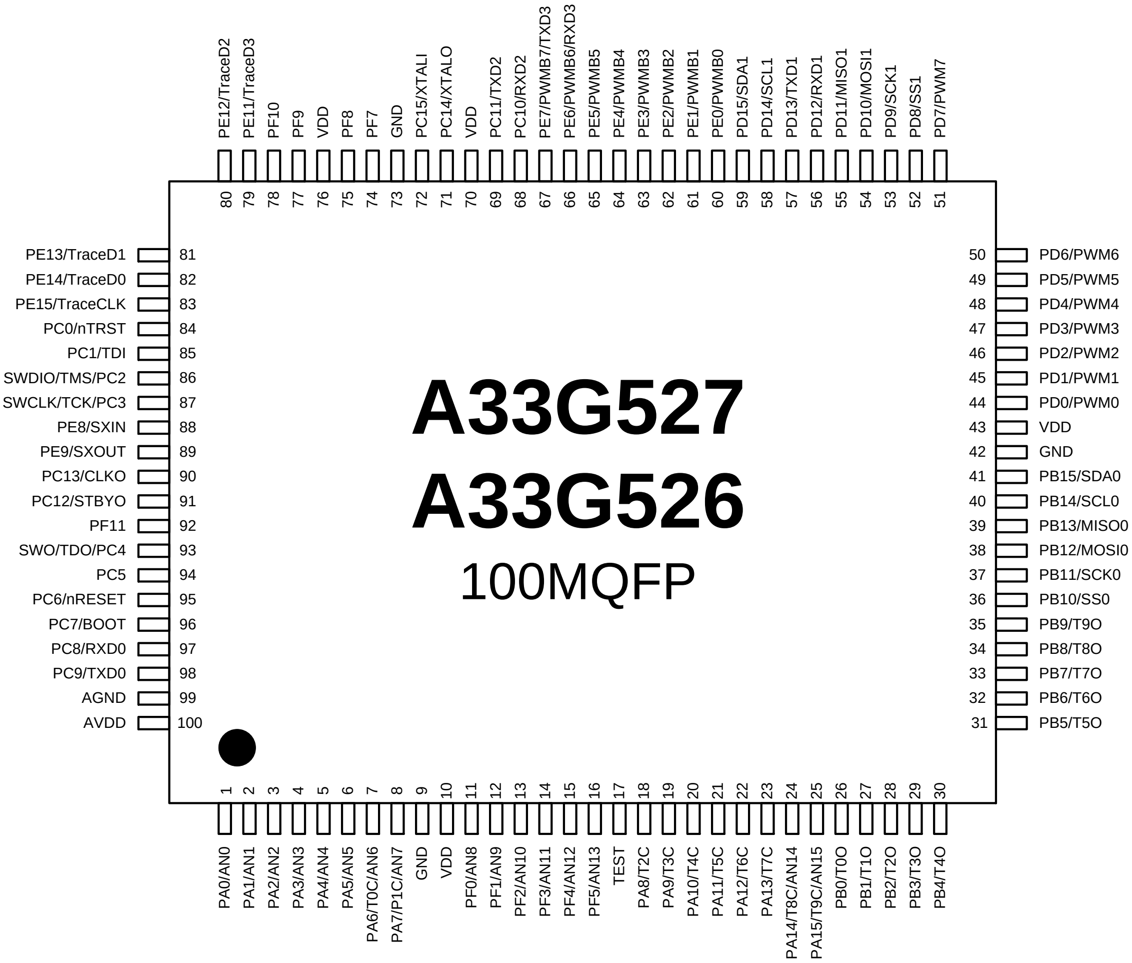


Figure 1.3. Pin layout of . A33G527VQ / A33G526VQ (100MQFP)

1.5.2 A33G527VL / A33G526VL (100LQFP14)

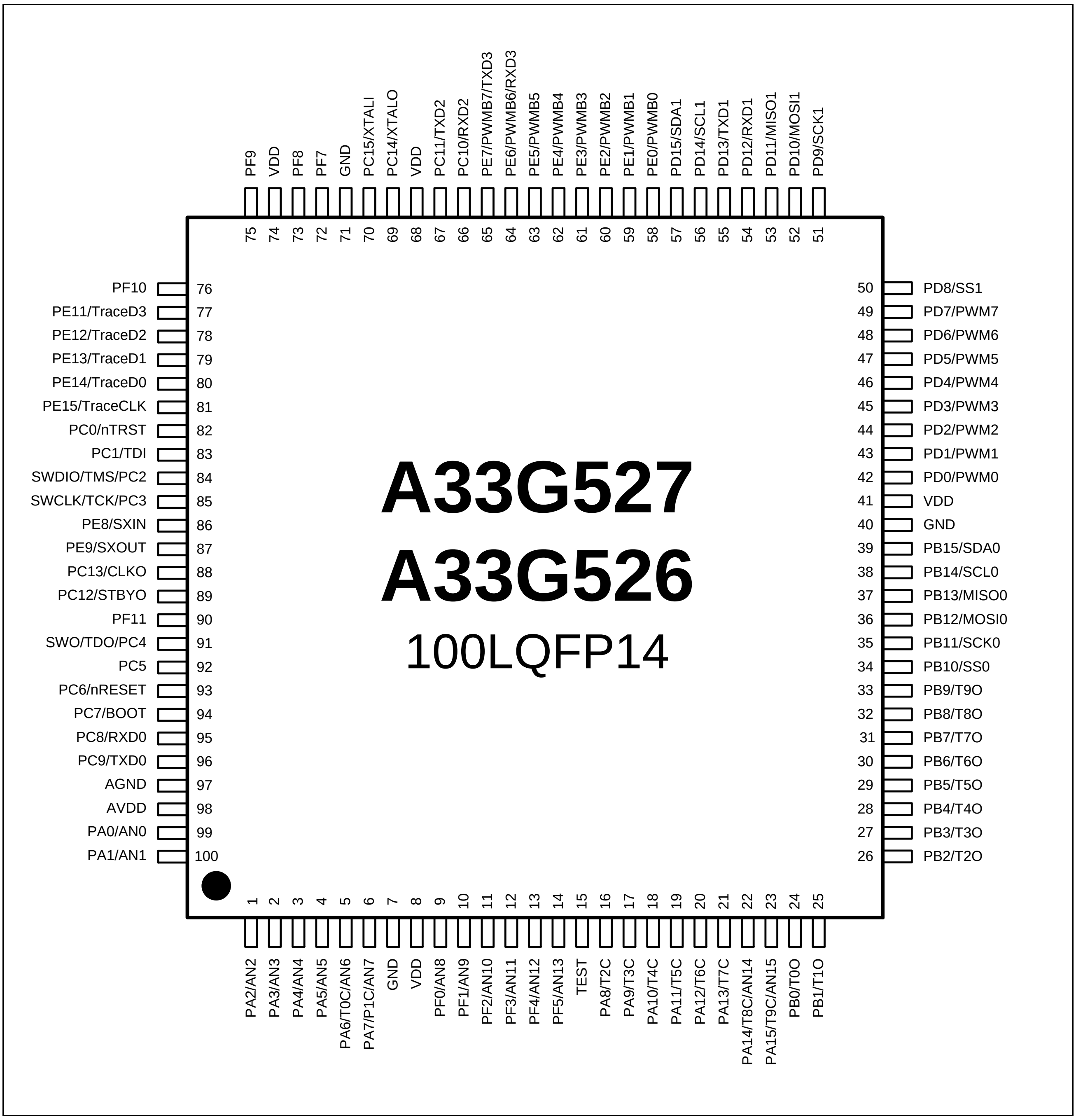


Figure 1.4. Pin layout of . A33G527VL / A33G526VL (100LQFP14)

1.5.3 A33G526MM / A33G524MM (80LQFP14)

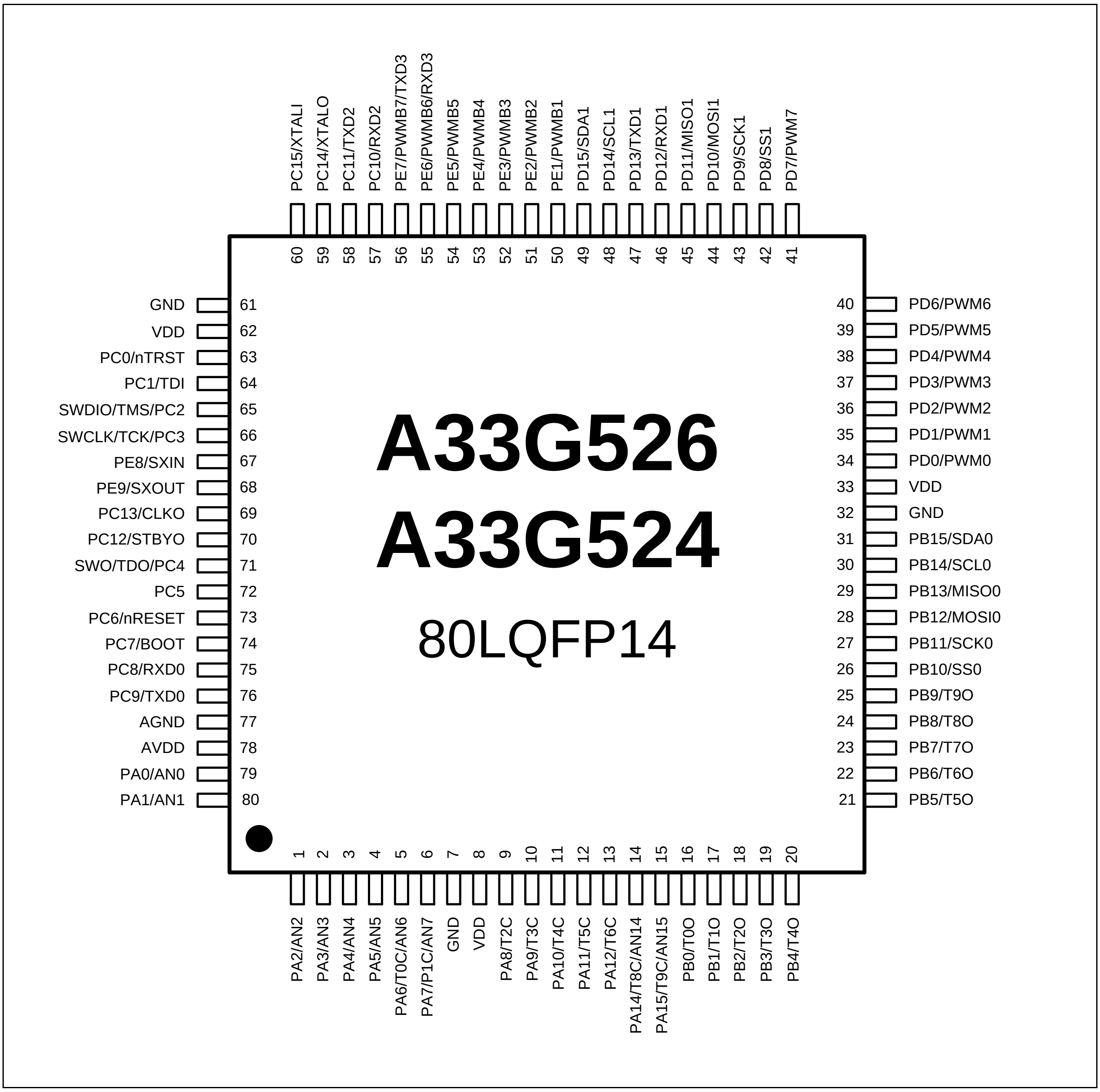


Figure 1.5. Pin layout of . A33G526MM / A33G524MM (80LQFP14)

1.5.4 A33G526ML / A33G524ML (80LQFP12)

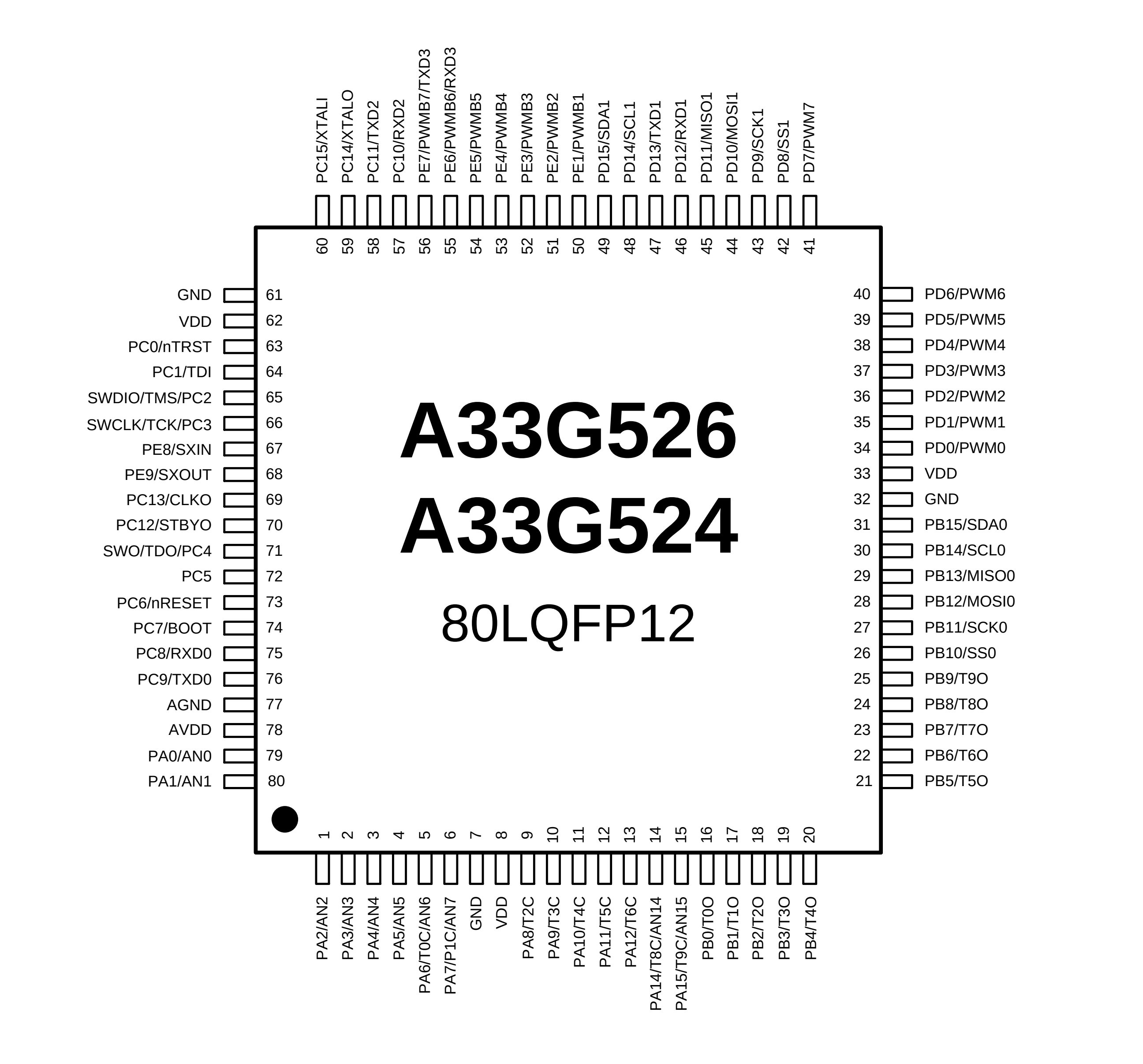


Figure 1.6. Pin layout of . A33G526ML / A33G524ML (80LQFP12)

1.5.5 A33G527RL / A33G526RL / A33G524RL (64LQFP12)

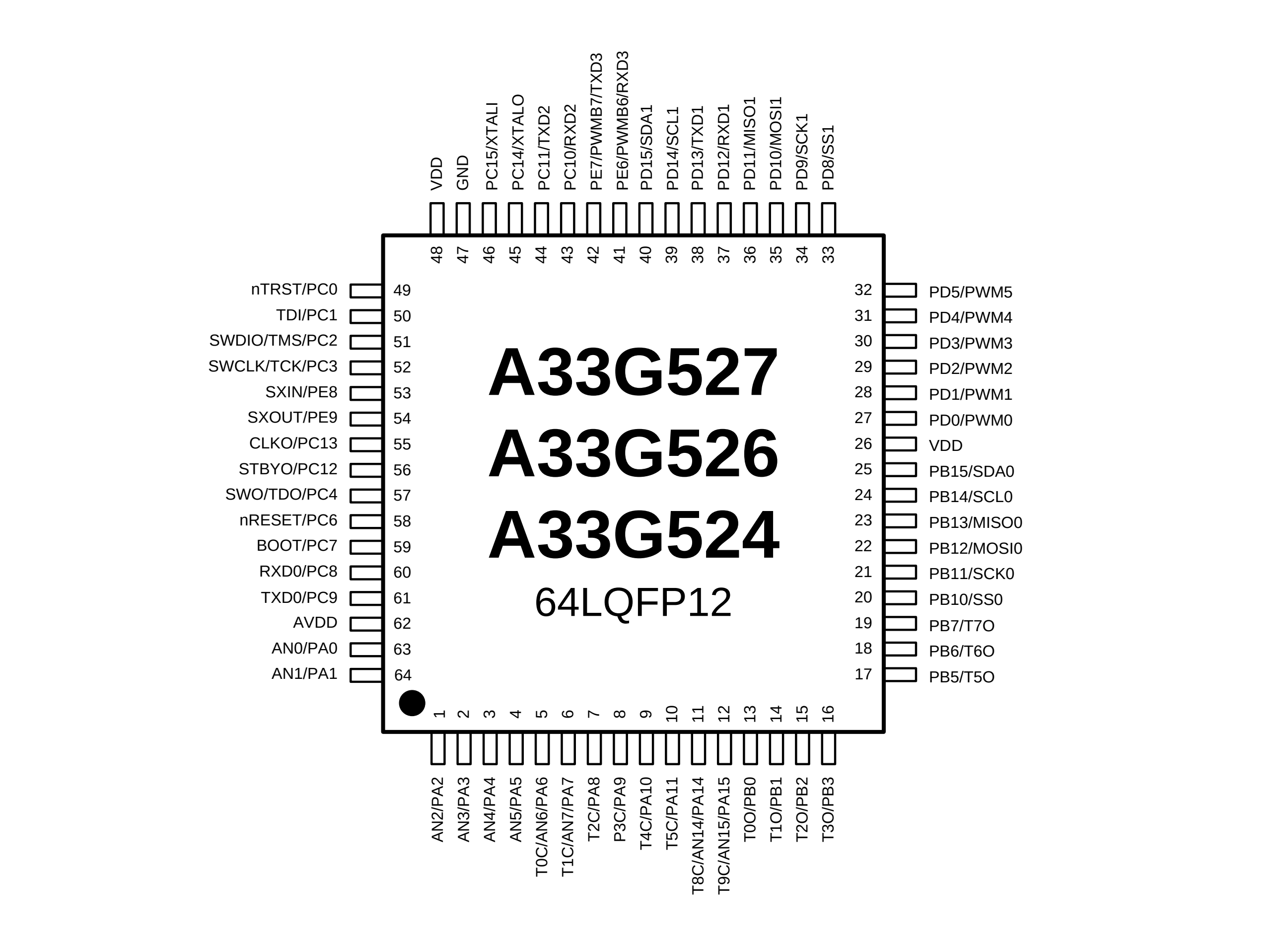


Figure 1.7. Pin layout of . . A33G527RL / A33G526RL / A33G524RL (64LQFP12)

1.5.6 A33G526RM / A33G524RM (64LQFP10)

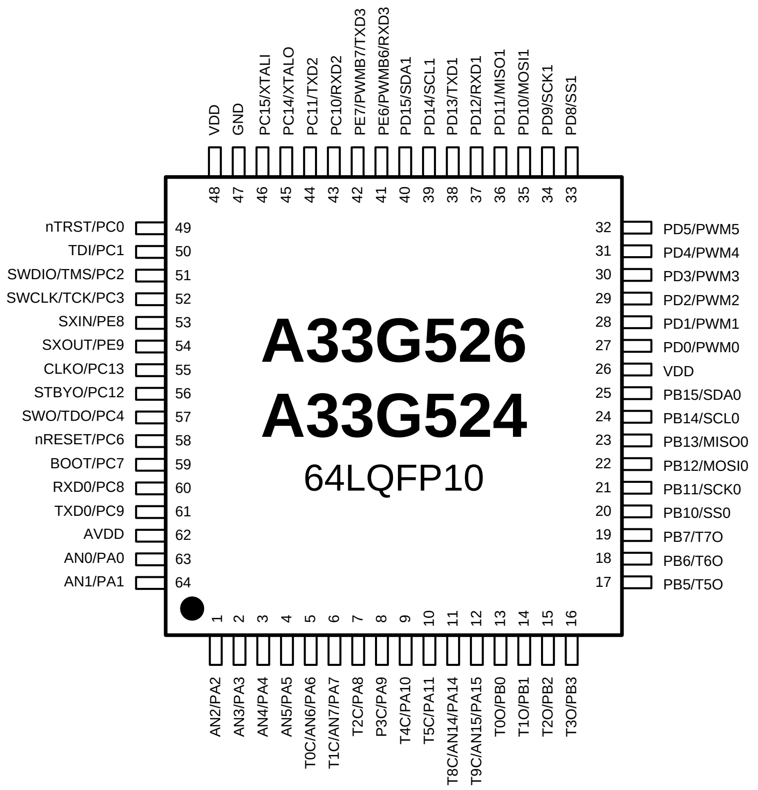


Figure 1.8. Pin layout of . A33G526RM / A33G524RM (60LQFP10)

1.6 Pin Configuration

Depending on the package type of each A33G52x microcontroller, there are some differences in the number of pins and configurations.
The pin configuration of A33G52x MCUs is as follows.

Table 1.3 Pin Configuration

Pin Number				Name	Type	Description	Remark
100LQFP14	100MQFP	80LQFP14 80LQFP12	64LQFP12 64LQFP10				
99	1	79	63	PA0*	IOUS	GPIO A bit	
				AN0	IA	ADC input 0	
100	2	80	64	PA1*	IOUS	GPIO A bit 1	
				AN1	IA	ADC input 1	
1	3	1	1	PA2*	IOUS	GPIO A bit 2 input/output	
				AN2	IA	ADC INPUT 2	
2	4	2	2	PA3*	IOUS	GPIO A bit 3 input/output	
				AN3	IA	ADC INPUT 3	
3	5	3	3	PA4*	IOUS	GPIO A bit 4 input/output	
				AN4	IA	ADC INPUT 4	
4	6	4	4	PA5*	IOUS	GPIO A bit 5 input/output	
				AN5	IA	ADC INPUT 5	
5	7	5	5	PA6*	IOUS	GPIO A bit 6 input/output	
				T0C	IPUS	Timer 0 Clock/Capture input	
				AN6	IA	ADC INPUT 6	
6	8	6	6	PA7*	IOUS	GPIO A bit 7 input/output	
				T1C	IUS	Timer 1 Clock/Capture input	
				AN7	IA	ADC INPUT 7	
7	9	7	-	GND	P	GND	
8	10	8	-	VDD	P	VDD (3.0~ 5.5V)	
9	11	-	-	PF0*	IOUS	GPIO F bit 0 input/output	
				AN8	IA	ADC INPUT 8	
10	12	-	-	PF1*	IOUS	GPIO F bit 1 input/output	
				AN9	IA	ADC INPUT 9	
11	13	-	-	PF2*	IOUS	GPIO F bit 2 input/output	
				AN10	IA	ADC INPUT 10	
12	14	-	-	PF3*	IOUS	GPIO F bit 3 input/output	
				AN11	IA	ADC INPUT 11	
13	15	-	-	PF4*	IOUS	GPIO F bit 4 input/output	
				AN12	IA	ADC INPUT 12	
14	16	-	-	PF5*	IOUS	GPIO F bit 5 input/output	
				AN13	IA	ADC INPUT 13	
15	17	-	-	TEST	IDS	Test mode input (default Pull-down)	
16	18	9	7	PA8*	IOUS	GPIO A bit 8 input/output	
				T2C	IUS	Timer 2 Clock/Capture input	
17	19	10	8	PA9*	IOUS	GPIO A bit 9 input/output	
				T3C	IUS	Timer 3 Clock/Capture input	
18	20	11	9	PA10*	IOUS	GPIO A bit 10 input/output	
				T4C	IUS	Timer 4 Clock/Capture input	
19	21	12	10	PA11*	IOUS	GPIO A bit 11 input/output	
				T5C	IUS	Timer 5 Clock/Capture input	
20	22	13	-	PA12*	IOUS	GPIO A bit 12 input/output	
				T6C	IUS	Timer 6 Clock/Capture input	
21	23	-	-	PA13*	IOUS	GPIO A bit 13 input/output	
				T7C	IUS	Timer 7 Clock/Capture input	
22	24	14	11	PA14*	IOUS	GPIO A bit 14 input/output	
				T8C	IUS	Timer 8 Clock/Capture input	
				AN14	IA	ADC INPUT 14	

23	25	15	12	PA15*	IOUS	GPIO A bit 15 input/output	
				T9C	IUS	Timer 9 Clock/Capture input	
				AN15	IA	ADC INPUT 15	
24	26	16	13	PB0*	IOUS	GPIO B bit 0 input/output	
				T0O	OUS	Timer 0 Output	
25	27	17	14	PB1*	IOUS	GPIO B bit 1 input/output	
				T1O	OUS	Timer 1 Output	
26	28	18	15	PB2*	IOUS	GPIO B bit 2 input/output	
				T2O	OUS	Timer 2 Output	
27	29	19	16	PB3*	IOUS	GPIO B bit 3 input/output	
				T3O	OUS	Timer 3 Output	
28	30	20	-	PB4*	IOUS	GPIO B bit 4 input/output	
				T4O	OUS	Timer 4 Output	
29	31	21	17	PB5*	IOUS	GPIO B bit 5 input/output	
				T5O	OUS	Timer 5 Output	
30	32	22	18	PB6*	IOUS	GPIO B bit 6 input/output	
				T6O	OUS	Timer 6 Output	
31	33	23	19	PB7*	IOUS	GPIO B bit 7 input/output	
				T7O	OUS	Timer 7 Output	
32	34	24	-	PB8*	IOUS	GPIO B bit 8 input/output	
				T8O	OUS	Timer 8 Output	
33	35	25	-	PB9*	IOUS	GPIO B bit 9 input/output	
				T9O	OUS	Timer 9 Output	
34	36	26	20	PB10*	IOUS	GPIO B bit 10 input/output	
				SS0	IOUS	SPI Channel 0 Select Signal input/output	
35	37	27	21	PB11*	IOUS	GPIO B bit 11 input/output	
				SCK0	IOUS	SPI Channel 0 Clock Signal input/output	
36	38	28	22	PB12*	IOUS	GPIO B bit 12 input/output	
				MOSI0	IOUS	SPI Channel 0 Master Out/Slave In Signal	
37	39	29	23	PB13*	IOUS	GPIO B bit 13 input/output	
				MISO0	IOUS	SPI Channel 0 Master In/Slave Out Signal	
38	40	30	24	PB14*	IOUS	GPIO B bit 14 input/output	
				SCL0	OUS	I2C Channel 0 SCL Signal	
39	41	31	25	PB15*	IOUS	GPIO B bit 15 input/output	
				SDA0	OUS	I2C Channel 0 SDA Signal	
40	42	32	-	GND	P	GND	
41	43	33	26	VDD	P	VDD (3.0 ~ 5.5V)	
42	44	34	27	PD0*	IOUC	GPIO D bit 0 input/output	
				PWM0	OUC	PWM Channel 0 Output	
43	45	35	28	PD1*	IOUC	GPIO D bit 1 input/output	
				PWM1	OUC	PWM Channel 1 Output	
44	46	36	29	PD2*	IOUC	GPIO D bit 2 input/output	
				PWM2	OUC	PWM Channel 2 Output	
45	47	37	30	PD3*	IOUC	GPIO D bit 3 input/output	
				PWM3	OUC	PWM Channel 3 Output	
46	48	38	31	PD4*	IOUC	GPIO D bit 4 input/output	
				PWM4	OUC	PWM Channel 4 Output	
47	49	39	32	PD5*	IOUC	GPIO D bit 5 input/output	
				PWM5	OUC	PWM Channel 5 Output	
48	50	40	-	PD6*	IOUC	GPIO D bit 6 input/output	
				PWM6	OUC	PWM Channel 6 Output	
49	51	41	-	PD7*	IOUC	GPIO D bit 7 input/output	
				PWM7	OUC	PWM Channel 7 Output	
50	52	42	33	PD8*	IOUC	GPIO D bit 8 input/output	
				SS1	IOUC	SPI Channel 0 Select Signal	

						input/output	
51	53	43	34	PD9*	IOUC	GPIO D bit 9 input/output	
				SCK1	IOUC	SPIO Channel Clock Signal input/output	
52	54	44	35	PD10*	IOUC	GPIO D bit 10 input/output	
				MOSI1	IOUC	SPI Channel 1 Master Out/Slave In Signal	
53	55	45	36	PD11*	IOUC	GPIO D bit 11 input/output	
				MISO1	IOUC	SPI Channel 1 Master In/Slave Out Signal	
54	56	46	37	PD12*	IOUC	GPIO D bit 12 input/output	
				RXD1	IUC	UART Channel 1 Receive Data input	
55	57	47	38	PD13*	IOUC	GPIO D bit 13 input/output	
				39TXD1	OUC	UART Channel 1 Transmit Data Output	
56	58	48	39	PD14*	IOUC	GPIO D bit 14 input/output	
				SCL1	OUC	I2C Channel 1 SCL Signal	
57	59	49	40	PD15*	IOUC	GPIO D bit 15 input/output	
				SDA1	OUC	I2C Channel 1 SDA Signal	
58	60	-	-	PE0*	IOUS	GPIO E bit 0 input/output	
				PWMB0	OUS	PWM Channel 0 inversion Output	
59	61	50	-	PE1*	IOUS	GPIO E bit 1 input/output	
				PWMB1	OUS	PWM Channel 1 inversion Output	
60	62	51	-	PE2*	IOUS	GPIO E bit 2 input/output	
				PWMB2	OUS	PWM Channel 2 inversion Output	
61	63	52	-	PE3*	IOUS	GPIO E bit 3 input/output	
				PWMB3	OUS	PWM Channel 3 inversion Output	
62	64	53	-	PE4*	IOUS	GPIO E bit 4 input/output	
				PWMB4	OUS	PWM Channel 4 inversion Output	
63	65	54	-	PE5*	IOUS	GPIO E bit 5 input/output	
				PWMB5	OUS	PWM Channel 5 inversion Output	
64	66	55	41	PE6*	IOUS	GPIO E bit 6 input/output	
				PWMB6	OUS	PWM Channel 6 inversion Output	
				RXD3	IUS	UART Channel 3 Receive Data input	
65	67	56	42	PE7*	IOUS	GPIO E bit 7 input/output	
				PWMB7	OUS	PWM Channel 7 inversion Output	
				TXD3	OUS	UART Channel 3 Transmit Data Output	
66	68	57	43	PC10*	IOUS	GPIO C bit 10 input/output	
				RXD2	IUS	UART Channel 2 Receive Data input	
67	69	58	44	PC11*	IOUS	GPIO C bit 11 input/output	
				TXD2	OUS	UART Channel 2 Transmit Data Output	
68	70	-	-	VDD	P	VDD (3.0 ~ 5.5V)	
69	71	59	45	PC14*	IOUS	GPIO C bit 14 input/output	
				XTALO	IA	Main Crystal Oscillator Output (4/8/10MHz)	
70	72	60	46	PC15*	IOUS	GPIO C bit 15 input/output	
				XTALI	IA	Main Crystal Oscillator input (4/8/10MHz)	
71	73	61	47	GND	P	GND	
72	74	-	-	PF7	IOUS	GPIO F bit 7 input/output	
73	75	-	-	PF8	IOUS	GPIO F bit 8 input/output	
74	76	62	48	VDD	P	VDD (3.0 ~ 5.5V)	
75	77	-	-	PF9	IOUS	GPIO F bit 9 input/output	
76	78	-	-	PF10	IOUS	GPIO F bit 10 input/output	
77	79	-	-	PE11*	IOUS	GPIO E bit 11 input/output	
				TraceD3	IOUS	ETM Trace Data 3	
78	80	-	-	PE12*	IOUS	GPIO E bit 12 input/output	

				TraceD2	IOUS	ETM Trace Data 2	
79	81	-	-	PE13*	IOUS	GPIO E bit 13 input/output	
				TraceD1	IOUS	ETM Trace Data 1	
80	82	-	-	PE14*	IOUS	GPIO E bit 14 input/output	
				TraceD0	IOUS	ETM Trace Data 0	
81	83	-	-	PE15*	IOUS	GPIO E bit 15 input/output	
				TraceCLK	IOUS	ETM Trace Clock	
82	84	63	49	PC0	IOUS	GPIO C bit 0 input/output	
				nTRST*	IUS	JTAG nTRST Signal input	
83	85	64	50	PC1	IOUS	GPIO C bit 1 input/output	
				TDI*	IUS	JTAG TDI Signal input	
84	86	65	51	PC2	IOUS	GPIO C bit 2 input/output	
				TMS*	IUS	JTAG TMS Signal input	
				SWDIO	IOUS	SWD data input/output	
85	87	66	52	PC3	IOUS	GPIO C bit 3 input/output	
				TCK*	IUS	JTAG TCK Signal input	
				SWCLK	IUS	SWD Clock signal	
86	88	67	53	PE8*	IOUS	GPIO E bit 8 input/output	
				SXIN	IA	Sub Crystal Oscillator Signal input (32.768kHz)	
87	89	68	54	PE9*	IOUS	GPIO E bit 9 input/output	
				SXOUT	IA	Sub Crystal Oscillator Signal Output (32.768kHz)	
88	90	69	55	PC13*	IOUS	GPIO C bit 13 input/output	
				CLKO	OUS	External Clock Output	
89	91	70	56	PC12*	IOUS	GPIO C bit 12 input/output	
				STBYO	OUS	Stand-by(Power-down) Indication Output Signal	
90	92	-	-	PF11	IOUS	GPIO F bit 11 input/output	
91	93	71	57	PC4	IOUS	GPIO C bit 4 input/output	
				TDO*	OUS	JTAG TDO Signal Output	
				SWO	OUS	SWD Output	
92	94	72	-	PC5	IOUS	GPIO C bit 5 input/output	
93	95	73	58	PC6	IOUS	GPIO C bit 6 input/output	
				nRESET*	IUS	Reset input	
94	96	74	59	PC7	IOUS	GPIO C bit 7 input/output	
				BOOT*	IUS	Boot Signal input (input mode when reset)	
95	97	75	60	PC8	IOUS	GPIO C bit 8 input/output	
				RXD0*	IUS	UART Channel 0 Receive Data input (UART Boot Channel)	
96	98	76	61	PC9	IOUS	GPIO C bit 9 input/output	
				TXD0*	OUS	UART Channel 0 Transmit Data Output (UART Boot Channel)	
97	99	77	-	AGND	P	ADC/Analog GND	
98	100	78	62	AVDD	P	ADC/Analog VDD	

*Legend: I=Input, O=Output, U=Pull-up, D=Pull-down,
S=Schmitt-Trigger Input Type, C=CMOS Input Type
A=Analog, P=Power

(*) Initial setting in reset state In reset state, all pins except for certain pin will be Hi-Z state.

1.7 A33G52x Memory Map

1.7.1 Overall Memory Map

The memory configuration of A33G52x MCUs are as follows. The code flash area is assigned differently depending on the device name.

- A33G527 (384KB)
- A33G526 (256KB)
- A33G524 (128KB)

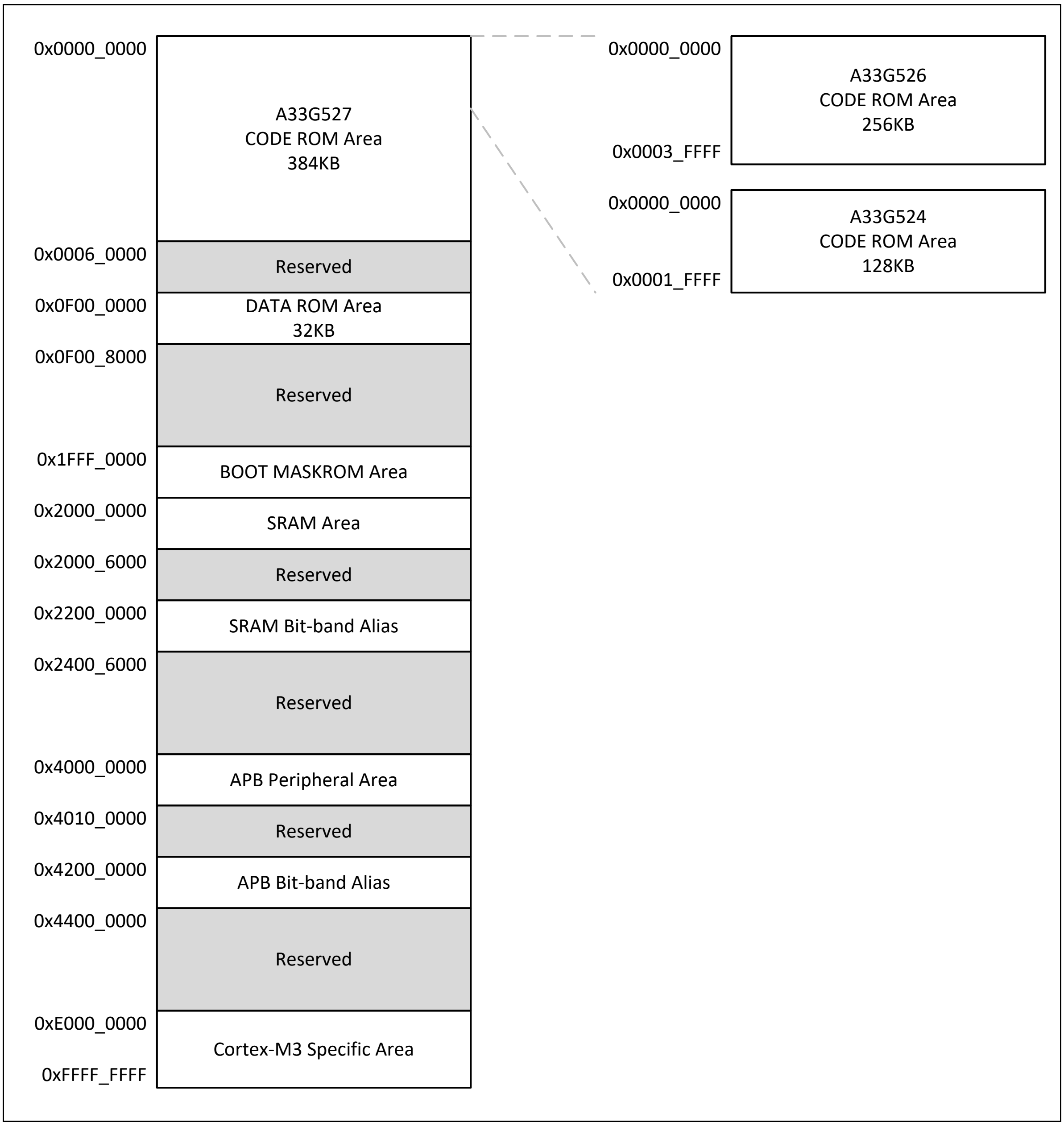


Figure 1.9. Overall Memory Map

1.7.2 Memory Map of Cortex-M3

The A33G52x has a fixed memory area for Cortex-M3 from 0xE000_0000 to 0xFFFF_FFFF, and the sections are divided according to the interfaces to the internal and private peripheral buses (PPB) supported by this CPU.

- Interfaces accessible to internal-only peripheral buses
  - Instrumentation Trace Macrocell (ITM)
  - Data Watchpoint and Trace (DWT)
  - Flashpatch and Breakpoint (FPB)
  - System Control Space (SCS) : Memory Protection Unit (MPU) + Nested Vectored Interrupt Controller (NVIC).
- Interfaces accessible to external dedicated peripheral buses
  - Trace Point Interface Unit (TPIU)
  - Embedded Trace Macrocell (ETM)
  - ROM table.
  - Implementation-specific areas of the PPB memory map.

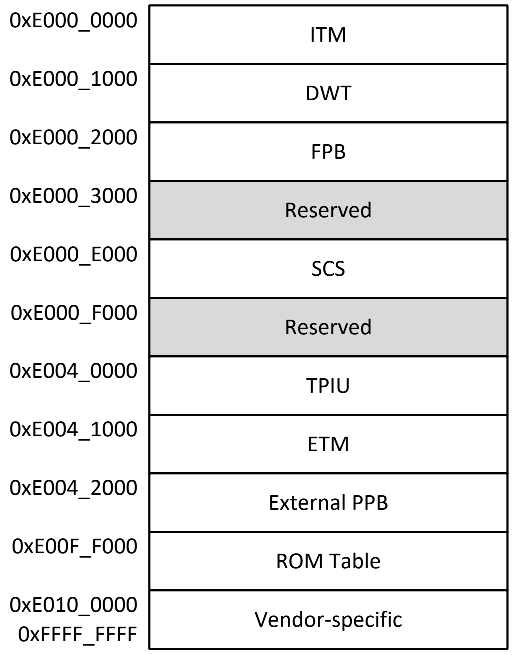


Figure 1.10. Cortex-M3 dedicated Memory Map

1.7.3 Memory map of internal peripehrals

The APB Peripheral Area, which is the total memory map of the A33G52x, is assigned to each peripheral area of the MCU. In this area, special function registers (SFRs) are provided for controlling peripheral devices. For information on the SFR for each peripheral device, refer to the register map in the A33G52x user’s manual.

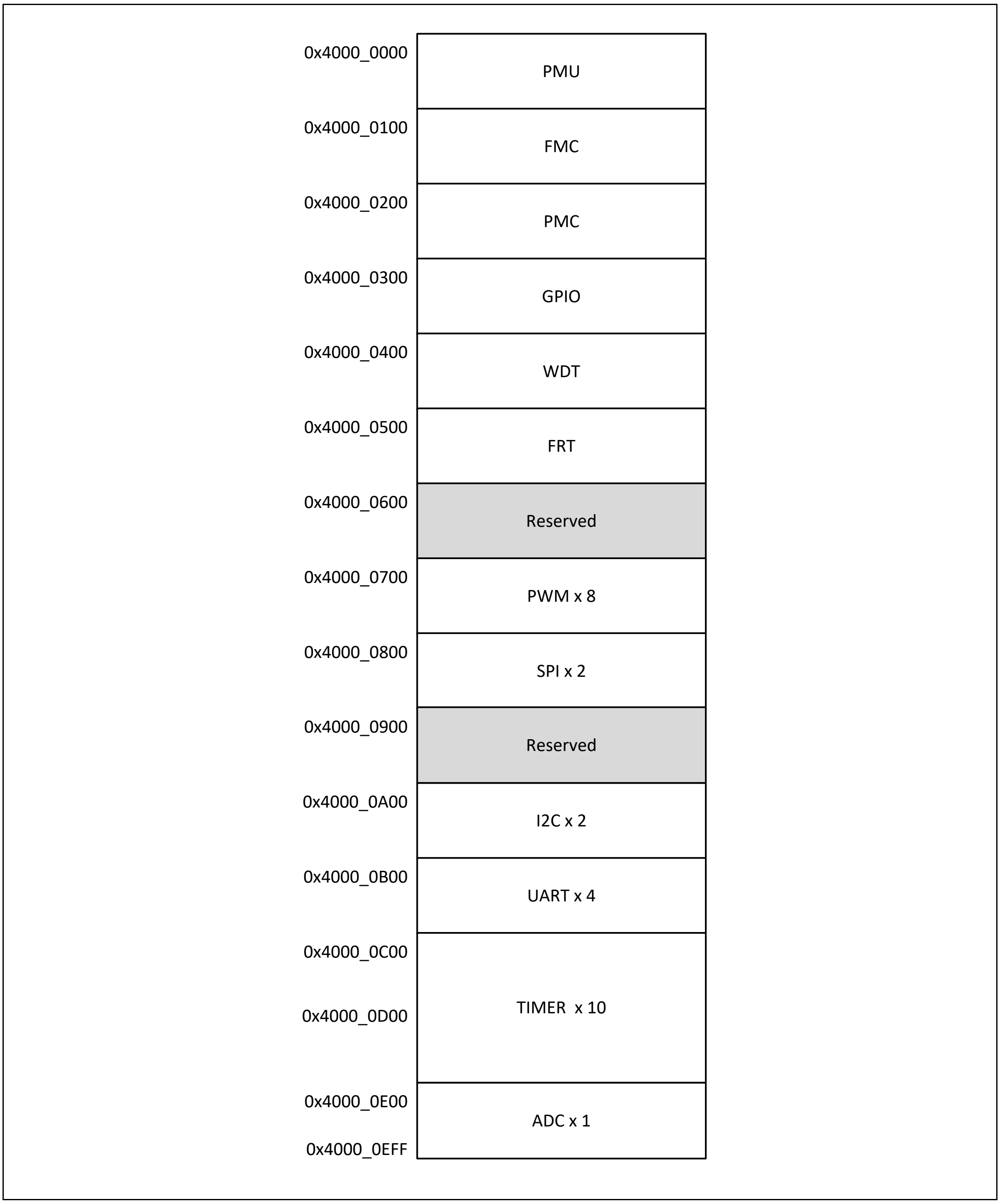


Figure 1.11. Memory Map of Internal Peripherals

CHAPTER 2. CORE

2.1 Cortex-M3

The A33G52x series is based on the Cortex-M3 CPU core, ARM's high-performance 32-bit MCU core. The Cortex-M3 is a processor designed for the harvard architecture and supporting the Thumb-2 instruction set.
For more information, please refer to the ARM document number “DDI337”

2.2 Interrupt Controller

A33G52x has the Cortex-M3's NVIC (Nested-Vectored Interrupt Controller). NVIC is designed to handle interrupts simultaneously in the CPU more effectively. The NVIC built into the Cortex-M3 makes interrupt processing faster and more efficient.

IRQ interrupts are disabled when the A33G52x system is initialized. To use the various interrupts supported by the A33G52x, you must enable the IRQ interrupt. Even if the peripheral's interrupt function is enabled, peripheral interrupts will NOT occur if the IRQ interrupt is NOT enabled.

The interrupt vector map of A33G52x is defined as follows:

Table 2.1. 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	MPU Fault Handler
-11	0x0000_0014	BUS Fault Handler
-10	0x0000_0018	Usage Fault Handler
-9	0x0000_001C	Reserved
-8	0x0000_0020	
-7	0x0000_0024	
-6	0x0000_0028	
-5	0x0000_002C	SVCall Handler
-4	0x0000_0030	Debug Monitor Handler
-3	0x0000_0034	Reserved
-2	0x0000_0038	PenSV Handler
-1	0x0000_003C	SysTick Handler
0	0x0000_0040	LVDFAIL
1	0x0000_0044	MXOSCFAIL
2	0x0000_0048	Reserved
3	0x0000_004C	WDT
4	0x0000_0050	FRT
5	0x0000_0054	TIMER0
6	0x0000_0058	TIMER1
7	0x0000_005C	TIMER2
8	0x0000_0060	TIMER3
9	0x0000_0064	TIMER4
10	0x0000_0068	TIMER5
11	0x0000_006C	TIMER6
12	0x0000_0070	TIMER7
13	0x0000_0074	TIMER8
14	0x0000_0078	TIMER9
15	0x0000_007C	MCKFAIL
16	0x0000_0080	GPIOA
17	0x0000_0084	GPIOB
18	0x0000_0088	GPIOC
19	0x0000_008C	GPIOD
20	0x0000_0090	GPIOE
21	0x0000_0094	GPIOF
22	0x0000_0098	Reserved
23	0x0000_009C	
24	0x0000_00A0	PWM0
25	0x0000_00A4	PWM1
26	0x0000_00A8	PWM2
27	0x0000_00AC	PWM3
28	0x0000_00B0	PWM4
29	0x0000_00B4	PWM5

30	0x0000_00B8	PWM6
31	0x0000_00BC	PWM7
32	0x0000_00C0	SPI0
33	0x0000_00C4	SPI1
34	0x0000_00C8	Reserved
35	0x0000_00CC	
36	0x0000_00D0	I2C0
37	0x0000_00D4	I2C1
38	0x0000_00D8	UART0
39	0x0000_00DC	UART1
40	0x0000_00E0	UART2
41	0x0000_00E4	UART3
42	0x0000_00E8	Reserved
43	0x0000_00EC	ADC
44	0x0000_00F0	Reserved
45	0x0000_00F4	
46	0x0000_00F8	
47	0x0000_00FC	
48	0x0000_0100	
49	0x0000_0104	
50	0x0000_0108	
51	0x0000_010C	
52	0x0000_0110	
53	0x0000_0114	
54	0x0000_0118	
55	0x0000_011C	
56	0x0000_0120	
57	0x0000_0124	
58	0x0000_0128	
59	0x0000_012C	
60	0x0000_0130	
61	0x0000_0134	
62	0x0000_0138	
63	0x0000_013C	

CHAPTER 3. PMU (Power Management Unit)

Table 3.1. Operation Summary

Item	Pin Name	Remark
Clock usage	Main XTAL, Sub XTAL RingOSC, IOSC16 HCLK, PCLK	Clock settings
Reset Source	External Reset Soft-Reset LVD (Low Voltage Detector) Clock Fail	PMU_CFGR LVD_CON PMU_CMR
Reset Generation	External Reset Soft-Reset LVD Clock Fail	PMU_CFGR LVD_CON PMU_CMR

Interrupt Generation	LVD(0)
	MXOSCFAIL(1)
		LVD_CON
		PMU_CMR

Interrupt Clear Method	Writing ‘1’ to the interrupt bit	PMU_CMR, LVD_CON

3.1 Overview

PMU (Power Managament Unit), the power management module of A33G52x MCU, can control and monitor the source clock of the chip, and set system operation speed. In addition, the power consumption of the application can be controlled by controlling the operation mode of the MCU. There is also a function to prevent malfunction of user application by setting reset.
It also provides the function to enable the peripherals' clocks and devices in the Power Managament Unit (PMU) block to enable control of each peripheral device in the MCU.

Features of PMU (Power Managament Unit)

- MCU operating mode for low power consumption
  - Run mode (Run)
  - Sleep mode (Idle)
  - Deep-sleep mode (Power Down)
- Input clock sources for MCU operation
  - 1MHz internal ring oscillator (RINGOSC)
  - 16MHz internal oscillator (IOSC16)
  - 4/8/10MHz external main oscillator (MXOSC)
  - 32.768kHz external auxiliary oscillator (SXOSC)
- PLL frequency output
  - High-speed clock driving up to 75MHz frequency output
  - Fine adjustment in 1MHz unit
- Clock monitoring and non-oscillation error check
  - Main clock and external main oscillator clock monitoring
  - No-oscillation error handling function
- Reset source and wake-up event
- Others
  - LDO (Low Drop Out)
  - POR (Power On Reset)
  - LVD (Low Level Detector)

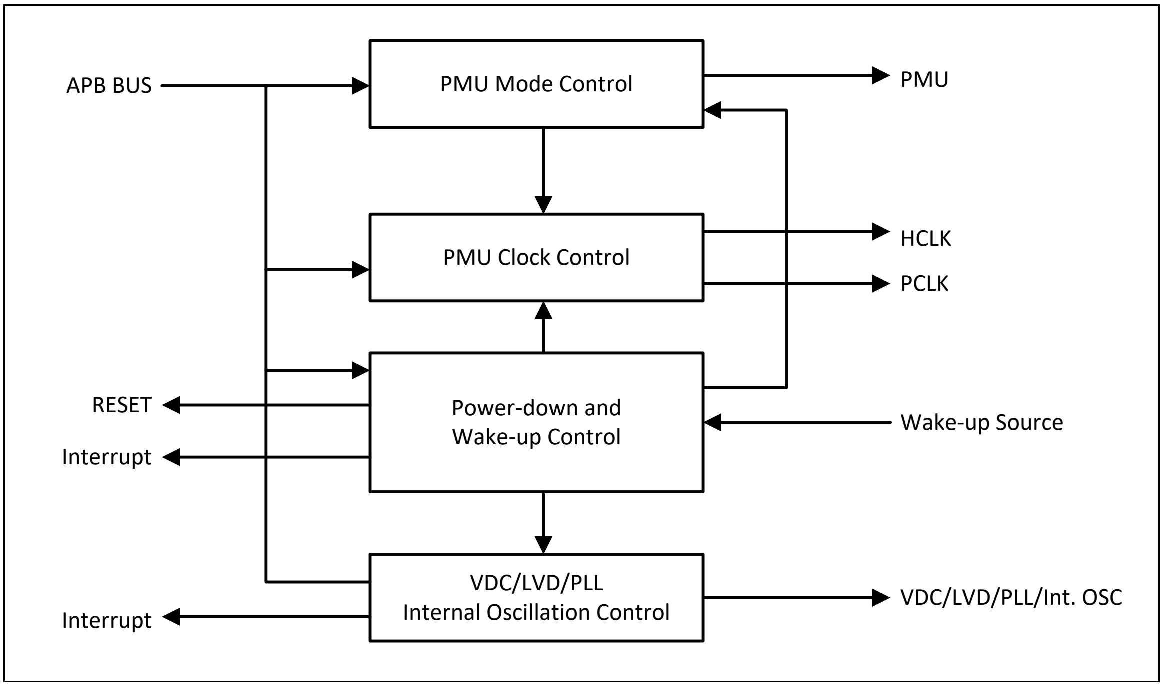


Figure 3.1. Block diagram of PMU

3.2 Pin configuration and clock sources

3.2.1 Pin Configurations

The Power Management Unit (PMU) of the A33G52x has input / output pins that can be set.
The input signals that can be set in the PMU include reset input (nRESET), main oscillator input (XTALI / XTALO), and auxiliary oscillator input (SXIN / SXOUT).
The output signals that can be set by the PMU include the STBYO pin indicating the power down state and the CLKO pin for dividing the internal PLL or main clock.

Table 3.2. Power Management Unit (PMU) and PLL related pins.

Pin	I/O	Description
nRESET	I	Initialize entire chip by the external reset input pin. This pin is Schmitt-Trigger type input.
XTALI/XTALO	OSC	Main crystal oscillator pin
SXIN/SXOUT	OSC	32kHz crystal sub-oscillatorp pin
STBYO	O	A indicator pin of Stand-by mode
CLKO	O	Clock output pin of internal PLL

3.2.2 Clock Sources

A ring oscillator (RINGOSC) of about 1MHz and a 16MHz RC oscillator (IOSC16) are built in the A33G52x MCUs. The external main crystal (MXOSC) oscillation can be used as a clock input, and the clock can be used to 75MHz using the internal PLL. In addition, the external clock 32768kHz crystal (SXOSC) oscillation as an auxiliary clock, more time-critical periodic time calculation is possible.
The A33G52x supports clock sources as shown below.

Table 3.3. The Clock Sources

Clock Source	Frequency	Description
RINGOSC	1MHz (±50%)	Clock monitoring for internal system Used for WDT, Clock monitoring (Large variations in temperature and voltage)
IOSC16	16MHz (±3%)	Intenal main clock (This clock can be used instead of XTAL)
MXOSC (MainOSC)	X-TAL(4MHz~10MHz)	External Main clock
SXOSC (SubOSC)	SX-TAL(32.768kHz)	External auxiliary clock (for real time clock)
PLL	8MHz ~ 75MHz	multiplier and divider of PLL clock High frequency output XTAL or IOSC16 input as clock source

3.3 Block Diagram

3.3.1 PMU Clock

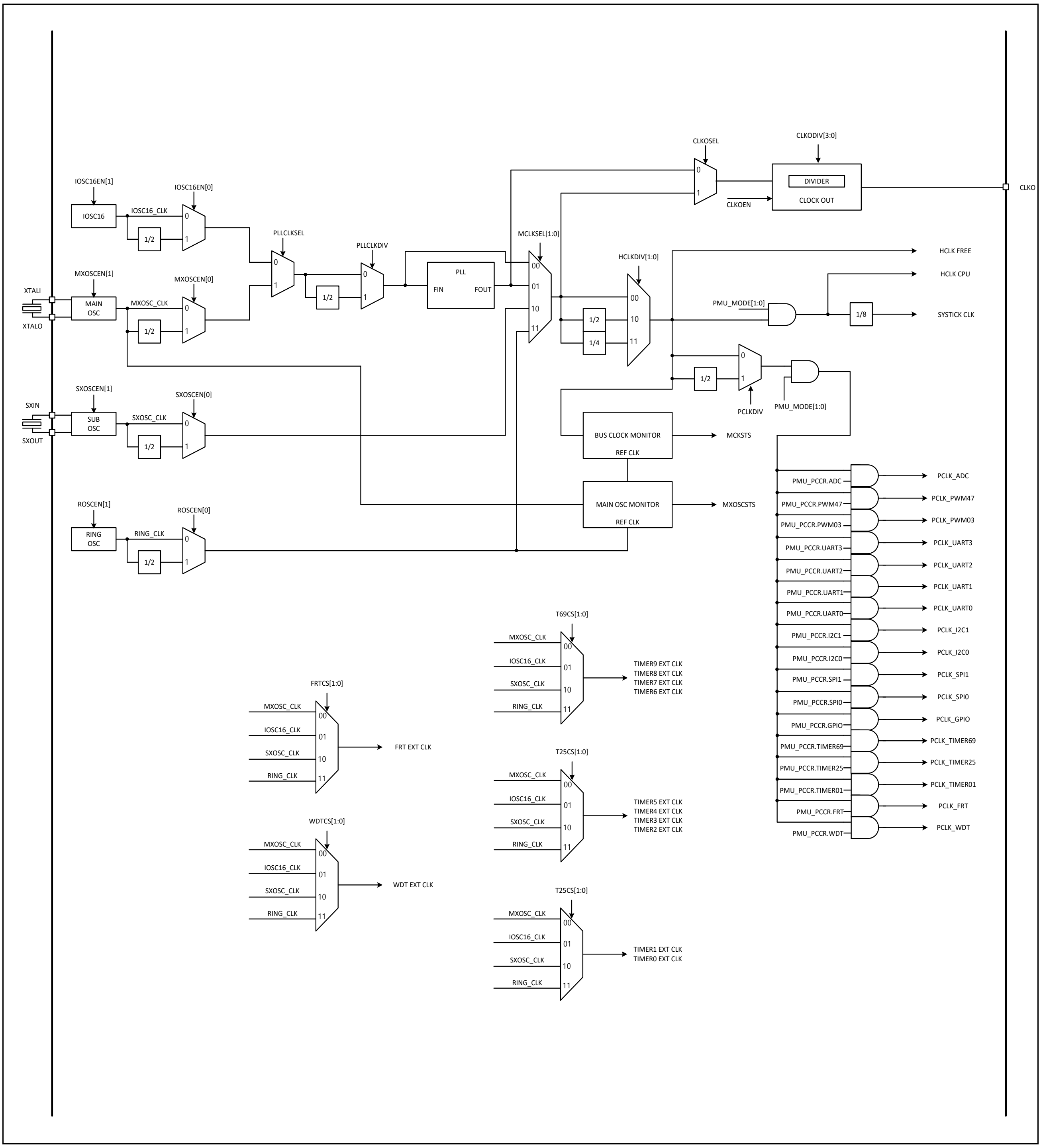


Figure 3.2. Block diagram of A33G52x PMU Clock

3.3.2 PMU Reset

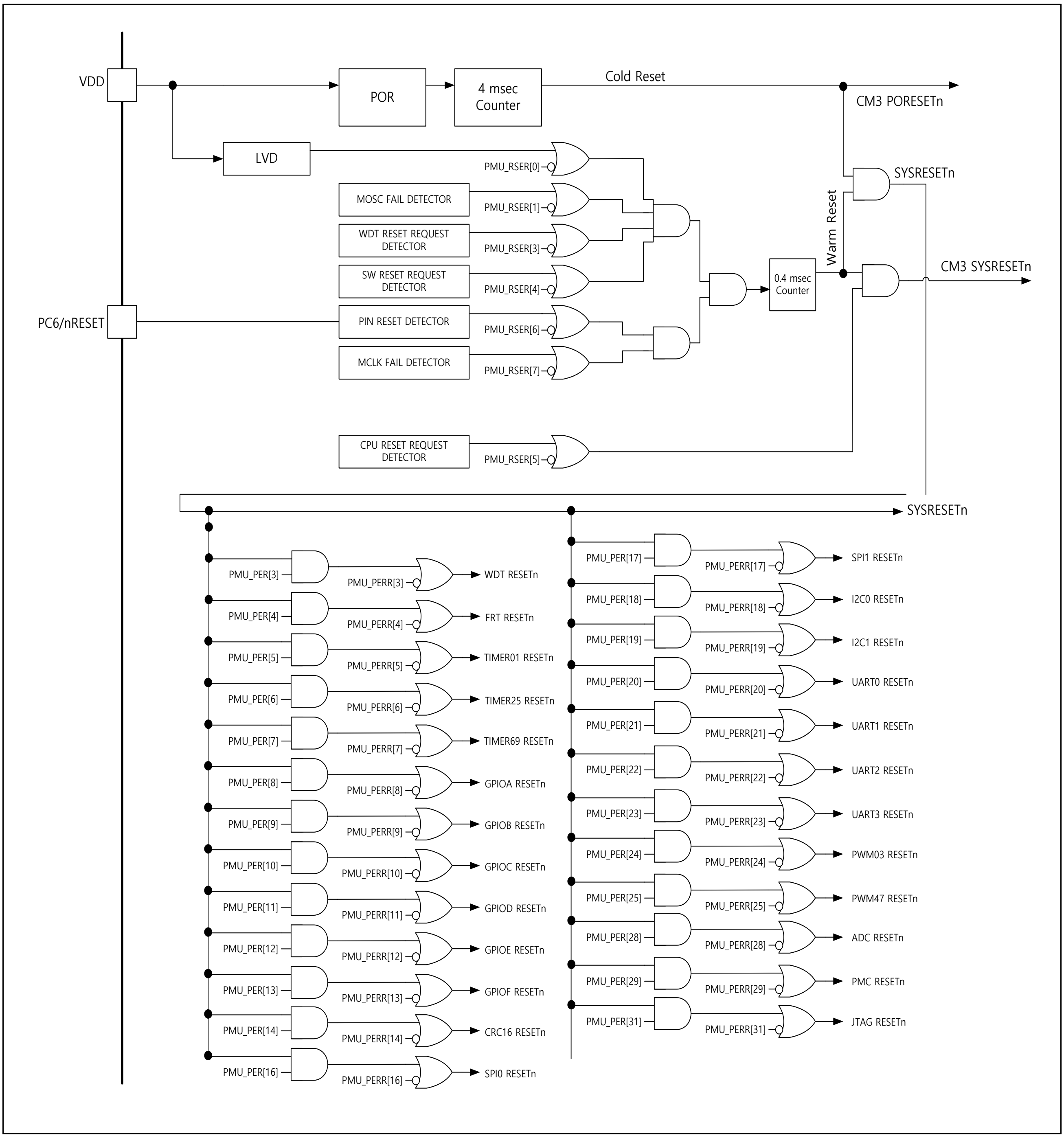


Figure 3.3. Block diagram of A33G52x PMU Reset

3.4 Register Map

The base address of PMU block is 0x4000_0000, the register list is shown as follows.

Table 3.4. The Registar List of the PMU

Register Name	Offset	Access Type	Description	Initial Value	Ref
PMU_IDR	0x00	RO	PMU ID Register	0xCEDA0000	3.5.1
PMU_MR	0x04	RW	PMU Mode Register	0x00000000	3.5.2
PMU_CFGR	0x08	RW	PMU Configuration Register	0x00000000	3.5.3
PMU_WSER	0x10	RW	PMU Wake-up Source Enable Register	0x00000000	3.5.4
PMU_WSSR	0x14	RO	PMU Wake-up Source Status Register	0x00000000	3.5.5
PMU_RSER	0x18	RW	PMU Reset Source Enable Register	0x00000069	3.5.6
PMU_RSSR	0x1C	RW	PMU Reset Source Status Register	0x00000040*	3.5.7
PMU_PERR	0x20	RW	PMU Peripheral Event Reset Register	0xB3FF3FF8	3.5.8
PMU_PER	0x24	RW	PMU Peripheral Enable Register	0xB3FF3FF8	3.5.9
PMU_PCCR	0x28	RW	PMU Peripheral Clock Control Register	0x00000118	3.5.10
PMU_CCR	0x30	RW	PMU Clock Control Register	0x00000080	3.5.11
PMU_CMR	0x34	RW	PMU Clock Monitoring Register	0x00000000	3.5.12
PMU_MCMR	0x38	RW	PMU Main Clock Monitoring Register	0x00000000	3.5.13
PMU_BCCR	0x3C	RW	PMU Bus Clock Control Register	0x00000000	3.5.14
PMU_PCSR	0x40	RW	PMU Peripheral Clock Select Register	0x00000000	3.5.15
PMU_COR	0x44	RW	PMU Clock Output Register	0x00000000	3.5.16
PMU_PLLCON	0x50	RW	PLL Control Register	0x00000000	3.5.17
PMU_LVDCON	0x54	RW	LVD (Low Voltage Detector) Control Register	0x00008800	3.5.18
PMU_VDCCON	0x58	RW	VDC/LVD Trimming Register	0xXXXXXXXX	3.5.19
PMU_IOSC16TRIM	0x5C	RW	Internal Oscillator (16MHz) Trimming Register	0x0000XXXX	3.5.20
PMU_EOSCCON	0x60	RW	Main Oscillator (XTAL) Control Register	0x00000001	3.5.21
PMU_EXTMODER	0x70	RW	External Mode Pin Read Register	0x00000000	3.5.22

Note) Marked with an asterisk (*) are reset POR reset only applies to register.

The base address of the device ID block of the A33G52x is 0x4000_F000, and the register map is as shown in the table below.

Table 3.5. The Registar List of the PMU ID Register Map

Register Name	Offset	Access Type	Description	Initial Value
VENDIDR	0x00	R	Vendor ID Register (ABOV)	0x41424F56
CHIPIDR	0x04	R	CHIP ID Register	0x4D33A000
REVIDR	0x08	R	REV. ID Register	0x00000000

3.5 Register Description

3.5.1 PMU_IDR PMU ID Register

PMUIDR is a 32-bit read-only register stores the product ID number and revision number. The revision number is CHIPREV value at the least significant byte of this register.

PMU_IDR=0x4000_0000

31	30	29	28	27	26	25	24	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																								CHIPREV							
0xCEDA00																								00							
RO																								RO							

31 **8**	**CHIPID**	The value of the A33G52x
7 **0**	**CHIPREV**	The value of the A33G52x revision

3.5.2 PMU_MR PMU Mode Register

PMU_MR is a register that can display the previous operation mode of the MCU or set the current operation mode. For example, to enter power-down mode using the FRT as a wake-up source, ECLKMD bit must be set to "1". For details on how to use the MCU operation mode, refer to. 3.7.1

PMU_MR=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																									PVDCLP	PREVMODE		ECLKMD	VDCLP	PMUMODE	
-																									0	00		0	0	00	
-																									RO	RO		RW	RW	RW	

31 7	-		
6	PVDCLP	VDC low power status of the previous state	
		0	VDC previous state is Normal mode before power-down
		1	VDC previous state is Low-power mode before power-down
5 4	PREVMODE	The previous MCU status before reset or Wake-up	
		00	RUN mode
		01	Sleep mode
		10	Power-down mode
		11	Initialization mode
3	ECLKMD	In Power-down mode, External Clock Oscillator(Main/Sub) is automatically OFF or set to allow user setting.	
		0	MOSC/SOSC automatically OFF in Power-down mode
		1	MOSC/SOSC User Setting in Power-down mode
2	VDCLP	Power-down mode, VDC switched to low-power	
		0	Normal VDC mode in power-down
		1	Low power VDC mode in power-down
1 0	PMUMODE	Indicates the current operating mode of the MCU, that the value of this mode when you write. However, writing '11' is prohibited.	
		00	RUN mode
		01	Sleep mode
		10	Power-down mode
		11	Intialization mode (Writing prohibited)

Note) To enter the power-down mode, the main clock must be set before the RingOSC. (BCCR = 0x00)

3.5.3 PMU_CFGR PMU Configuration Register

This register supports the software reset function and the STBYO output polarity selection feature.

PMU_CFGR=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
Reserved																											STBYOP	Reserved	Reserved	Reserved	SWRST
-																											0	-	-	-	0
-																											RW	-	-	-	WO

31 **5**	**-**		
4	**STBYOP**	Output polarity of STBY pin (except for sleep mode)	
		0	Low-Active output in power-down
		1	High-Active output in power-down
0	**SOFTRST** **(WARMRESET)**	Setting for internal soft-reset	
		0	Normal operation
		1	Internal Soft-reset (Auto-clear)

3.5.4 PMU_WSER PMU Wake-up Source Enable Register

PMU_WSER provides a wake-up function that allows the A33G52x to change its operation to the Run state after entering the sleep mode or power-down mode. You can choose GPIOA ~ F, FRT, WDT, main oscillator, LVD as wakeup source. To wake up using a specific wake-up source, the wake-up source bit must be set to '1'.

PMU_WSER=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																					GPIOFE	GPIOEE	GPIODE	GPIOCE	GPIOBE	GPIOAE	FRTE	WDTE	Reserved	MXFAILE	LVDE
-																					0	0	0	0	0	0	0	0	-	0	0
-																					RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW

31 **11**	**-**	
10	**GPIOFE**	Setting of the wake-up by GPIOF (PD/SLEEP)
9	**GPIOEE**	Setting of the wake-up by GPIOE (PD/SLEEP)
8	**GPIODE**	Setting of the wake-up by GPIOD (PD/SLEEP)
7	**GPIOCE**	Setting of the wake-up by GPIOC (PD/SLEEP)
6	**GPIOBE**	Setting of the wake-up by GPIOB (PD/SLEEP)
5	**GPIOAE**	Setting of the wake-up by GPIO (PD/SLEEP)
4	**FRTE**	Setting of the wake-up by FRT (PD/SLEEP)
3	**WDTE**	Setting of the wake-up by WDT (PD/SLEEP)
2	**-**	
1	**MXFAILE**	Setting of the wake-up by Main OSC-fail event (SLEEP) Setting value of the wake-up by Sub OSC Fail Event (SLEEP)
0	**LVDE**	Setting of the wake-up by GPIOF (PD/SLEEP)

3.5.5 PMU_WSSR PMU Wak-up Source Status Register

This register shows what was the event source of the recent wake-up condition. When a wake-up event occurs, relevant bit will be '1'. The condition will be cleared to '0' by writing '1' to the bit or by clearing the event.

PMU_WSSR=0x4000_0014

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">Reserved</td><td>GPIOF</td><td>GPIOE</td><td>GPIOD</td><td>GPIOC</td><td>GPIOB</td><td>GPIOA</td><td>FRT</td><td>WDT</td><td>Reserved</td><td>MXFAIL</td><td>LVD</td></tr>
<tr><td colspan="21">-</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-</td><td>0</td><td>0</td></tr>
<tr><td colspan="21">-</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>RO</td><td>-</td><td>RO</td><td>RO</td></tr>
</table>

31 **11**	**-**	
10	**GPIOF**	Status flag of the wake-up by GPIOF
9	**GPIOE**	Status flag of the wake-up by GPIOE
8	**GPIOD**	Status flag of the wake-up by GPIOD
7	**GPIOC**	Status flag of the wake-up by GPIOC
6	**GPIOB**	Status flag of the wake-up by GPIOB
5	**GPIOA**	Status flag of the wake-up by GPIOA
4	**FRT**	Status flag of the wake-up by FRT
3	**WDT**	Status flag of the wake-up by WDT
2	**-**	
1	**MXFAIL**	Status flag of the wake-up by Main OSC Fail Event
0	**LVD**	Status flag of the wake-up by LVD

3.5.6 PMU_RSER PMU Reser Source Enable Register

PMU_RSER is a register to set the MCU reset event. When the bit of the corresponding reset source is set to 1, the MCU is reset when an event condition occurs.

PMU_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																								MCKFAILRSTE	RSTRSTE	SYSRSTE	SWRSTE	WDTRSTE	Reserved	MXFAILRSTE	LVDRSTE
-																								0	1	1	0	1	-	0	1
-																								RW	RW	RW	RW	RW	-	RW	RW

31 **8**	**-**	
7	**MCKFAILRSTE**	Enable/Disable main clock oscillation failure reset
6	**RSTRSTE**	Enable/Disable of external nRESET pin reset
5	**SYSRSTE**	Enable/Disable of system reset from the core
4	**SWRSTE**	Enable/Disable of software reset in PMU_CFGR[0]
3	**WDTRSTE**	Enable/Disable of WDT reset
2	**-**	
1	**MXFAILRSTE**	Enable/Disable of external main oscillator (XTAL) failure reset
0	**LVDRSTE**	Enable/Disable of LVD reset

3.5.7 PMU_RSSR PMU Reset Source Status Register

The PMU_RSSR register is a register that informs the source of the reset event when the PMU (Power Management Unit) has reset by an event reset. This register is set to '1' when an event of the corresponding reset source occurs, and cleared when the bit is set to '0'.

PMU_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																								MCKFAILRST	RSTRST	SYSRST	SWRST	WDTRST	Reserved	MXFAILRST	LVDRST
-																								0	0	0	0	0	-	0	0
-																								RW	RW	RW	RW	RW	-	RW	RW

31 **8**	**-**	
7	**MCKFAILRST**	Occurrence of main clock oscillation failure reset
6	**RSTRST**	Occurrence of external nRESET pin reset
5	**SYSRST**	Occurrence of system reset from the core
4	**SWRST**	Occurrence of software reset in PMU_CFGR[0]
3	**WDTRST**	Occurrence of WDT reset
2	**-**	
1	**MXFAILRST**	Occurrence of external main oscillator (XTAL) failure reset
0	**LVDRST**	Occurrence of LVD reset

3.5.8 PMU_PERR PMU Peripheral Event Reset Register

The PMU_PERR register is used by the power management unit (PMU) to reset each peripheral in the MCU when an event reset occurs. An event reset can determine whether to reset each peripheral device. If the peripheral bit is set to '1', the peripheral device is reset when a reset event occurs. If set to '0', the previous value will be maintained even if the peripheral device is NOT reset and a reset event occurs.

PMU_PERR=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
JTAG	Reserved	PMC	ADC	Reserved	Reserved	PWM47	PWM03	UART3	UART2	UART1	UART0	I2C1	I2C0	SPI1	SPI0	Reserved	CRC16	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	TIMER69	TIMER25	TIMER01	FRT	WDT	Reserved	Reserved	Reserved
1	-	1	1	-	-	1	1	1	1	1	1	1	1	1	1	-	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
RW	-	RW	RW	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-	-

31	JTAGEN	Enable JTAG reset
30	-	
29	PMC	Enable PMC (Port Map Controller) reset
28	ADC	Enable ADC reset
27 26	-	
25	PWM47	Enable PWM4, 5,6,7 reset
24	PWM03	Enable PWM0, 1,2,3 reset
23	UART3	Enable UART3 reset
22	UART2	Enable UART2 reset
21	UART1	Enable UART1 reset
20	UART0	Enable UART0 reset
19	I2C1	Enable I2C1 reset
18	I2C0	Enable I2C0 reset
17	SPI1	Enable SPI1 reset
16	SPI0	Enable SPI0 reset
15	-	
14	CRC16	Enable GPIOF reset
13	GPIOF	Enable GPIOE reset
12	GPIOE	Enable GPIOD reset
11	GPIOD	Enable GPIOC reset
10	GPIOC	Enable GPIOB reset
9	GPIOB	Enable GPIOA reset
8	GPIOA	Enable TIMER6, 7,8,9 reset
7	TIMER69	Enable TIMER2, 3,4,5 reset
6	TIMER25	Enable GPIOF reset
5	TIMER01	Enable TIMER0, 1 reset
4	FRT	Enble Free-run Timer reset
3	WDT	Enabled Watchdog Timer reset

2 0	-

3.5.9 PMU_PER PMU Peripheral Enable Register

The PMU_PER register is used to set whether to use each peripheral device of the MCU. When a specific bit is set to '1', the peripherals of that bit are enabled. When a speific bit is set to '0', the bit periphery is disabled and set to the Stop (Reset) state. To enable peripherals required by the application, you must set to '1' the peripheral bits in the PMU_PER and PMU_PCCR registers.

PMU_PER=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
JTAG	Reserved	PMC	ADC	Reserved	Reserved	PWM47	PWM03	UART3	UART2	UART1	UART0	I2C1	I2C0	SPI1	SPI0	Reserved	CRC16	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	TIMER69	TIMER25	TIMER01	FRT	WDT	Reserved	Reserved	Reserved
1	-	1	1	-	-	1	1	1	1	1	1	1	1	1	1	-	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
RW	-	RW	RW	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	-	-

31	JTAGEN	JTAG enable / reset
30	-	
29	PMC	Enable/Reset PMC
28	ADC	Enable/Reset ADC
27 26	-	
25	PWM47	Enable/Reset PWM4, 5,6,7
24	PWM03	Enable/Reset PWM0, 1,2,3
23	UART3	Enable/Reset UART3
22	UART2	Enable/Reset UART2
21	UART1	Enable/Reset UART1
20	UART0	Enable/Reset UART0
19	I2C1	Enable/Reset I2C1
18	I2C0	Enable/Reset I2C0
17	SPI1	Enable/Reset SPI1
16	SPI0	Enable/Reset SPI0
15	-	
14	CRC16	Enable/Reset CRC16
13	GPIOF	Enable/Reset GPIOF, PMC_F
12	GPIOE	Enable/Reset GPIOE, PMC_E
11	GPIOD	Enable/Reset GPIOD, PMC_D
10	GPIOC	Enable/Reset GPIOC, PMC_C
9	GPIOB	Enable/Reset GPIOB, PMC_B
8	GPIOA	Enable/Reset GPIOA, PMC_A
7	TIMER69	Enable/Reset TIMER6, 7,8,9
6	TIMER25	Enable/Reset TIMER2, 3,4,5
5	TIMER01	Enable/Reset TIMER0, 1
4	FRT	Enable/Reset Free-run Timer
3	WDT	Enable/Reset Watch-dog Timer
2 0	-	

3.5.10 PMU_PCCR PMU Peripheral Clock Control Register

The PMU_PCCR register is a register that sets the clock supply to the internal peripheral device. When the bit is set to 1, the peripheral device is clocked. Writing a 0 to the bit will stop the clock supply to the device and cause the device to stop functioning.

To use the peripherals required by your application, you must set the bits of the peripheral device to '1' in the PMU_PER register and the PMU_PCCR register.

Minimizing the clock supply by deactivating the clocks of peripheral devices NOT used by user applications can reduce the MCU current consumption.

PMU_PCCR=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
Reserved			ADC	Reserved		PWM47	PWM03	UART3	UART2	UART1	UART0	I2C1	I2C0	SPI1	SPI0	Reserved	CRC16	Reserved					GPIO	TIMER69	TIMER25	TIMER01	FRT	WDT	Reserved		
-			0	-		0	0	0	0	0	0	0	0	0	0	-	0	-					1	0	0	0	1	1	-		
-			RW	-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	-	RW	-					RW	RW	RW	RW	RW	RW	-		

31 29	-	
28	**ADC**	Enable/Disable ADC clock
27 26	-	
25	**PWM47**	Enable/Disable PWM4, 5,6,7 clock
24	**PWM03**	Enable/Disable PWM0, 1,2,3 clock
23	**UART3**	Enable/Disable UART3 clock
22	**UART2**	Enable/Disable UART2 clock
21	**UART1**	Enable/Disable UART1 clock
20	**UART0**	Enable/Disable UART0 clock
19	**I2C1**	Enable/Disable I2C1 clock
18	**I2C0**	Enable/Disable I2C0 clock
17	**SPI1**	Enable/Disable SPI1 clock
16	**SPI0**	Enable/Disable SPI0 clock
15	-	
14	**CRC16**	Enable/Disable CRC16 clock
13 9	-	
8	**GPIO**	Enable/Disable GPIO clock
7	**TIMER69**	Enable/Disable TIMER6, 7,8,9 clock
6	**TIMER25**	Enable/Disable TIMER2, 3,4,5 clock
5	**TIMER01**	Enable/Disable TIMER0, 1 clock
4	**FRT**	Enable/Disable Free-run Timer clock
3	**WDT**	Enable/Disable Watch-dog Timer clock
2 0	-	

3.5.11 PMU_CCR PMU Clock Control Register

The PMU_CCR register can control the clock oscillation of the MCU internal and external oscillators. The internal oscillator has a RINGOSC that outputs a 1 MHz waveform and an internal oscillator (IOSC16) that outputs 16 MHz. The external oscillator uses the output signals of the external main crystal (SXTAL) outputting a 32.768 kHz RTC (Real Time Clock) and the external main crystal (XTAL) outputting 4 MHz to 10 MHz frequency.

PMU_CCR=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
Reserved																								ROSCEN		IOSC16EN		SXOSCEN		MXOSCEN	
-																								10		00		00		00	
-																								RW		RW		RW		RW	

31 8	-		
7 6	ROSCEN	Enable internal ring oscillator (RINGOSC)	
		0x	Internal ring oscillator stop
		10	Internal ring oscillator oscillating – 1:1 output
		11	Internal ring oscillator oscillating – 1/2 output
5 4	IOSC16EN	Enable internal 16MHz oscillator (IOSC16)	
		0x	Internal 16MHz oscillator stop
		10	Internal 16MHz oscillator oscillation – 1:1 output
		11	Internal 16MHz oscillator oscillation – 1/2 output
3 2	SXOSCEN	Enable external sub-crystal oscillator (SXOSC)	
		0x	External sub-crystal oscillator stop
		10	External sub-crystal oscillator – 1:1 output
		11	External sub-crystal oscillator – 1/2 output
1 0	MXOSCEN	Enable external main crystal oscillator (MXOSC)	
		0x	External crystal oscillator stop
		10	External crystal oscillator – 1:1 output
		11	External crystal oscillator – 1/2 output

3.5.12 PMU_CMR PMU Clock Monitoring Register

The PMU_CMR register supports an oscillator activated in the PMU_CCR (PMU Clock Control Register) to monitor or verify the normal oscillation of the external main crystal (MXOSC) using an internal ring oscillator (RINGOSC). MXOSCFAIL (1) occurs when the oscillation of the external main source oscillates abnormally after activating the external main crystal oscillation interrupt.

PMU_CMR=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
Reserved																	MXOSCIE	Reserved					MXOSCMNT	Reserved	MXOSCIF	Reserved					MXOSCSTS
																	0						0		0						0
																	RW						RW		RC						RC

31 15	-		
14	MXOSCIE		Enable/Disable the interrupt event of monitoring external main crystal oscillation (AND with MXOSCSTS)
		0	Disble
		1	Enable
13 19	-		
8	MXOSCMNT		Enable/Disable the monitoring function of external main crystal oscillator
		0	Disable
		1	Enable
7	-		
6	MXOSCIF		Failure dectection flag of external main crystal oscillation (To clear this flag, write ‘1’ to the bit)
		0	No failure detected
		1	Failure detected
5 1	-		
0	MXOSCSTS		The oscillation status of external main crystal oscillation (To clear this flag, write ‘1’ to the bit)
		0	No oscillation
		1	Oscillation detected

3.5.13 PMU_MCMR PMU Main Clock Monitoring Register

This register can set the monitoring function and interrupt function to check the oscillation status of the system main clock (HCLK). When using this register, be sure to set the RECOVER bit to 1 to automatically switch the main clock to the internal ring oscillator when the main clock oscillation error occurs. MCKFAIL (15) interrupt occurs when main clock oscillation becomes abnormal opereation after enabling main clock oscillation monitoring interrupt.

PMU_MCMR=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																RECOVER	MCKIE	Reserved					MCKMNT	Reserved	MCKIF	Reserved					MCKSTS
-																0	0	-					0	-	0	-					0
-																RW	RW	-					RW	-	RC	-					RO

31 **16**			
15	**RECOVER**	Automatically switches the internal ring oscillator when the main clock oscillator error occurred.	
		0	Automatic switching function is disable
		1	Automatic switching function is enable
14	**MCKIE**	Main oscillator monitoring interrupt enable/disable selection (MCKSTS and use)	
		0	MMCKFAIL interrupt disable
		1	MMCKFAIL interrupt enable
13 **9**	**-**		
8	**MCKMNT**	Main oscillator monitoring enable	
		0	Disable
		1	Enable
7	**-**		
6	**MCKIF**	Main oscillator state interrupt (main oscillator defect detection. MMCKIE&AND is the previous value) (cleared by written with ‘1’)	
		0	No failure
		1	Failure occurred
5 **1**	**-**		
0	**MCKSTS**	Main oscillator operating state If MMCKMNT is activated, it will be displayed in real time. (cleared by written with ‘1’).	
		0	Main oscillator operation error
		1	Main oscillator normal operating

3.5.14 PMU_BCCR PMU Bus Clock Control Register

This register is used to set the main system clock (HCLK). By setting the PLLCLKSEL bit value, IOSC16 or external XTAL can be selected as the PLL input clock source. At this time, IOSC16 must be turned on even if external XTAL is set as PLL input.

PMU_BCCR=0x4000_003C

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">Reserved</td><td>PCLKDIV</td><td colspan="2">HCLKDIV</td><td colspan="2">Reserved</td><td>PLLCLKDIV</td><td>PLLCLKSEL</td><td colspan="2">Reserved</td><td colspan="2">MCLKSEL</td></tr>
<tr><td colspan="21">-</td><td>0</td><td colspan="2">00</td><td colspan="2">-</td><td>0</td><td>0</td><td colspan="2">-</td><td colspan="2">00</td></tr>
<tr><td colspan="21">-</td><td>RW</td><td colspan="2">RW</td><td colspan="2">-</td><td>RW</td><td>RW</td><td colspan="2">-</td><td colspan="2">RW</td></tr>
</table>

31 **11**	**-**		
10	**PCLKDIV**	PCLK clock divider setting	
		0	PCLK = HCLK
		1	PCLK = HCLK/2
9 **8**	**HCLKDIV**	HCLK clock dvider setting	
		0x	HCLK is System Clock (1:1)
		10	HCLK is System Clock/2
		11	HCLK is System Clock/4
7 **6**	**-**		
5	**PLLCLKDIV**	PLL input clock divider	
		0	1:1
		1	PLL input/2
4	**PLLCLKSEL**	PLL input clock selection (IOSC16 must be turned on even if external XTAL is set as PLL input)	
		0	IOSC16 clock
		1	XTAL clock
3 **2**	**-**		
1 **0**	**MCLKSEL**	Main clock source selection	
		00	The ring oscillator
		01	Sub Oscillator
		10	PLL
		11	PLL bypass (PLL input clock)

3.5.15 PMU_PCSR PMU Peripheral Clock Source Register

This register selects the source clock to be supplied to each peripheral device.

PMU_PCSR=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
Reserved																						T69CS		T25CS		T0CS		FRTCS		WDTCS	
-																						00		00		00		00		00	
-																						RW		RW		RW		RW		RW	

31 **10**	**-**		
9 **8**	**T69CS**	Select the timer 6,7,8,9 input clock	
		00	Main XTAL
		01	IOSC16 (slower than PCLK/2)
		10	Sub XTAL (slower than PCLK/2)
		11	RingOSC (slower than PCLK/2)
7 **6**	**T25CS**	Select timer 2,3,4,5 input clock	
		00	Main XTAL
		01	IOSC16 (slower than PCLK/2)
		10	Sub XTAL (slower than PCLK/2)
		11	RingOSC (slower than PCLK/2)
5 **4**	**T01CS**	Select timer 0,1 input clock	
		00	Main XTAL
		01	IOSC16 (slower than PCLK/2)
		10	Sub XTAL (slower than PCLK/2)
		11	RingOSC (slower than PCLK/2)
3 **2**	**FRTCS**	Select FRT input clock	
		00	Main XTAL
		01	IOSC16
		10	Sub XTAL
		11	RingOSC
1 **0**	**WDTCS**	Select WDT input clock	
		00	Main XTAL
		01	IOSC16
		10	Sub XTAL
		11	RingOSC

3.5.16 PMU_COR PMU Clock Output Register

The PMU_COR register is used to set the CLKO and TRACE functions.
The CLKO function is used to monitor the internal main system clock through the CLKO port.
The TRACE function represents the TRACE function of the debugger.
When changing the clock or delay of TRACE, you must write the TRACEDIVKEY key value together. TRACECLK INV and TRACEDATA DELAY are used to fine-tune the TRACE signal.

PMU_COR=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
Reserved																TRACEDIVKEY				Reserved				TRACECLKINV	TRACEDATADELAY	CLKOSEL	CLKOEN	CLKODIV			
-																0000				-				0	0	0	0	0000			
-																WO				-				RW	RW	RW	RW	RW			

31 **16**	-		
15 **12**	**TRACEDIVKEY**	To change the value of TRACECLKINV, you must use 4'hA at the same time.	
		0xA	TRACE key value when changing the trace clock (TRACECLKINV) or data delay(TRACEDATADELAY)
11 **8**	-		
7	**TRACECLK INV**	TRACECLK output Inversion.	
		0	Original TRACECLK output
		1	TRACECLK output Inversion
6	**TRACEDATA DELAY**	Internal TRACE data delay selection	
		0	TRACE delay is 1/1 output (No delay)
		1	TRACE delay is 1/4 output
5	**CLKOSEL**	Select the CLKO output source	
		0	CLKO output is PLL Clock
		1	CLKO output is MCLK
4	**CLKOEN**	Enable CLKO divider output	
		0	Stop CLKO devider output (output is always 'L')
		1	Enable CLKO divider output
3 **0**	**CLKODIV**	CLKOEN is activated, the external output clock, set the value of the division of CLKO. Clock divider will be calculated by the following formula. If CLKODIV = 0, CLKO = PLLOUT (or MCLK) If CLKODIV > 0, $CLKO(Hz) = \frac{PLLOUT\ or\ MCLK}{2 \cdot CLKODIV}$	

3.5.17 PMU_PLLCON PLL Control Register

PMU_PLLCON is a register for PLL frequency output. It can output precise frequency in 1MHz unit by setting the value of PLL synthesizer of A33G52x. To control the PLL, first write 0x80750000 in the PMU_PLLCON register and calculate the parameter value for the PLL output frequency according to the following formula and write to each corresponding bit. Refer to 3.7.4 for setting the PLL frequency.

PMU_PLLCON=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	VCOMODE	MULT	DIV	nPLLRST	PLLEN	nBYPASS	LOCKSTS	Reserved	PREDIV	Reserved	POSTDIV
-	0	00000000	0000	0	0	0	0	-	000	-	0000
-	RW	RW	RW	RW	RW	RW	RO	-	RW	-	RW

31 **29**	**-**		
28	**VCOMODE**		To set the operation mode of the VCO Determine the D value of the frequency calculation
		0	Determine the D value of the frequency calculation
		1	Normal VCO mode
27 **20**	**MULT**		Set the output multiplication of VCO Determine the N1 value of the frequency calculation
19 **16**	**DIV**		Set the output multiplication of VCO Determine the N2 value of the frequency calculation
15	**nPLLRST**		PLL reset
		0	PLL reset signal
		1	Release PLL reset signal PLL (PLL On)
14	**PLLEN**		Enable/Disable PLL Operation
		0	disable
		1	enable
13	**nBYPASS**		Select PLL output clock
		0	Outputs by bypassing the PLL input clock (FIN). Refer to PMU_MCCR[5:4] in 3.5.14
		1	Outputs PLL frequency (FOUT) calculated by the PLL formula. Refer to 3.6.2
12	**LOCKSTS**		PLL lock status
		0	PLL is unlock (Unstable output of PLL clock)
		1	PLL is locked (Stable output of PLL clock)
11	**-**		
10 **8**	**PREDIV**		Pre-Divider division control Determine the R value of the frequency calculation
7 **4**	**-**		
3 **0**	**POSTDIV**		Post divider control. Determine the P value of the frequency calculation
			The final output PLL frequency can be calculated by the following formula. $\mathrm{FOUT} = \frac{\mathrm{PLLINCLK} \times (\mathrm{N_1} + 1)}{(\mathrm{R} + 1) \times (\mathrm{N_2} + 1) \times (\mathrm{P} + 1)} \times (\mathrm{D} + 1)$

3.5.18 PMU_LVDCON LVD Control Register

This register controls the LVD (Low Voltage Detector) operation of the MCU. After enabling the LVD reset (or LVD interrupt) function, an LVD reset (or LVD interrupt) will occur if the system voltage detects the level set at LVDIL (or LVDRL).

PMU_LVDCON=0x4000_0054

31 30 29 28 27 26 25 24 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	LVDEN	LVDRF	Reserved	LVDREN	LVDRL	Reserved	LVDIF	LVDICS	Reserved	LVDIEN	LVDIL
-	1	0	-	1	000	-	0	0	-	0	000
-	RW	RC	-	RW	RW	-	RC	RO	-	RW	RW

31 16	-		
15	LVDEN	Enable/Disable LVD (Low Voltage Detection)	
		0	Disable
		1	Enable
13 12	-		
14	LVDRF	LVD reset flag (Cleared when written ‘1’)	
		0	LVD reset voltage has NOT been detected
		1	LVD reset voltage has been detected
11	LVDREN	Enable/Disable LVD Reset	
		0	Disable LVR Reset
		1	Enable LVR Reset
10 8	LVDRL	Select LVD Reset voltage level	
		000	2.60V (Default)
		001	2.80V
		010	3.00V
		011	3.30V
		100	3.75V
		101	4.00V
		110	4.25V
		111	4.50V
7	-		
6	LVDIF	LVD interrupt flag (cleared when writing ‘1’)	
		0	LVD interrupt has NOT been detected
		1	LVD interrupt has been detected
5	LVDICS	LVD interrupt current status	
		0	Below LVD interrupt voltage
		1	Above LVD interrupt voltage
4	-		
3	LVDIEN	Enable LVD Interrupt	
		0	LVD interrupt disable
		1	LVD interrupt enable
2 0	LVDIL	Select LVD Interrupt voltage level	
		000	2.6V
		001	2.8V
		010	3.0V
		011	3.3V
		100	3.75V
		101	4.0V
		110	4.25V
		111	4.5V

3.5.19 PMU_VDCCON VDC/LVD Trimming Register

PMU_VDCCON is a register that sets the VDC built in the MCU. This register has the internal configuration information of the chip.
This mode is NOT available to general users. (Write prohibited)

PMU_VDCCON=0x4000_0058

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFLVL	Reserved	BMR_TRIM		Reserved	BOPENB	Reserved	VDROP	DFLVL_EN		VDROP_EN		BMRT_EN	LVDT_EN	VDCT_EN	VDCD_EN	LVDTRIM				VDCTRIM				VDCDELAY							
0	-	00		-	0	-	0	00		00		0	0	0	0	0000				0000				FF							
RW	-	RW		-	RW	-	RW	WO		WO		WO	WO	WO	WO	RW				RW				RW							

31	DFLVL	BOD default level(Hidden) signal Setting	
		0	BODRSEL_I/BODISEL_I Enable (BOD Level = 2.6~4.5V)
		1	BODRSEL_I/BODISEL_I Disable (BOD Level = 2.2V)
30	-		
29 28	BMRTRIM	BMR voltage trimming value Setting	
		00	0.9860V
		01	1.0736V
		10	0.9173V
		11	0.9394V
27	-		
26	BOPENB	BURST OFF / PULLUP enable / Circuit OFF (Changable by writing ‘10 ’ to PFLVL_EN field)	
		0	Pull-up Circuit Enable
		1	Pull-up Circuit Disable
25	-		
24	VDROP	VDC Level Down Mode	
		0	VDD = 1.5V
		1	VDD = 1.39V (8% down)
23 22	DFLVL_EN	Maintain/Change DFLVL value (Write Only, The value currently being written is applied)	
		00 11	Maintain the value of DFLVL, BOPENB
		10	Changeable BOPENB value
		01	Changeable DFLVL value
21 20	VDROP_EN	Maintain/Change VDROP value	
		00 x1	Maintain VDROP value
		10	VDROP value can be changed
19	BMRT_EN	Maintain/Change BMRTRIM value	
		0	Maintain BMRTRIM value
		1	Changeable BMRTRIM value
18	LVDT_EN	Maintain/Change LVDTRIM (Write Only, The value currently being written is applied)	
		0	Maintain LVDTRIM value
		1	Changeable LVDTRIM value
17	VDCT_EN	Maintain/Change VDCTRIM value (Write Only, The value currently being written is applied)	
		0	Maintain VDCTRIM value
		1	Changeable VDCTRIM value
16	VDCD_EN	Maintain/Change VDCDELAY value (Write Only, The value currently being written is applied)	
		0	Maintain VDCDELAY value

		1 Changeable VDCDELAY value
15 **12**	**LVDTRIM**	Setting LVD (Low Voltage Detector) voltage trimming value
11 **8**	**VDCTRIM**	Setting VDC voltage trimming value
7 **0**	**VDCDELAY**	Setting VDC Output delay time during wake-up from power-down mode

3.5.20 PMU_IOSC16TRIM IOSC16 Trimming Register

PMU_IOSC16TRIM controls or controls the operation of the 16MHz internal oscillator on the Chip.
This mode is NOT available to general users. (Write prohibited)

PMU_IOSC16TRIM=0x4000_005C

31	30	29	28	27	26	25	24	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											LT_EN	LTM_EN	TSL_EN	UDCH_EN	TCAL_EN	Reserved		LT				LTM		TSL			UDCH		TCAL1		
-											0	0	0	0	0	-		1000				10		100			01		011		
-											WO	WO	WO	WO	WO	-		RW				RW		RW			RW		RW		

31 **21**	-	
20	**LT_EN**	Maintain/Change LT value 0 Maintain LT value 1 Changeable LT value
19	**LTM_EN**	Maintain/Change LTM value 0 Maintain LTM value 1 Changeable LTM value
18	**TSL_EN**	Maintain/Change TSL value 0 Maintain TSL value 1 Changeable TSL value
17	**UDCH_EN**	Maintain/Change UDCH value 0 Maintain UDCH value 1 Changeable UDCH value
16	**TCAL_EN**	Maintain/Change TCAL value 0 Maintain TCAL value 1 Changeable TCAL value
15 **14**	-	
13 **10**	**LT**	Period tuning: +/-2.8% (default: 1000)
9 **8**	**LTM**	Period tuning: +/-0.7% (default: 1000)
7 **5**	**TSL**	Period & Temp tuning: +/-1.8% (default: 100)
4 **3**	**UDCH**	Period & Temp tuning (default: 01)
2 **0**	**TACL1**	Temp tuning (default: 011)

3.5.21 PMU_EOSCCON External OSC Control Register

This register controls the external oscillator operation.
When changing the field value, PMU_EOSCCON [31:16] must be written with '0x18A2'.

PMU_EOSCCON=0x4000_0060

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td>SOSCNFOFF</td><td>Reserved</td><td colspan="2">SOSCISEL</td><td colspan="4">Reserved</td><td>MOSCNFOFF</td><td>Reserved</td><td colspan="2">MOSCNFSEL</td><td colspan="2">Reserved</td><td colspan="2">MOSCISEL</td></tr>
<tr><td colspan="16">-</td><td>0</td><td>-</td><td colspan="2">00</td><td colspan="4">-</td><td>0</td><td>-</td><td colspan="2">00</td><td colspan="2">-</td><td colspan="2">01</td></tr>
<tr><td colspan="16">-</td><td>RW</td><td>-</td><td colspan="2">RW</td><td colspan="4">-</td><td>RW</td><td>-</td><td colspan="2">RW</td><td colspan="2">-</td><td colspan="2">RW</td></tr>
</table>

31 **16**	-		
15	**SOSCNFOFF**	Enable/Disable Noise Filter of SUB OSC	
		0	Enable Nosie Filter
		1	Disable Nosie Filter
14	-		
13 **12**	**SOSCISEL**	Current strength Control of SUB OSC	
		00	Strong
		≀	
		11	Weak
11 **8**	-		
7	**MOSCNFOFF**	Enable/Disable Noise Filter of MAIN OSC	
		0	Enable Nosie Filter
		1	Disable Nosie Filter
6	-		
5 **4**	**MOSCNFSEL**	Noise Filter Control of MAIN OSC	
		00	Slow
		≀	
		11	Fast
3 **2**	-		
1 **0**	**MOSCISEL**	Current strength Control of MAIN OSC	
		00	Strong
		≀	
		11	Weak

3.5.22 PMU_EXTMODERExternal Mode Read Register

The PMU_EXTMODER register reads the external pin value in test mode to determine which mode is set. This mode is NOT available to general users. (Write prohibited)

PMU_EXTMODER=0x4000_0070

31	30	29	28	27	26	25	24	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																							FMTEST	Reserved		PDON	XRSTON	ROSCAON	SCANMD	TEST	BOOT
-																							0	-		0	0	0	0	0	0
-																							WO	-		RO	RO	RO	RO	RO	RO

31 9	-		
8	FMTEST	Configure all pins to test mode. (Prohibited)	
		0	Set all pins to general mode
		1	Set all pins to test mode
7 6	-		
5	PDON	Read the value of external PDON pin.	
		0	PDON pin = ‘L’
		1	PDON pin = ‘H’
4	XRSTON	Read the value of external XRSTON pin.	
		0	XRSTON pin = ‘L’
		1	XRSTON pin = ‘H’
3	ROSCAON	Read the value of external ROSCAON pin.	
		0	ROSCAON pin = ‘L’
		1	ROSCAON pin = ‘H’
2	SCANMD	Read the value of external SCANMD pin.	
		0	SCANMD pin = ‘L’
		1	SCANMD pin = ‘H’
1	TEST	Read the value of external TEST pin.	
		0	TEST pin = ‘L’
		1	TEST pin = ‘H’
0	BOOT	Read the value of external BOOT pin.	
		0	BOOT pin = ‘L’
		1	BOOT pin = ‘H’

3.6 Functional Description

3.6.1 Operating Modes for Low-Power Consumption

The A33G52x series has various power-saving options and operating modes that help its applications to reduce power consumption. When designing an embedded system with an A33G52x microcontroller unit (MCU), it is important to fully understand the device details of the intended system to optimize the use of the MCU's peripherals and operating modes for minimal power consumption.

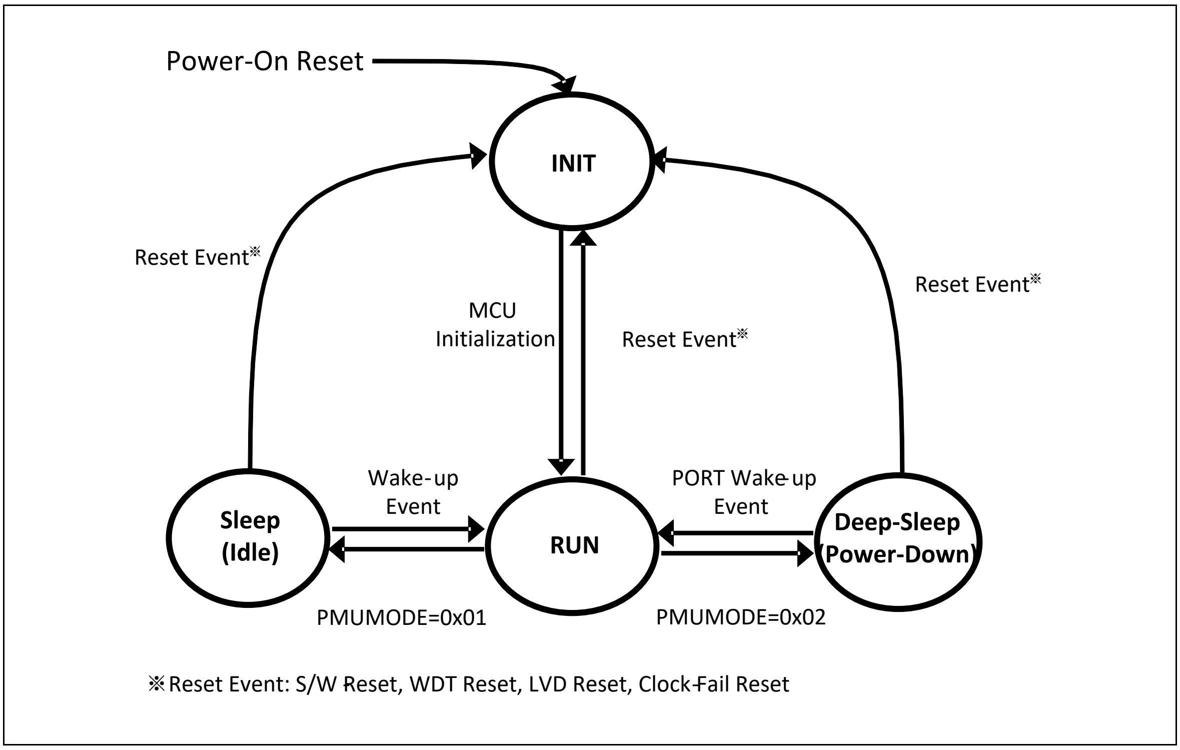


Figure 3.4. Operating modes of A33G52x

Run Mode
This mode is the normal operating mode of A33G52x MCUs. Once activated by an external power supply, the power-on reset (POR) holds the MCU chip in a reset state until it stabilizes. Upon its stabilization, the MCU initializes all clock sources and registers in initialization mode and then enters run mode. In run mode, all clock sources, buses, and peripherals can be enabled, and the current consumption will vary with their operating frequencies.

Sleep (Idle) Mode
When in run mode, writing 0x01 to the PMUMODE of the PMU_MR register triggers the MCU to enter sleep mode, in which the clock input to the CPU becomes disabled.
By using the wake-up source enable register (PMU_WSER), designers can set the wake-up sources of various peripherals, including GPIOA through GPIOF (5–10), FRT (4), WDT (3), main oscillator error (2), and LVD (0). When a wake-up event occurs based on the settings, the MCU will return to run mode.
In addition, the clock for each peripheral device can be individually enabled or disabled to supply power to only the minimum number of peripherals, thereby reducing power consumption.

Deep-Sleep (Power-Down) Mode
An A33G52x MCU operating in run mode can be switched to deep-sleep (power-down) mode by writing 0x00 to the main clock select (MCLKSEL) of the power management unit's (PMU's) bus clock control register (PMU_BCCR), which will set the ring oscillator as the main clock source, and then writing 0x02 to the PMUMODE of the PMU mode register (PMU_MR). In deep-sleep mode, the PLL and its clock source, the internal oscillator, are disabled, along with the system clock (HCLK) and peripheral clock (PCLK). This system stop allows for operation with minimal power consumption.
From among the various wake-up sources for peripherals provided by the wake-up source enable register (PMU_WSER), only those for the external GPIOA through GPIOF (5–10) can render wake-up events (levels) in deep-sleep mode.

By default, this mode automatically powers off the main oscillator (MXOSC) and sub-oscillator (SXOSC). As such, the peripherals fed with a clock signal from the main or sub-oscillator will stop operating in deep-sleep mode. To keep those peripherals running in this mode, the MCU can be configured in either of the following ways:
1) Set the ring oscillator as the clock source to feed the peripherals that need to continue running in deep-sleep mode.
2) To continue using the main or sub-oscillator in deep-sleep mode, set the ECLKMD bit of the PMU mode register (PMU_MR) to "1." This will prevent the main and sub-oscillators from automatically being powered off. Next, configure the PMU clock control register (PMU_CCR) manually to disable the clock sources that are NOT in use in deep-sleep mode. However, this method requires special care to ensure effective power saving in deep-sleep mode.

Table 3.6. Operability of the clock sources, buses, and modules in each operating mode

Functions \ MCU modes		Run mode	Sleep mode	Deep sleep mode
Clock sources	IOSC16	O	O	X
	RING	O	O	X
	MXOSC	O	O	△
	SXOSC	O	O	△
	PLL	O	O	X
Buses	AHB	O	O	X
	APB	O	O	X
Modules	CORE	O	X	X
	PMU	O	O	X
	FLASH	O	O	X
	WDT	O	O	X
	FRT	O	O	X
	TIMER	O	O	X
	PWM	O	O	X
	GPIO (PCU)	O	O	X*
	ADC	O	O	X
	UART	O	O	X
	I2C	O	O	X
	SPI	O	O	X

O: Operable
X: Not operable
△: Operability depends on the settings of PMU_MR<ECLKMD>.
*: If the GPIO's wake-up source is set in the PMU_WSER resister, its level event is available.

3.6.2 PLL Output Frequency Setting

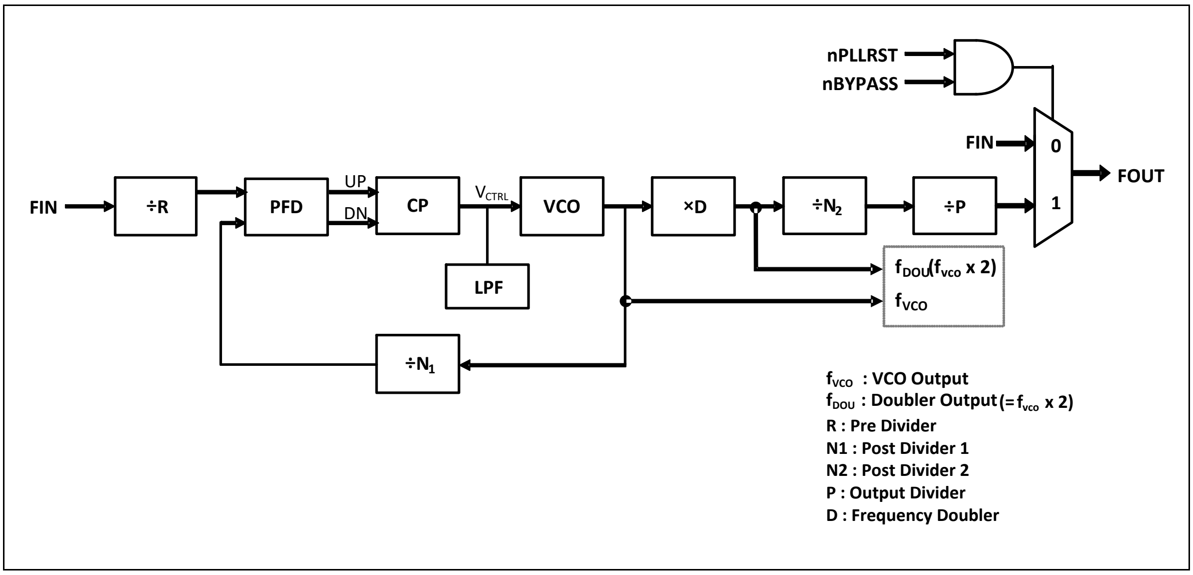


Figure 3.5. Block diagram of A33G52x PLL

[Calculating the PLL output frequency value]

The PLL of the A33G52x can accurately set the output frequency, F_{OUT}, in 1MHz increments. To control the PLL, first write 0x80750000 in the PMU_PLLCON register and calculate the parameter value for the PLL output frequency according to the following formula and write to each corresponding bit. 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.

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

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

$$\mathrm{VCO} = \mathrm{FIN} \times (\mathrm{N}_1 + 1), \quad 50\mathrm{MHz} \le \mathrm{VCO} \le 200\mathrm{MHz}\ if\ \mathrm{D} = 0$$

PMU_PLLCON also supports a doubler function that can double the VCO output through the bit setting of VCOMODE. When using this doubler, the output of the VCOx2 must be set between minimum 100 MHz and maximum 250 MHz.

$$\mathrm{VCOx2} = \mathrm{VCO} \times (\mathrm{D}+1), \qquad 100\mathrm{MHz} \le \mathrm{VCOx2} \le 250\mathrm{MHz}\ if\ \mathrm{D} = 1$$

As a result, the final frequency of PLL, FOUT, can be obtained from the formula below using the formula above.

$$\mathrm{FOUT}\ = \frac{\mathrm{PLLINCLK} \times (\mathrm{N}_1+1)}{(\mathrm{R}+1) \times (\mathrm{N}_2+1) \times (\mathrm{P}+1)} \times (\mathrm{D}+1)\ = \frac{\mathrm{FIN} \times (\mathrm{N}_1+1)}{(\mathrm{N}_2+1) \times (\mathrm{P}+1)} \times (\mathrm{D}+1)$$

3.6.3 Cold-Reset and Wram-Reset

The A33G52x offers a variety of reset functions to protect the entire system from malfunctions in a service routine or core and to ensure the operation of the MCUs that the user can trust. Since the reset timing of the A33G52x is based on the ring oscillator (RINGOSC), it is necessary to consider the error (± 50%) of RINGOSC when applying the reset function to the user application.
The A33G52x supports two types of reset: Cold-Reset, which causes the system to be powered off and then on to initialize the system, and Warm-Reset, which initializes the system without any system power fluctuations. After the reset occurs, the user code is executed when the system boots normally.

In the case of the former, when the system power is on, a power on reset (POR) reset occurs when the power reaches 1.2V, and a LVD reset occurs when the voltage value set by LVD reset is detected. After performing a cold-reset, a warm-reset occurs and the system is initialized and the boot ROM and main code are executed.
The figure below shows the A33G52x cold reset timing diagram.

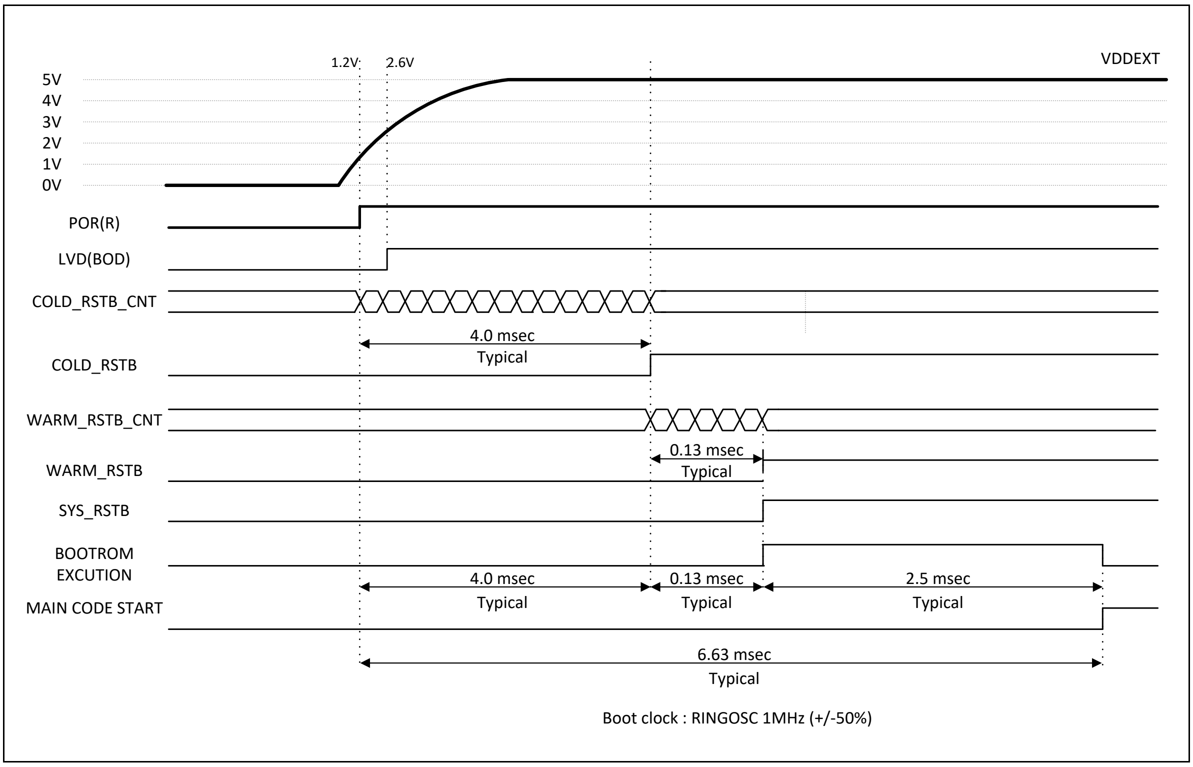


Figure 3.6. Cold Reset timing of A33G52x

In the latter case, it is also called a soft reset. While the system is operating at VVDD voltage, the mcu can occur reset itself by running a specific routine or by configuring a reset source. When this warm reset is happend, the system executes the boot ROM and main code.

A reset event can be performed for each module by setting the reset source provided by PMU_RSER (PMU reset source enable register) of A33G52x. The figure below shows the timing of the warm-reset.

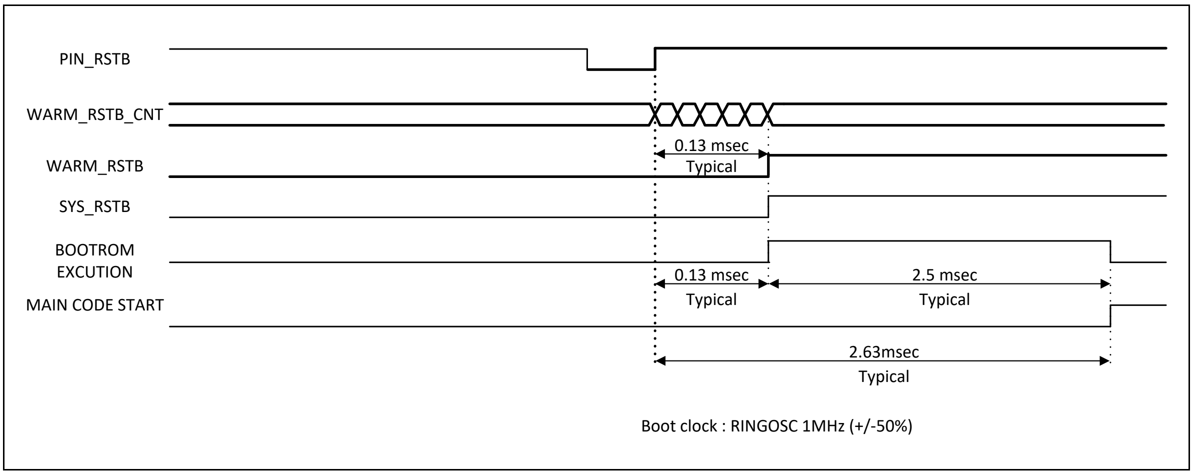


Figure 3.7. Warm Reset timing of A33G52x

3.6.4 Clock Monitoring

Using the clock monitoring function of A33G52x causes a reset or an interrupt when the oscillator stops oscillation or occurs erroneously oscillation due to external noise or the like.
However, when this interrupt occurs, the core clock must NOT be the main oscillator. This function uses the internal ring oscillator for monitoring and clock switching. To enable this function, it is necessary to set the internal ring oscillator to always operate.
Procedures for main oscillator oscillation and monitoring function are shown below.

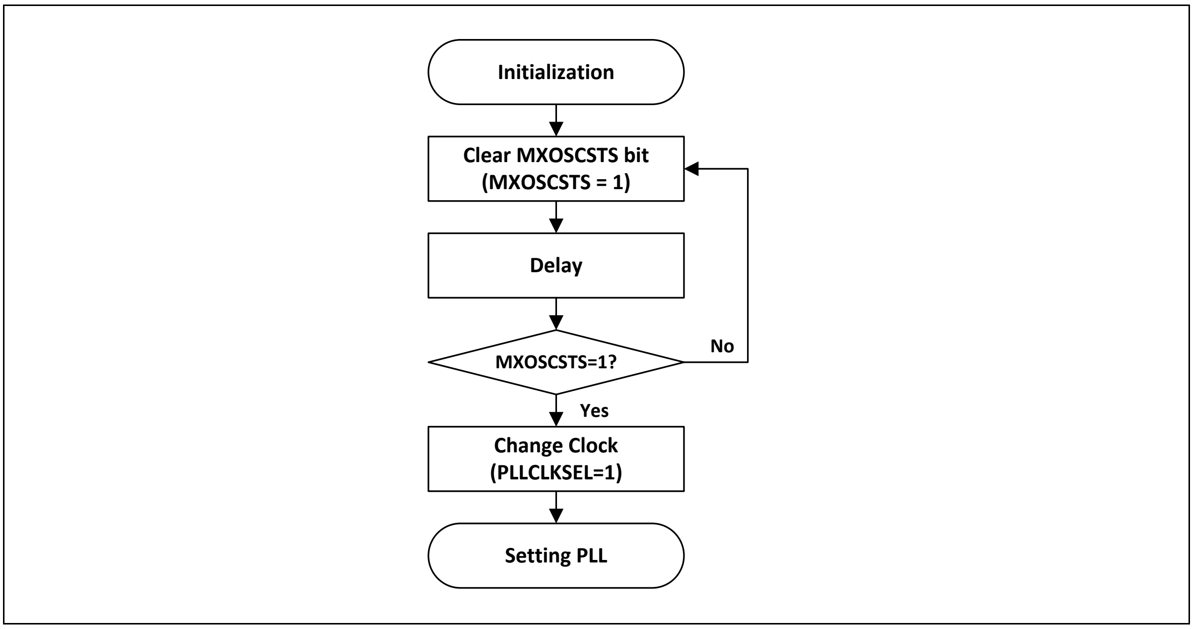


Figure 3.8. The main oscillator operating procedures

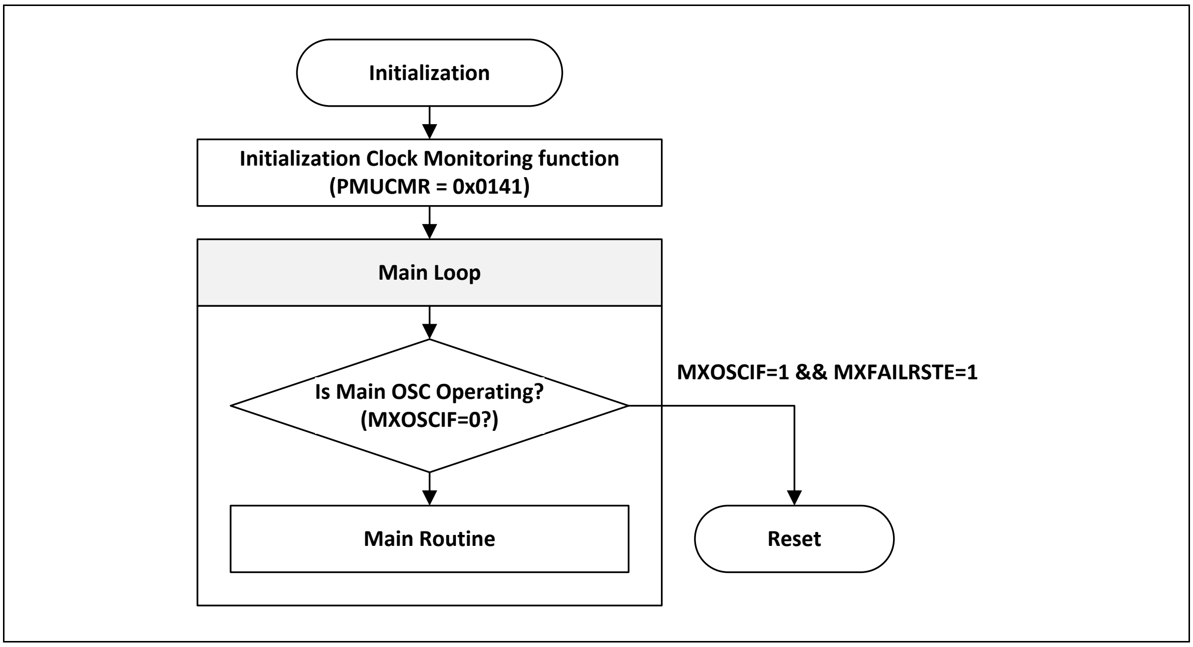

Initialization
Initialization Clock Monitoring function (PMUCMR = 0x0141)
Main Loop
Is Main OSC Operating? (MXOSCIF=0?)
MXOSCIF=1 && MXFAILRSTE=1
Main Routine
Reset

Figure 3.9. Functional flow chart of clock monitoring

3.6.5 Setting Example

<Example1> Set ring oscillator to system clock

WDT_CON<WDH> = ‘0’	: Disables WDT (Watchdog timer)
PMU_CCR<ROSCEN[7:6]> = ‘10’	: Oscillation 1:1 internal ringosc
PMU_BCCR<PCLKDIV> = ‘0’	: PCLK = HCLK
PMU_BCCR<HCLKDIV[9:8]> = ‘00’	: HCLK = Main Clock
PMU_BCCR<MCLKSEL[1:0]> = ‘00’	: Sets ring oscillator to system clock

<Example2> 8MHz external main crystal oscillation and monitoring setup.

PC_MR<P15[31:30]> = ‘01’	: Sets the function of PC15 to XTALI
PC_MR<P14[29:28]> = ‘01’	: Sets the function of PC14 XTALO
PC_CR<P15[31:30]> = ‘10’	: Sets the direction of XTALI port to analog input
PC_CR<P14[29:28]> = ‘10’	: Set the direction of XTALO to analog input
PC_PCR<D15> = ‘0’	: Select pull-up resistor of XTALI
PC_PCR<P15> = ‘0’	: Disable pull-up/pull-down register of XTALI
PC_PCR<D14> = ‘0’	: Select pull-up resistor of XTALO
PC_PCR<P14> = ‘0’	: Disable pull-up/pull-down register of XTALO
PMU_CCR<MXOSCEN[1:0]> = ‘10’	: Enable oscillation of external main crystal (1:1)
PMU_CMR<MXOSCMNT> = ‘1’	: Enable monitoring of external main oscillatior
[READ] PMU_CMR<MXOSCSTS>	: Initialize status of external clock oscillation (Read Clear)

<Example3> PLL output frequency setting (74MHz)

FM_CFG<CWAIT[4:0]> = '00011’	: Set wait-time of code flash.
FM_CFG<DWAIT[12:8]> = ‘00011’	: Set wait-time of data flash.
PLL_CON = ‘0x80750000’	: Set key value 0x80750000 for using A33G52x PLL.
PLL_CON<PLLRESB>= ‘1’	: Enable PLL reset
PLL_CON<PLLEN> = ‘1’	: Enable PLL operation
PLL_CON<PREDIV[10:8]> = ‘011’	: Set the value of pre-divider (R = 3)
PLL_CON<MULT[27:20]> = ‘00110110’	: Set the value of VCO (N1 = 36)
PLL_CON<DIV[19:16]> = ‘0000’	: Set the value of VCO divider (N2 = 0)
PLL_CON<POSTDIV[3:0]> = ‘0000’	: Set the value of Post-Divider (P = 0)
PLL_CON<VCOMODE> = ‘0’	: Set the value of VCO doubler (D = 0)
PLL_CON<BYPASS> = ‘1’	: Set PLL clock as multiplier output

CHAPTER 4. PMC (Port Map Controller)

Table 4.1. Operation Summary

Item	Pin Name	Remark
Clock usage	PCLK	Debounce settings
Reset Source	PMU_PER settings	
Reset Generation	None	
Interrupt Generation	Depend on the conditions of port interrupts (GPIOA(16)~GPIOF(21))	Pn_ICR
Interrupt Clear Method	Writing ‘1’ to the interrupt bit	Pn_ISR

4.1 Overview

The PMC (Port Map Controller) controls the pins of the A33G52x MCU. By setting the PMC block register according to the purpose of the user application, you can set functions such as pin function setting, use I / O function, pull-up / pull-down, and debounce.

Features of PMC (Port Map Controller)

- MUX registers for setting the purpose of each pin.
- Direction(Input/Output) setting of each pin
  - Push-pull output
  - Open-drain output
  - Logic input
  - Analog input
- Pull-up/Pull-down resistor setting of each pin
- Interrupt settings for each input pin
  - Level interrupt
  - Rising-edge interrupt
  - Falling-edge interrupt
  - Supports up to 6 GPIO interrupts : GPIOA(16)~GPIOF(21)
- Debounce setting of each pin

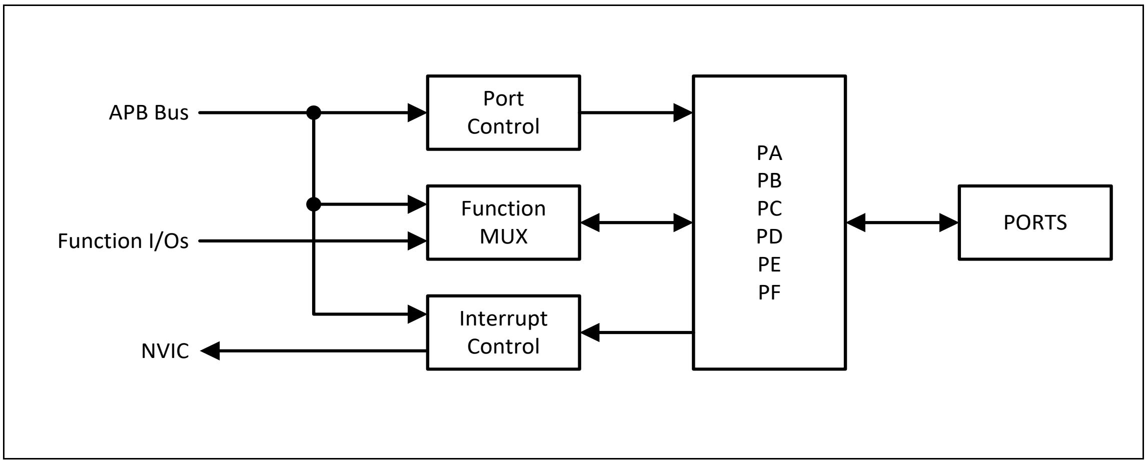


Figure 4.1. Block Diatram of PMC (Port Map Controller)

4.2 Block Diagram

4.2.1 PA Port

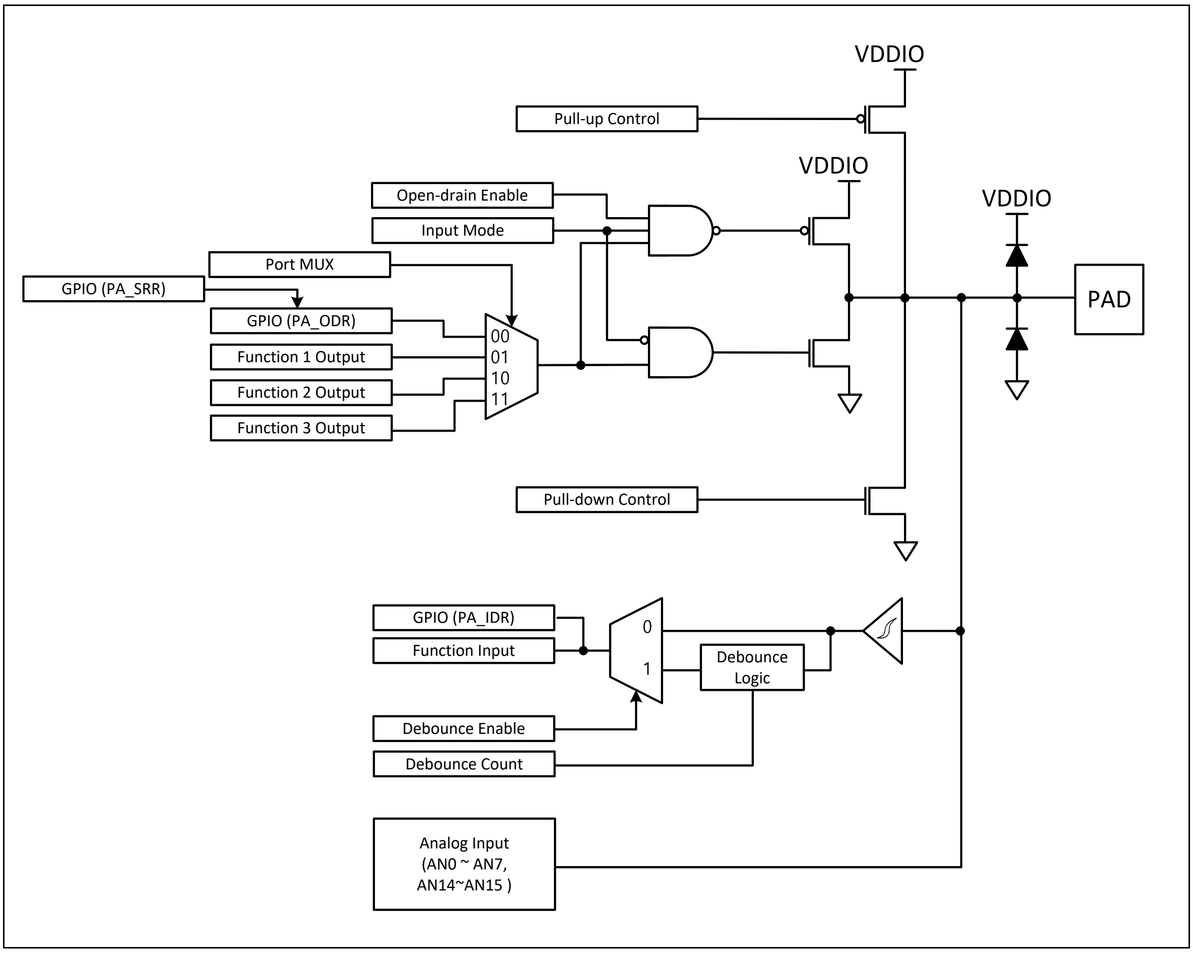


Figure 4.2. Block diagram of PA port

4.2.2 PB Port

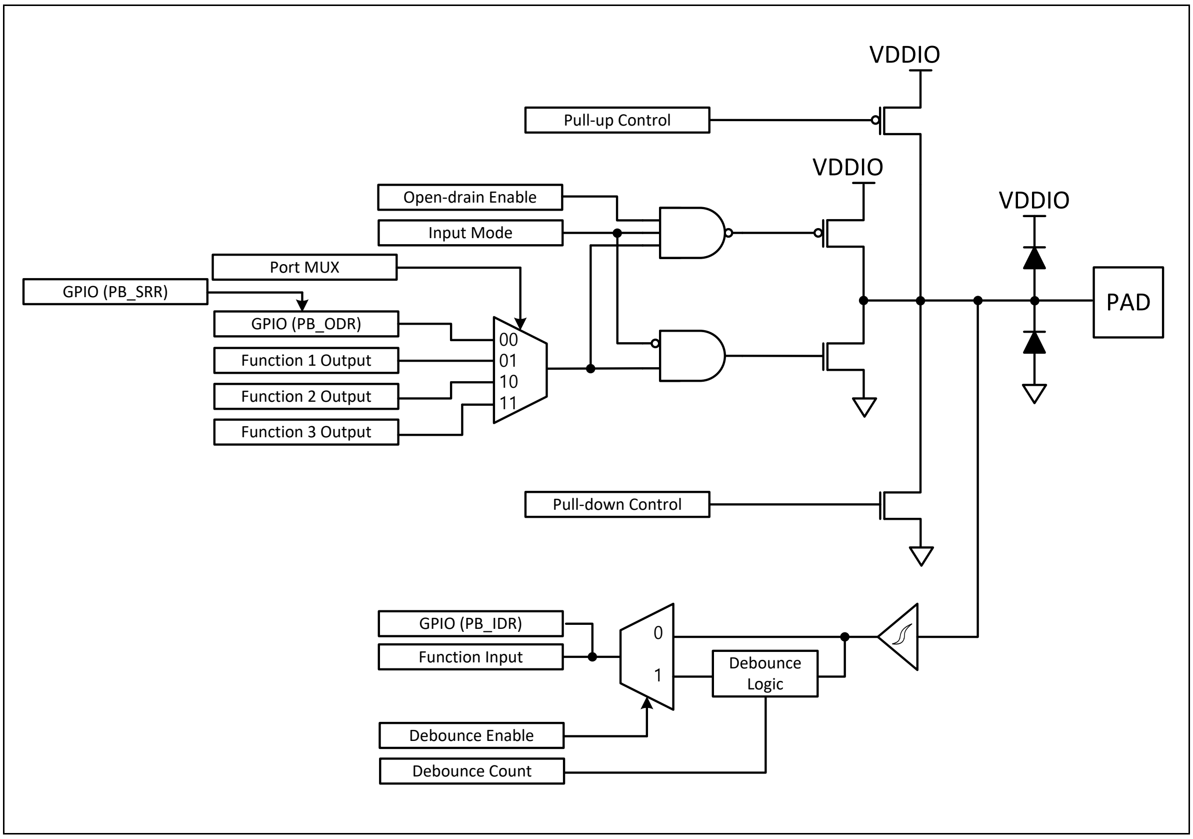


Figure 4.3. Block diagram of PB port

4.2.3 PC Port

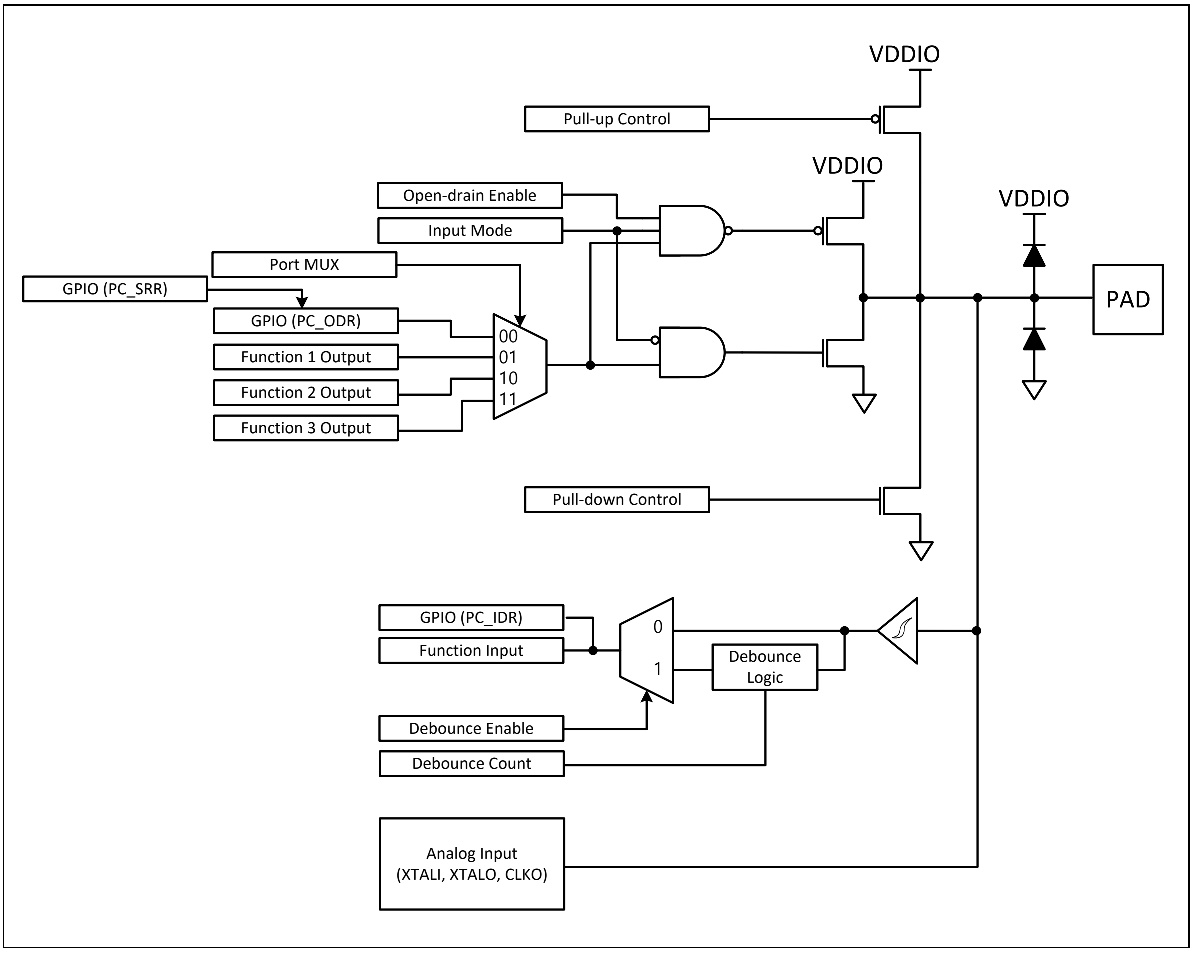


Figure 4.4. Block diagram of PC port

4.2.4 PD Port

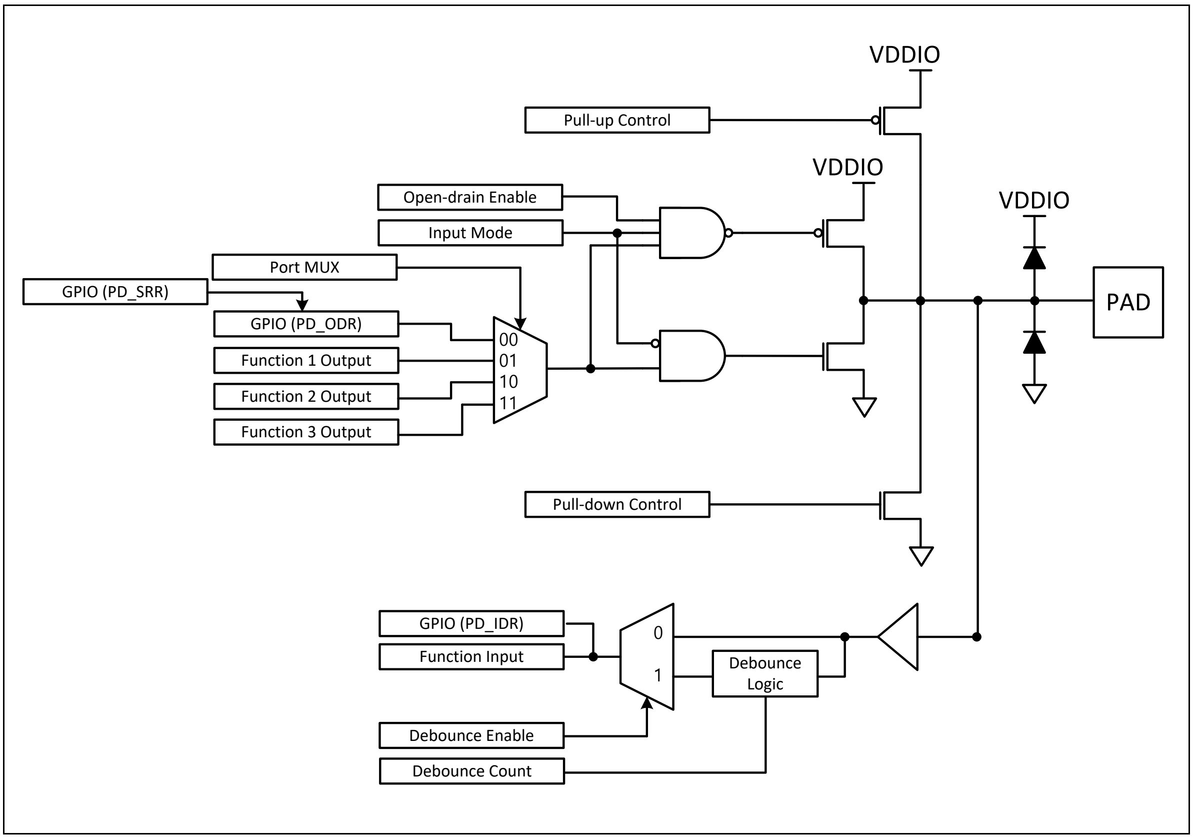


Figure 4.5. Block diagram of PD port

4.2.5 PE Port

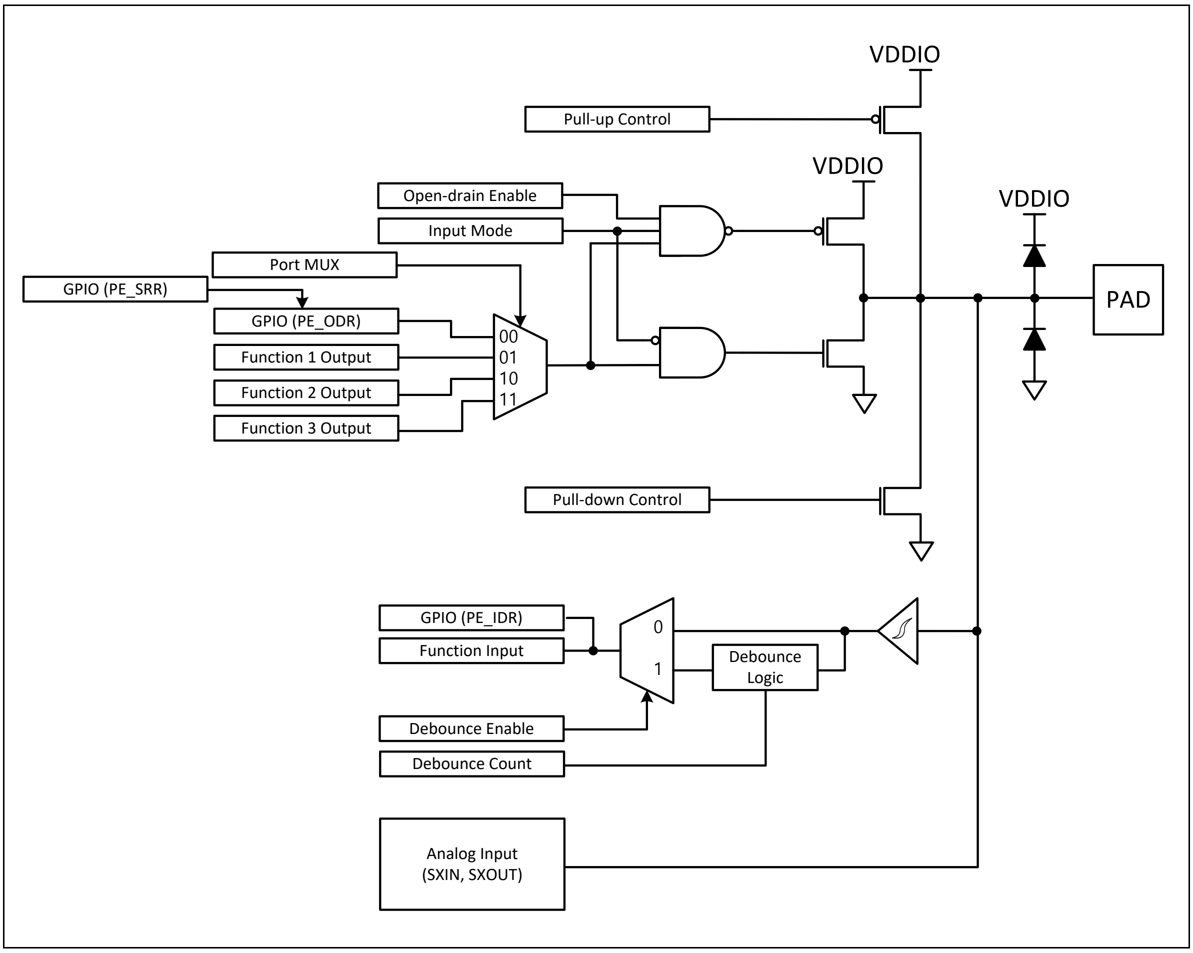


Figure 4.6. Block diagram of PE port

4.2.6 PF Port

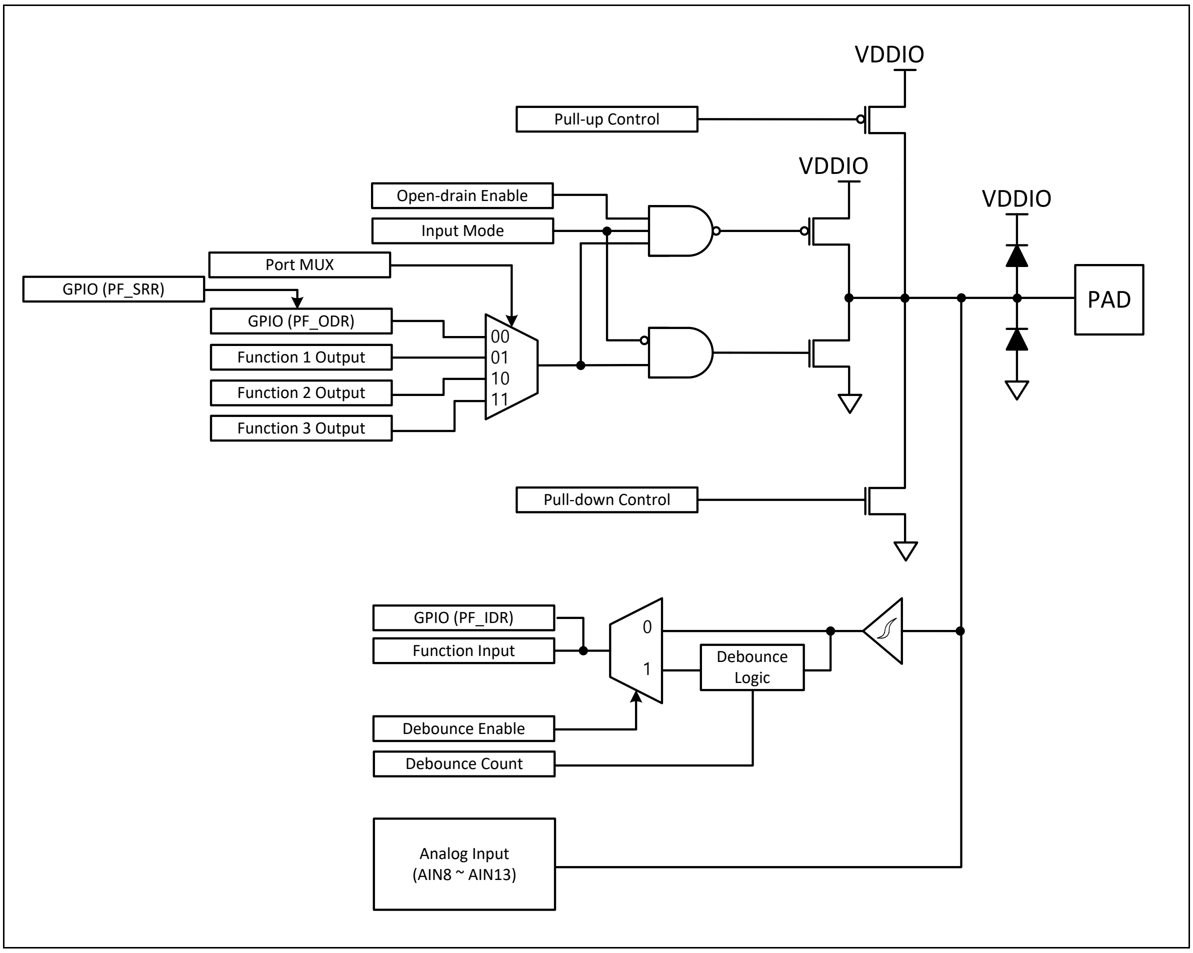


Figure 4.7. Block diagram of PF port

4.3 Pin Configuration

Every pin of A33G52x is a multi-function pin. The function selection for each port is shown as follows.

Table 4.2. Function selection of the external pins

Port	Pin	Function			
		00			**00(default)**
PA	0	**PA0**			AN0
	1	**PA1**			AN1
	2	**PA2**			AN2
	3	**PA3**			AN3
	4	**PA4**			AN4
	5	**PA5**			AN5
	6	**PA6**	T0C		AN6
	7	**PA7**	T1C		AN7
	8	**PA8**	T2C		
	9	**PA9**	T3C		
	10	**PA10**	T4C		
	11	**PA11**	T5C		
	12	**PA12**	T6C		
	13	**PA13**	T7C		
	14	**PA14**	T8C		AN14
	15	**PA15**	T9C		AN15
PB	0	**PB0**	T0O		
	1	**PB1**	T1O		
	2	**PB2**	T2O		
	3	**PB3**	T3O		
	4	**PB4**	T4O		
	5	**PB5**	T5O		
	6	**PB6**	T6O		
	7	**PB7**	T7O		
	8	**PB8**	T8O		
	9	**PB9**	T9O		
	10	**PB10**	SS0		
	11	**PB11**	SCK0		
	12	**PB12**	MOSI0		
	13	**PB13**	MISO0		
	14	**PB14**	SCL0		
	15	**PB15**	SDA0		

Table 4.3. Function selection of the external pins (continued-1)

포트	핀	기능			
		00	01	10	11
PC	0	PC0	***nTRST**		
	1	PC1	***TDI**		
	2	PC2	***TMS (SWDIO)**		
	3	PC3	***TCK (SWCLK)**		
	4	PC4	***TDO (SWO)**		
	5	**PC5**			
	6	PC6	***nRESET**		
	7	**PC7/BOOT**			
	8	**PC8**	RXD0		
	9	**PC9**	TXD0		
	10	**PC10**	RXD2		
	11	**PC11**	TXD2		
	12	**PC12**	STBYO		
	13	**PC13**	CLKO		
	14	**PC14**	XTALO		
	15	**PC15**	XTALI	CLKIN	
PD	0	**PD0**	PWMA0		
	1	**PD1**	PWMA1		
	2	**PD2**	PWMA2		
	3	**PD3**	PWMA3		
	4	**PD4**	PWMA4		
	5	**PD5**	PWMA5		
	6	**PD6**	PWMA6		
	7	**PD7**	PWMA7		
	8	**PD8**	SS1		
	9	**PD9**	SCK1		
	10	**PD10**	MOSI1		
	11	**PD11**	MISO1		
	12	**PD12**	RXD1		
	13	**PD13**	TXD1		
	14	**PD14**	SCL1		
	15	**PD15**	SDA1		

Table 4.4. Function selection of the external pins (continued-2)

포트	핀	기능			
		00	01	10	11
PE	0	**PE0**	PWM0B		
	1	**PE1**	PWM1B		
	2	**PE2**	PWM2B		
	3	**PE3**	PWM3B		
	4	**PE4**	PWM4B		
	5	**PE5**	PWM5B		
	6	**PE6**	PWM6B	RXD3	
	7	**PE7**	PWM7B	TXD3	
	8	**PE8**	SXIN		
	9	**PE9**	SXOUT		
	11	**PE11**	TraceD3		
	12	**PE12**	TraceD2		
	13	**PE13**	TraceD1		
	14	**PE14**	TraceD0		
	15	**PE15**	TraceCLK		
PF	0	**PF0**			AN8
	1	**PF1**			AN9
	2	**PF2**			AN10
	3	**PF3**			AN11
	4	**PF4**			AN12
	5	**PF5**			AN13
	7	**PF7**			
	8	**PF8**			
	9	**PF9**			
	10	**PF10**			
	11	**PF11**			

4.4 Register Map

The base address of the PMC block is 0x4000_0200 and it is connected to the APB bus. The register map is shown in the table below.

Table 4.5. Address for each port

Port	Address
PA	0x4000_0200
PB	0x4000_0220
PC	0x4000_0240
PD	0x4000_0260
PE	0x4000_0280
PF	0x4000_02A0

Table 4.6. Register map of PMC block

Register	Offset	Access Type	Description	Initial Value	Ref
PA_MR	0x00	RW	Port A Pin MUX Register	0x00000000	4.5.1
PB_MR	0x00	RW	Port B Pin MUX Register	0x00000000	4.5.2
PC_MR	0x00	RW	Port C Pin MUX Register	0x00001155	4.5.3
PD_MR	0x00	RW	Port D Pin MUX Register	0x00000000	4.5.4
PE_MR	0x00	RW	Port E Pin MUX Register	0x00000000	4.5.5
PF_MR	0x00	RW	Port F Pin MUX Register	0x00000000	4.5.6
Pn_CR	0x04	RW	Port n Control Register (Except PC)	0xFFFFFFFF	4.5.7
PC_CR	0x04	RW	Port C Control Register	0xFFFFECAA	4.5.8
Pn_PCR	0x08	RW	Port n Pull-up/Pull-down Control Register (Except PC)	0x00000000	4.5.9
PC_PCR	0x08	RW	Port C Pull-up/Pull-down Control Register	0x0000004F	4.5.10
Pn_DER	0x0C	RW	Port n Debounce Enable Register	0x00000000	4.5.11
Pn_IER	0x10	RW	Port n Interrupt Enable Register	0x00000000	4.5.12
Pn_ISR	0x14	RC	Port n Interrupt Status Register	0x00000000	4.5.13
Pn_ICR	0x18	RW	Port n Interrupt Configuration Reigster	0x00000000	4.5.14
Pn_DPR	0x1C	RW	Port n Debounce Prescaler	0x00000000	4.5.15

4.5 Register Description

4.5.1 PA_MR Port A Pin MUX Register

This register selects a function of the multi-function pins of port A for the purpose.

PA_MR=0x4000_0200

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	**P14**	**P13**	**P12**	**P11**	**P10**	**P9**	**P8**	**P7**	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
00	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**
RW	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**

Port	**Bit Value**			
	00	**01**	**10**	**11**
PA0	PA0			AN0
PA1	PA1			AN1
PA2	PA2			AN2
PA3	PA3			AN3
PA4	PA4			AN4
PA5	PA5			AN5
PA6	PA6	T0C		AN6
PA7	PA7	T1C		AN7
PA8	PA8	T2C		
PA9	PA9	T3C		
PA10	PA10	T4C		
PA11	PA11	T5C		
PA12	PA12	T6C		
PA13	PA13	T7C		
PA14	PA14	T8C		AN14
PA15	PA15	T9C		AN15

4.5.2 PB_MR Port B Pin MUX Register

This register selects a function of the multi-function pins of port B for the purpose.

PB_MR=0x4000_0220

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Port	Bit Value			
	00	01	10	11
PB0	PB0	T0O		
PB1	PB1	T1O		
PB2	PB2	T2O		
PB3	PB3	T3O		
PB4	PB4	T4O		
PB5	PB5	T5O		
PB6	PB6	T6O		
PB7	PB7	T7O		
PB8	PB8	T8O		
PB9	PB9	T9O		
PB10	PB10	SS0		
PB11	PB11	SCK0		
PB12	PB12	MOSI0		
PB13	PB13	MISO0		
PB14	PB14	SCL0		
PB15	PB15	SDA0		

4.5.3 PC_MR Port C Pin MUX Register

This register selects a function of the multi-function pins of port C for the purpose.

어드레스: PC_MR=0x4000_0240

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
00	00	00	00	00	00	00	00	00	01	00	01	01	01	01	01
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Port	Bit Value			
	00	01	10	11
PC0	PC0	nTRST		
PC1	PC1	TDI		
PC2	PC2	TMS (SWDIO)		
PC3	PC3	TCK (SWCLK)		
PC4	PC4	TDO (SWO)		
PC5	PC5			
PC6	PC6	nRESET		
PC7	PC7/BOOT			
PC8	PC8	RXD0		
PC9	PC9	TXD0		
PC10	PC10	RXD2		
PC11	PC11	TXD2		
PC12	PC12	STBYO		
PC13	PC13	CLKO		
PC14	PC14	XTALO		
PC15	PC15	XTALI	CLKIN	

- The initial function of the PC0,PC1,PC2,PC3,PC4 pins are JTAG (nTRST,TDI,TMS,TCK,TDO).
- When PC0/nTRST pin is set to GPIO, JTAG input pins are fixed as below:

Pin Name	Fixed Level	Remark
nTRST	H	JTAG Enable
TDI	H	-
TMS	H	-
TCK	H	-
TDO	X	-

- To use SWD (Serial Wire Debug) interface, the port is handled as follows.

Pin Name	Fixed Level	Remark
SWDIO	H	-
SWCLK	H	-
SWO	X	Optional function

4.5.4 PD_MR Port D Pin MUX Register

This register selects a function of the multi-function pins of port D for the purpose.

PD_MR=0x4000_0260

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Port	Bit Value			
	00	01	10	11
PD0	PD0	PWMA0		
PD1	PD1	PWMA1		
PD2	PD2	PWMA2		
PD3	PD3	PWMA3		
PD4	PD4	PWMA4		
PD5	PD5	PWMA5		
PD6	PD6	PWMA6		
PD7	PD7	PWMA7		
PD8	PD8	SS1		
PD9	PD9	SCK1		
PD10	PD10	MOSI1		
PD11	PD11	MISO1		
PD12	PD12	RXD1		
PD13	PD13	TXD1		
PD14	PD14	SCL1		
PD15	PD15	SDA1		

4.5.5 PE_MR Port E Pin MUX Register

This register selects a function of the multi-function pins of port E for the purpose.

PE_MR=0x4000_0280

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	Reserved	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
00	00	00	00	00	-	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Port	Bit value			
	00	01	10	11
PE0	PE0	PWMB0		
PE1	PE1	PWMB1		
PE2	PE2	PWMB2		
PE3	PE3	PWMB3		
PE4	PE4	PWMB4		
PE5	PE5	PWMB5		
PE6	PE6	PWMB6	RXD3	
PE7	PE7	PWMB7	TXD3	
PE8	PE8	SXIN		
PE9	PE9	SXOUT		
PE11	PE11	TraceD3		
PE12	PE12	TraceD2		
PE13	PE13	TraceD1		
PE14	PE14	TraceD0		
PE15	PE15	TraceCLK		

4.5.6 PF_MR Port F Pin MUX Register

This register selects a function of the multi-function pins of port F for the purpose.

PF_MR=0x4000_02A0

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="2">P11</td><td colspan="2">P10</td><td colspan="2">P9</td><td colspan="2">P8</td><td colspan="2">P7</td><td colspan="2">Reserved</td><td colspan="2">P5</td><td colspan="2">P4</td><td colspan="2">P3</td><td colspan="2">P2</td><td colspan="2">P1</td><td colspan="2">P0</td></tr>
<tr><td colspan="8">-</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">-</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td><td colspan="2">00</td></tr>
<tr><td colspan="8">-</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">-</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">RW</td><td colspan="2">RW</td></tr>
</table>

Port	Bit Value			
	00	01	10	11
PF0	PF0			AN8
PF1	PF1			AN9
PF2	PF2			AN10
PF3	PF3			AN11
PF4	PF4			AN12
PF5	PF5			AN13
PF7	PF7			
PF8	PF8			
PF9	PF9			
PF10	PF10			
PF11	PF11		FRTO_FMF	

4.5.7 Pn_CR Port *n* Control Register (Except PC)

Pn_CR is a register that controls I / O settings for each pin of the port.
push-pull output, open drain output, logic-input and analog input can be set for each pin. This register can set the input/output of GPIO or various peripheral devices. The ADC and Oscillator must be set to analog inputs, in this case the logic part of that pin is disabled.

PA_CR=0x4000_0204, PB_CR=0x4000_0224

PD_CR=0x4000_0264, PE_CR=0x4000_0284, PF_CR=0x4000_02A4

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Pn	00	Push-pull output
	01	Open-drain output
	10	Logic input
	11	Analog input(ADC/OSC)

4.5.8 PC_CR Port C Control Register

PC_CR is a register that controls I / O settings for each pin of the port C.
push-pull output, open drain output, logic-input and analog input can be set for each pin. This register can set the input/output of GPIO or various peripheral devices. The Oscillator must be set to analog inputs, in this case the logic part of that pin is disabled.

PC_CR=0x4000_0244

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
11	11	11	11	11	11	11	11	11	10	11	00	10	10	10	10
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PC	00	Push-pull output
	01	Open-drain output
	10	Logic input
	11	Analog input(ADC/OSC)

4.5.9 Pn_PCR Port n Pull-up/Pull-down Control Register (Except PC)

Pn_PCR is a register to selects the pull-up option of each pin of port C. Pull-up resistor for each pin can be turned on or off. Lower 16-bits are used to determine whether pull-up/pull-down is enable or disable, and Upper 16-bits are used to determine use of pull-up or pull-down resistor.

PA_PCR=0x4000_0208, PB_PCR=0x4000_0228

PD_PCR=0x4000_0268, PE_PCR=0x4000_0288, PF_PCR=0x4000_02A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Dn	Select Pull-up or Pull-down Registor	
	0	Select Pull-up Resistor
	1	Select Pull-down Resistor
Pn	Enable or Disable Pull-up/Pull-Down	
	0	Pull-up/Pull-down Disable
	1	Pull-up/Pull-down Enable

4.5.10 PC_PCR Port C Pull-up/Pull-down Control Register

Pn_PCR is a register to selects the pull-up option of each pin of port C. Pull-up resistor for each pin can be turned on or off. Lower 16-bits are used to determine whether pull-up/pull-down is enable or disable, and Upper 16-bits are used to determine use of pull-up or pull-down resistor.

PC_PCR=0x4000_0248

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1

Dn	Select Pull-up or Pull-down Registor	
	0	Select Pull-up Resistor
	1	Select Pull-down Resistor
Pn	Enable or Disable Pull-up/Pull-Down	
	0	Pull-up/Pull-down Disable
	1	Pull-up/Pull-down Enable

4.5.11 Pn_DER Port *n* Debounce Enable Register

This register enables the function of debounce for each pin.

PA_DER=0x4000_020C, PC_DER=0x4000_022C, PC_DER=0x4000_024C

PD_DER=0x4000_026C, PE_DER=0x4000_028C, PF_DER=0x4000_02AC

31	30	29	28	27	26	25	24	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																P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
-																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
-																0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1

Pn	Enable/Disable Debounce of pin	
	0	Disable Debounce of Pn
	1	Enable Debounce of Pn

4.5.12 Pn_IER Port *n* Interrupt Enable Register

These registers are for enabling of the GPIO interrupt for each pin and can set the interrupt type such as level interrupt or edge interrupt. Level interrupt occur when a high or low level signal is detected through the input pin. The edge interrupt occurs when an edge that made is detected when the input signal changes from low to high or from high to low.

PA_IER=0x4000_0210, PB_IER=0x4000_0230, PC_IER=0x4000_0250

PD_IER=0x4000_0270, PE_IER=0x4000_0290, PF_IER=0x4000_02B0

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Pn	Setting of interrupt enable/disable	
	x0	Disable GPIO interrupt
	01	Enable level interrupt (‘H’ or ‘L’l)
	11	Enable edge interrupt (‘H→L’ or ‘L→H’)

4.5.13 Pn_ISR Port *n* Interrupt Status Register

This register indicates the GPIO interrupt status for all pins of each port. All pins of each port can be GPIO interrupt source. The interrupt signal generated onthe port is sent to the interrupt controller.
Because interrupt controller just can detect ‘H’ level only, PMC block sends the signal generated level interrupt or edge interrupt, or ‘H’ or ‘L’ signal interrupt by trigger to interrupt controller.
Since the interrupt controller can detect the interrupt signal only at the 'H' level, the PMC block sends the interrupt signal generated from all triggers ('H' or 'L') of the level or edge to the interrupt controller.
When the port interrupt is set to edge interrupt, if an interrupt event occurs, the event information is displayed in the status register. If the port interrupt is set to level interrupt, the event information is not displayed in the status register.
To initialize the state of any port in this register, a ‘1’ must be written to the corresponding bit.

PA_ISR=0x4000_0214, PB_ISR=0x4000_0234, PC_ISR=0x4000_0254

PD_ISR=0x4000_0274, PE_ISR=0x4000_0294, PF_ISR=0x4000_02B4

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC

Pn	indicates an interrupt request according to the condition of each bit. Write ‘1’ to the flag to clear the flag.	
	00	No interrupt occurred
	01	‘L’ level or falling edge interrupt occurred
	10	‘H’ level or rising edge interrupt occurred
	11	Either of rising or falling edge interrupt occurred (Not available on level mode interrupt)

4.5.14 Pn_ICR Port *n* Interrupt Control Register

Pn_ICR register is to control interrupt type of each pin. Interrupt type is

PA_ICR=0x4000_0218, PB_ICR=0x4000_0238, PC_ICR=0x4000_0258

PD_ICR=0x4000_0278, PE_ICR=0x4000_0298, PF_ICR=0x4000_02B8

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	**P14**	**P13**	**P12**	**P11**	**P10**	**P9**	**P8**	**P7**	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
00	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**	**00**
RW	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**	**RW**

Pn	Select interrupt occurrence type of each pin	
	00	Disable interrupt
	01	‘L’ level or falling edge interrupt input
	10	‘H’ level or rising edge interrupt input
	11	Either of rising or falling edge interrupt (Not available on level mode interrupt)

4.5.15 Pn_DPR Port *n* Debounce Prescaler

PnDPR is a 5-bit register that divides the debounce reference clock for each port.

PC_DPR=0x4000_021C, PB_DPR=0x4000_023C, PC_DPR=0x4000_025C

PD_DPR=0x4000_027C, PE_DPR=0x4000_029C, PF_DPR=0x4000_02BC

31	30	29	28	27	26	25	24	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																											DPR				
-																											00000				
-																											RW				

4 **0**	**DPR**	The debounce clock input divider. The debounce clock can be obtained by dividing the main clock by the DPR squared of 2, as in the following formula:

$$Debouce\ Clock = \frac{PCLK}{2^{DPR}}$$

Input debounce clock function is specified for an input signal based on four-consecutive clock When you get the same value as the input values for the recognition function. Therefore, the input filter time,

$$Tfilter = \frac{4}{Debounce\ Clock}$$

Debounce filter window of the prescaler in the following table.

Table 4.7. The examples of filtering windows of each debounce prescale value

DPR	PCLK Count	Filtering window (16MHz)	Filtering window (75MHz)	DPR	PCLK Cout	Filtering window (16MHz)	Filtering window (75MHz)
0	4	250ns	53ns	16	256K	16ms	3.4ms
1	8	500ns	106ns	17	512K	32ms	6.9ms
2	16	1µs	213ns	18	1024K	65ms	13.9ms
3	32	2µs	426ns	19	2048K	131ms	27.9ms
4	64	4µs	853ns	20	4096K	262ms	55.9ms
5	128	8µs	1.7µs	21	8192K	524ms	111ms
6	256	16µs	3.4µs	22	16M	1.0s	223ms
7	512	32µs	6.8µs	23	32M	2.0s	447ms
8	1024	64µs	13.6µs	24	64M	4.1s	894ms
9	2048	128µs	27.3µs	25	128M	8.3s	1.7s
10	4096	256µs	54.6µs	26	512M	16s	3.5s
11	8192	512µs	109µs	27	1G	33s	7.1s
12	16384	1ms	218µs	28	2G	67s	14.3s
13	32768	2ms	436µs	29	4G	134s	28.6s
14	65536	4ms	873µs	30	8G	268s	57.2s
15	128K	8.1ms	1.74ms	31	16G	536s	114.5s

4.6 Functional Description

4.6.1 Port Function

All pins except a IO function pins of A33G52x can be used as GPIO. To use GPIO, first set the pin MUX register of the corresponding port to GPIO.
When the I/O port is set as the input function by the pin control register, the output function of the I/O port is disabled.
Depending on the function selection of the Pn_MR register and the direction selection of the Pn_PCR register,
The port function can be used differently. The input data register captures the current data or debounced input data to the I/O pins per GPIO clock cycle.

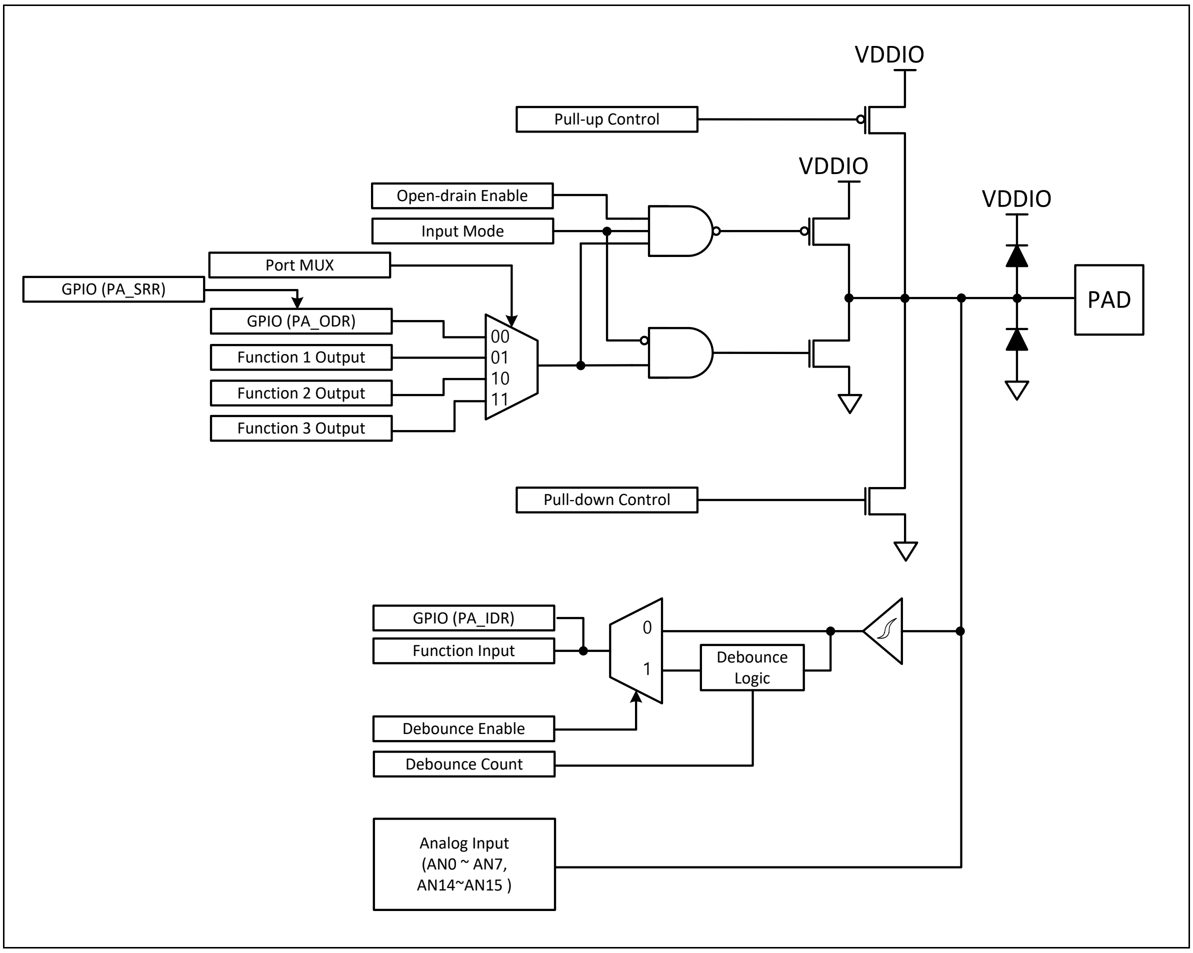


Figure 4.8. Block diagram of PA port (Example)

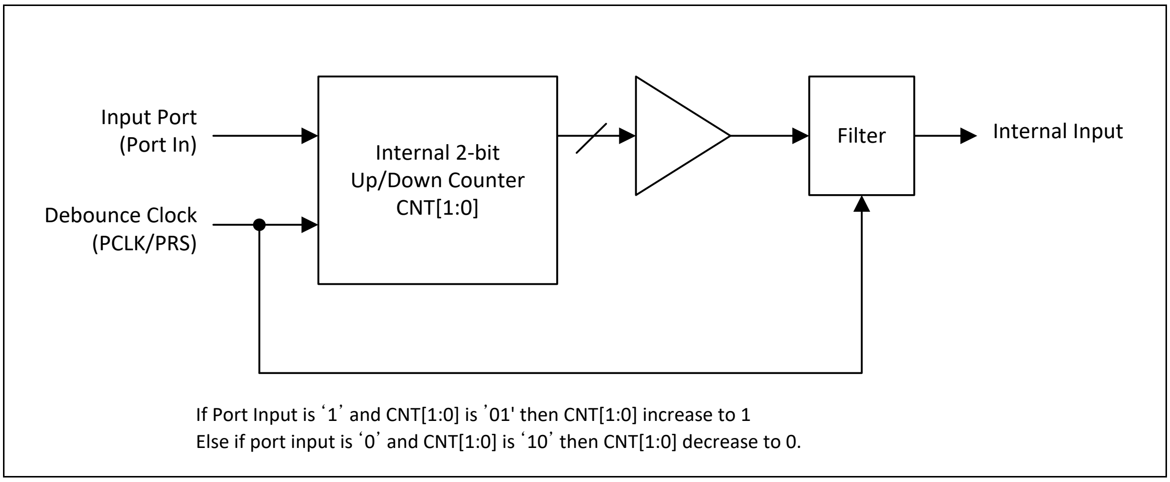
4.6.2 Debounce of Input Port

The input debounce function is supported on each port of the A33G52x. The debounce function is used to filter the interference noise of the input port to filter out the normal signal.

You can set the Pn_DPR register to adjust the filtering level of each 16-pin port and enable / disable for each pin.

The filtering level uses a clock that divides the PCLK clock by a 16-bit counter and can set this value.

The simplified block configuration is shown below.



If Port Input is ‘1’ and CNT[1:0] is ’01' then CNT[1:0] increase to 1
Else if port input is ‘0’ and CNT[1:0] is ‘10’ then CNT[1:0] decrease to 0.

Figure 4.9. Simplified block diagram of a debounce port

When the debounce logic of the logic input is activated, the debounce function operates on the basis of the clock determined by DPR <4: 0> of Pn_DPR.

Debounce will change the input recognition value if the changed input is maintained for a maximum of 3 clocks continuously. If the internal counter is '01', the input is set to '1'. If the internal counter is '10', it is set to '0'.

Otherwise, it retains its previous value without change.

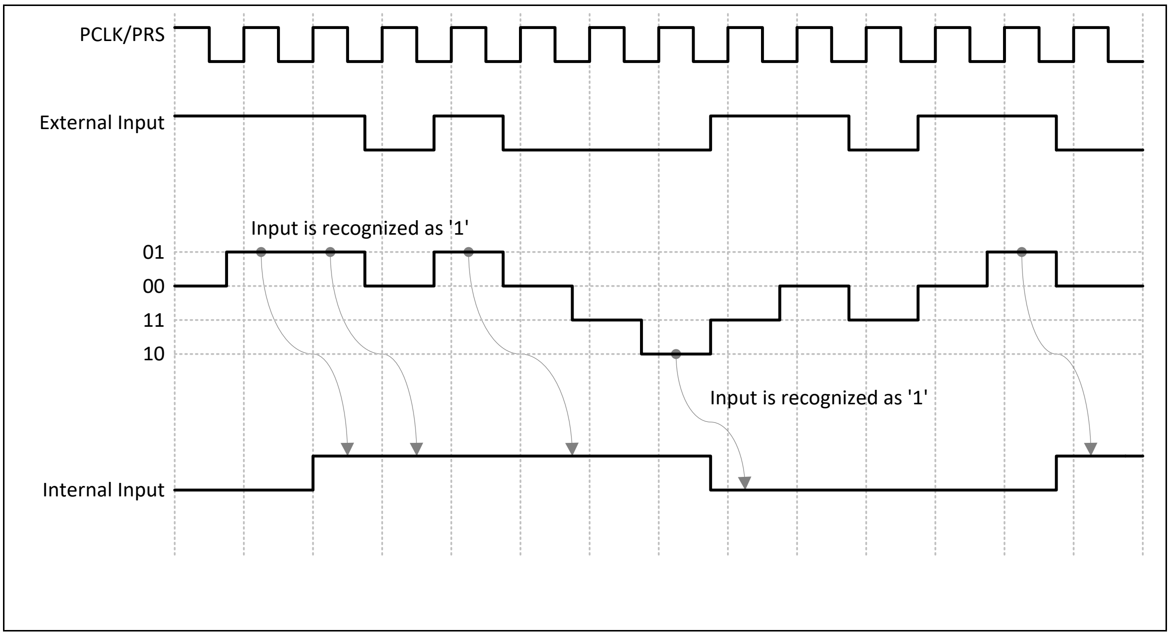


Figure 4.10. The timing of debounce function

4.6.3 GPIO Interrupt

There are six interrupt sources in the GPIO block of the A33G52x, and each port can be an interrupt source. To use an interrupt on any port of the GPIO, set the Pn_IER register to one of two ways to use the interrupt level or edge interrupt. You can set the level or rising edge, falling edge, and rising / falling edge in this next Pn_ICR register. If an interrupt is generated in this way, the interrupt flag bit of the corresponding port can be checked in the interrupt Pn_ISR register. To clear this bit value, write '1' to the correspondng bit of Pn_ISR.

Table 4.8. The pin configuration for each port

Port	Pins	Pin Name	Remark
PA	16	PA0 - PA15	Output or Schmitt-trigger input
PB	16	PB0 - PB15	Output or Schmitt-trigger input
PC	16	PC0 - PC15	Output or Schmitt-trigger input
PD	16	PD0 - PC15	Strong output or CMOS input
PE	15	PE0 - PE9, PE11 - PE15	Output or Schmitt-trigger input
PF	6	PF0 - PF5	Output or Schmitt-trigger input

4.6.4 Setting Example

<Example 1> Example of edge interrupt setting of PA0 input port

PA_MR<P0[1:0]> = “00”	: Sets PA0 port to GPIO
PA_CR<P0[1:0]> = “10”	: Sets the direction of PA0 port to logic input
PA_PCR<P0>= “1”	: Enables pull-up/pull-down of PA0
PA_PCR<D0>= “0”	: Enables pull-up resistor of PA0
PA_IER<P0[0:1]> = “11”	: Enables edge interrupt of PA0
PA_ICR<P0[0:1]> = “11”	: Enables Rising and Falling edge interrupt
PA_ISR<P0[0:1]> = “11”	: Enables rising and falling edge interrupt event display

<Example 2> Example of debounce of PA port input port

PA_DPR<DPR[0:4]> = “00000”	: Sets debounce prescaler of PA port (53ns @75MHz)
PA_DER<P0>= “1	: Enables debounce function of PA0 port

CHAPTER 5. GPIO (General Purpose I/O)

Table 5.1. Operation Summary

Item	Pin Name	Remark
Clock usage	PCLK	
Reset Source	Set by PMU_PER	
Reset Generation	None	
Interrupt Generation	None	Included in PMC
Interrupt Clear Method	None	

5.1 Overview

The A33G52x can be used for application purposes by setting all pins to GPIO (General Purpose Input Output) except the VDD and GND pins. Each port consists of 16 bits. When the GPIO pin is set to output mode, the signal can be output by setting the value of each bit to 'H' or 'L'. When the GPIO pin is set to logic input, the signal input status can be checked by the corresponding bit value of each pin.

Features of GPIO (General Purpose Input Output)

- Output value (High/Low) selection function of each pin
- Function to check logic input status of each pin
- General Purpose GPIO Usage
- Pin function selection by PMC register

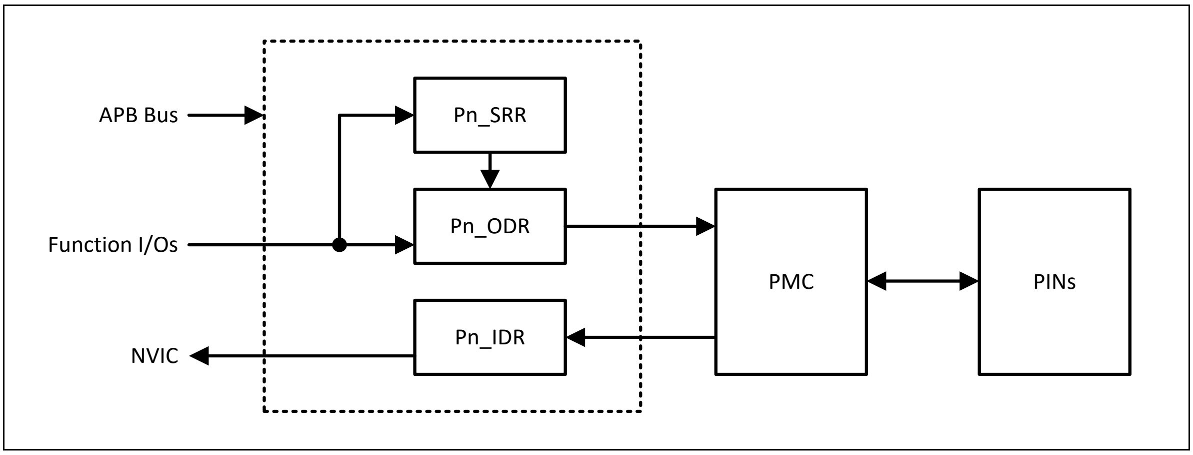


Figure 5.1. Block diagram of GPIO

5.2 Block Diagram

5.2.1 PA Port

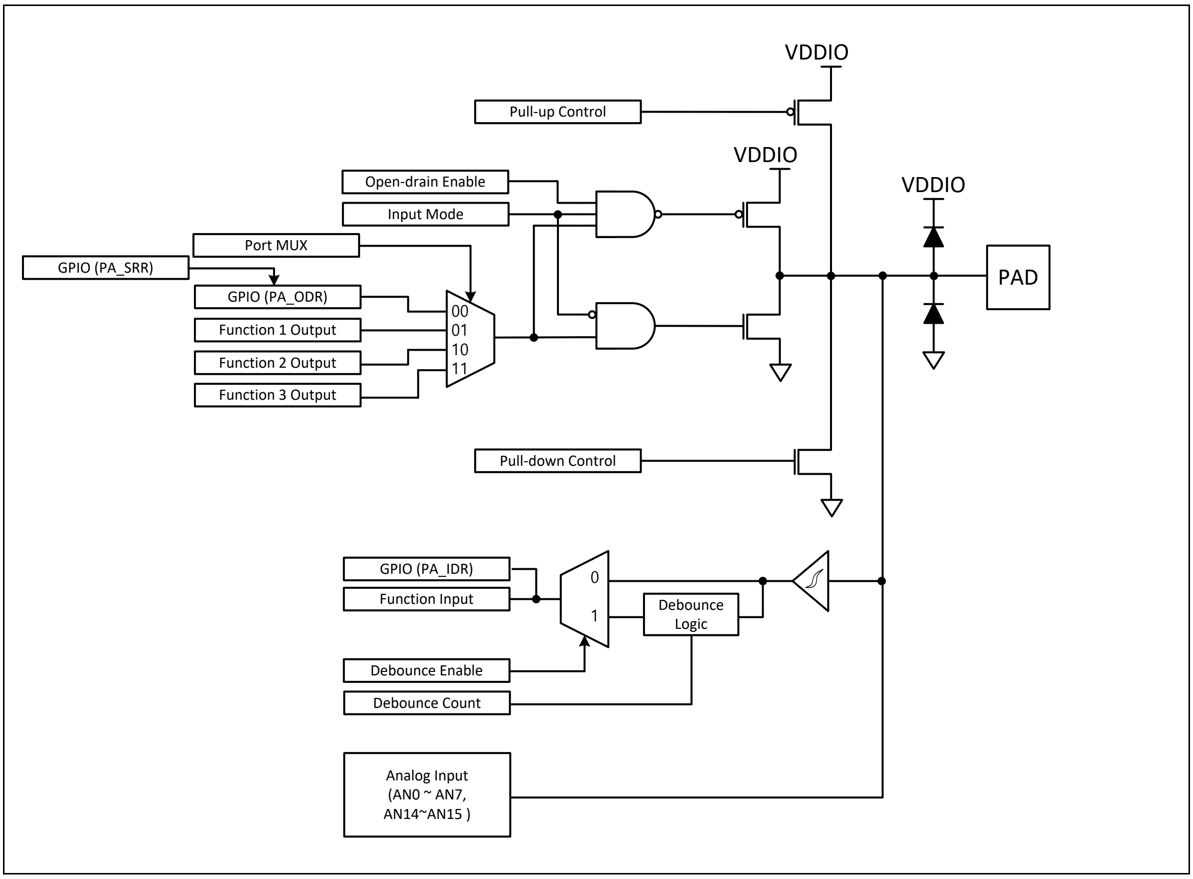


Figure 5.2. Block diagram of PA port

5.2.2 PB Port

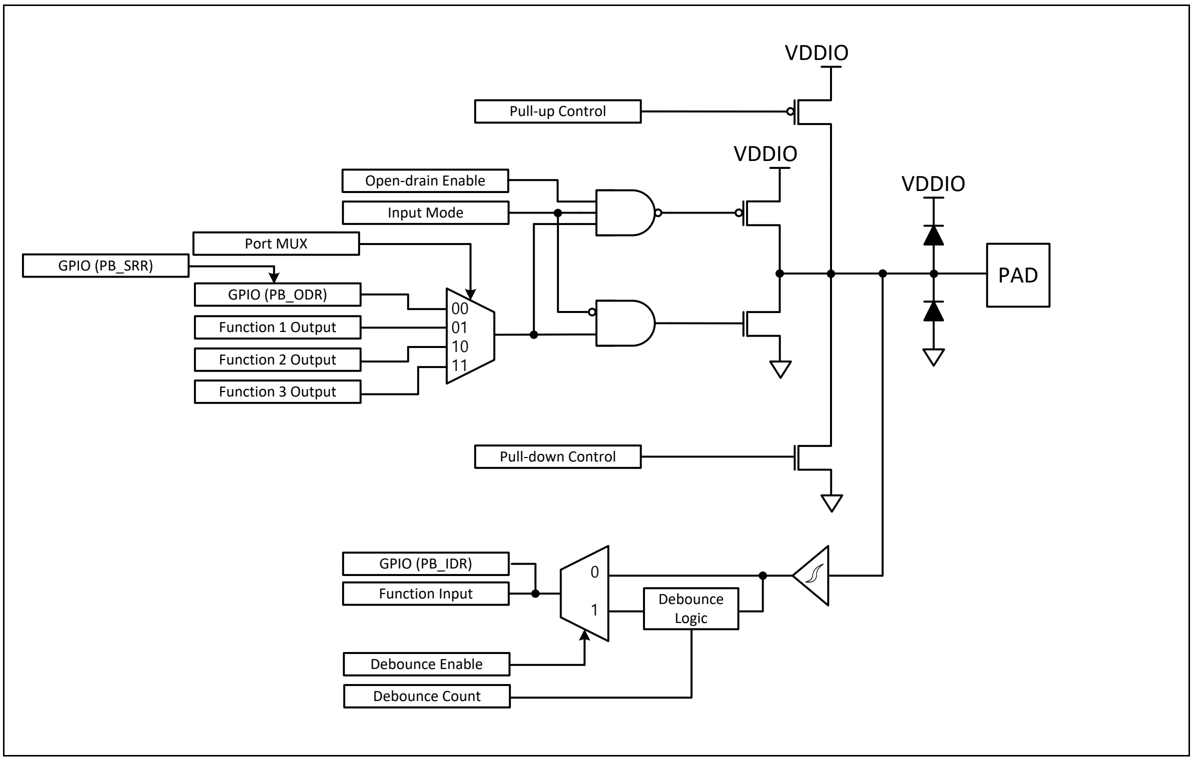


Figure 5.3.Block diagram of PB port

5.2.3 PC Port

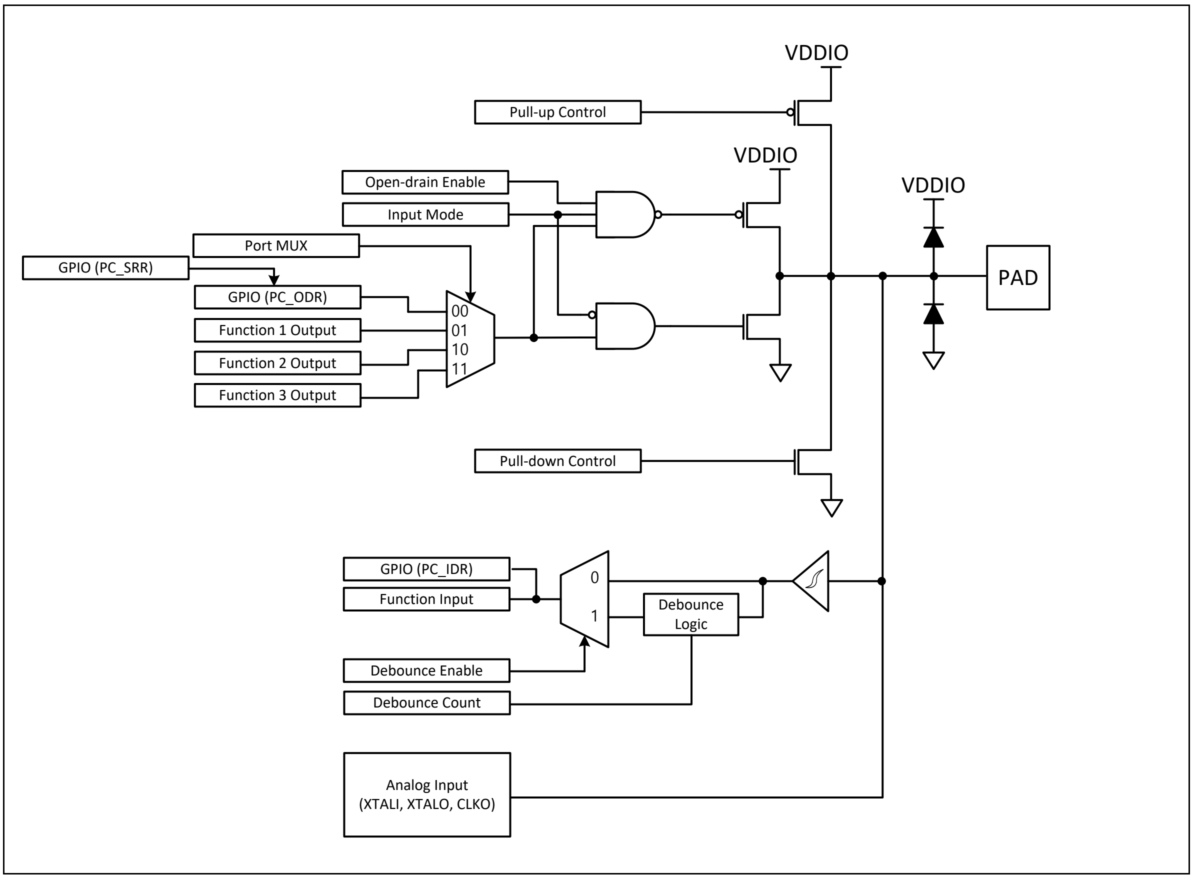


Figure 5.4. PC Block diagram of PC port

5.2.4 PD Port

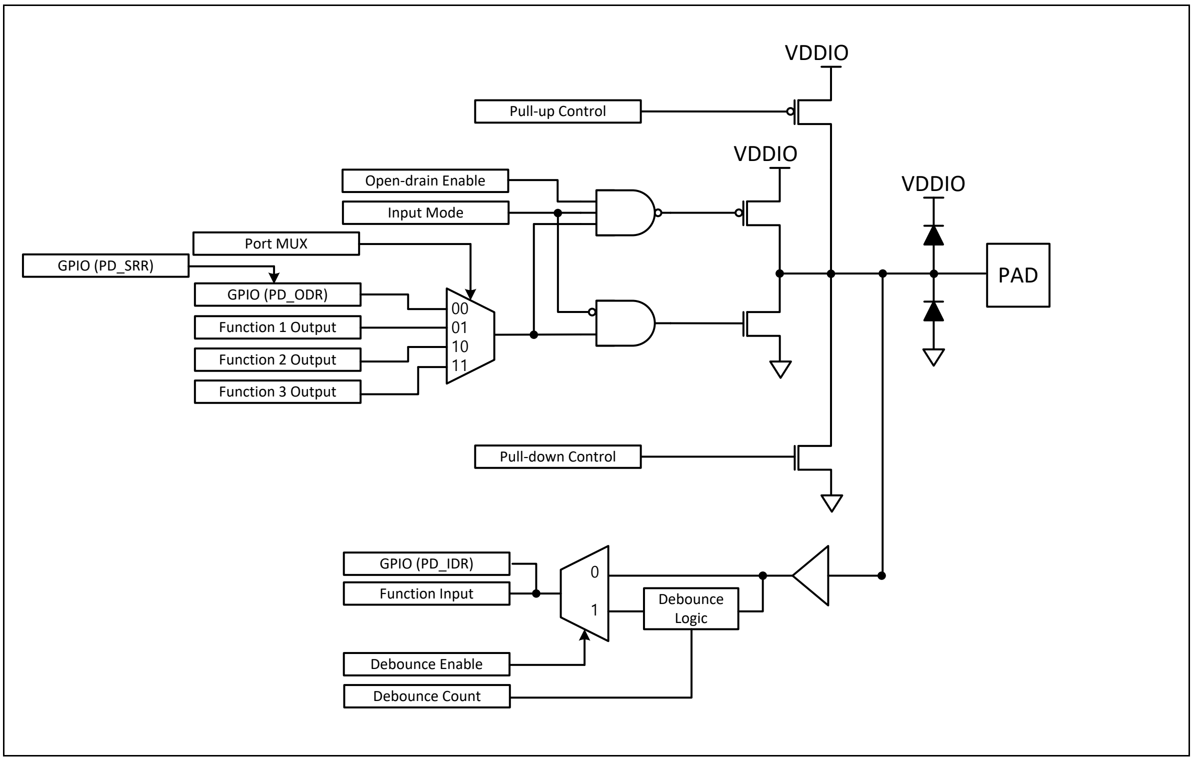


Figure 5.5. Block diagram of PD port

5.2.5 PE Port

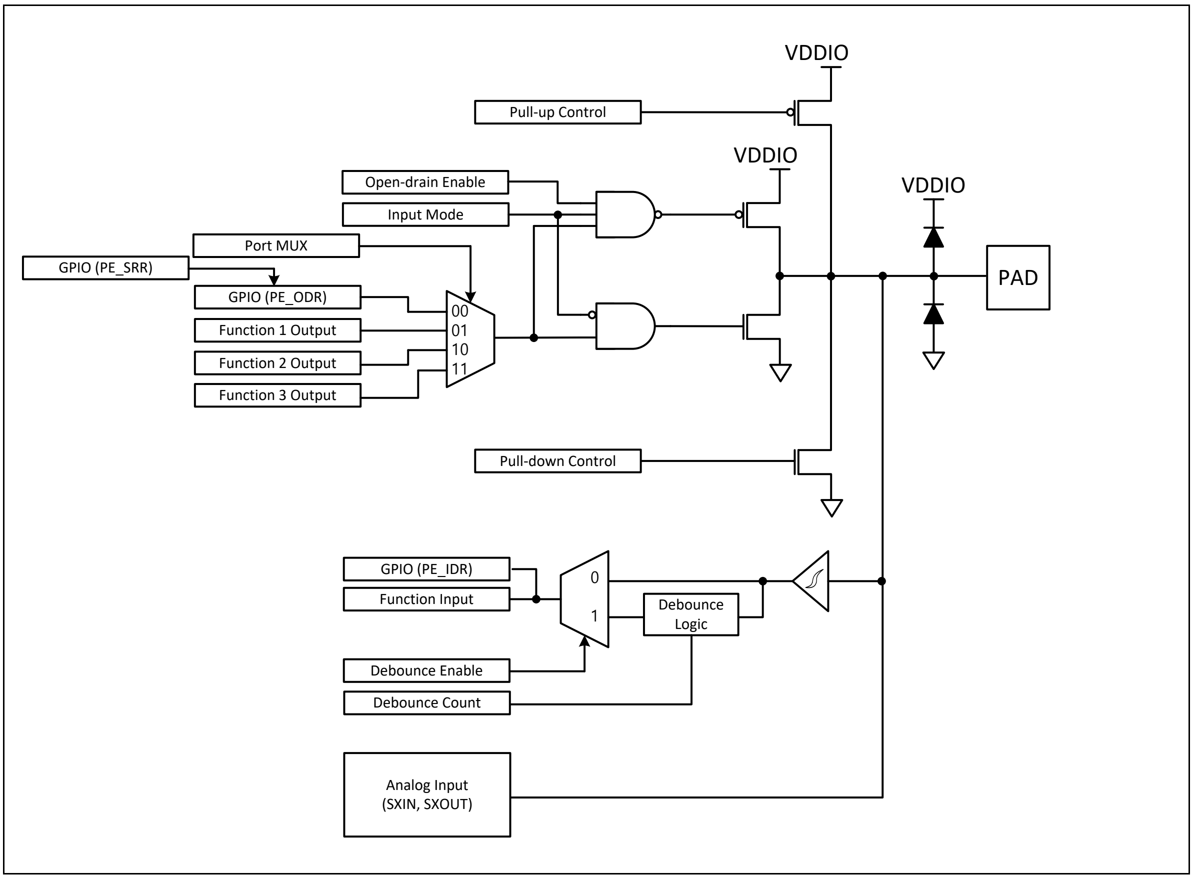


Figure 5.6. Block diagram of PE port

5.2.6 PF Port

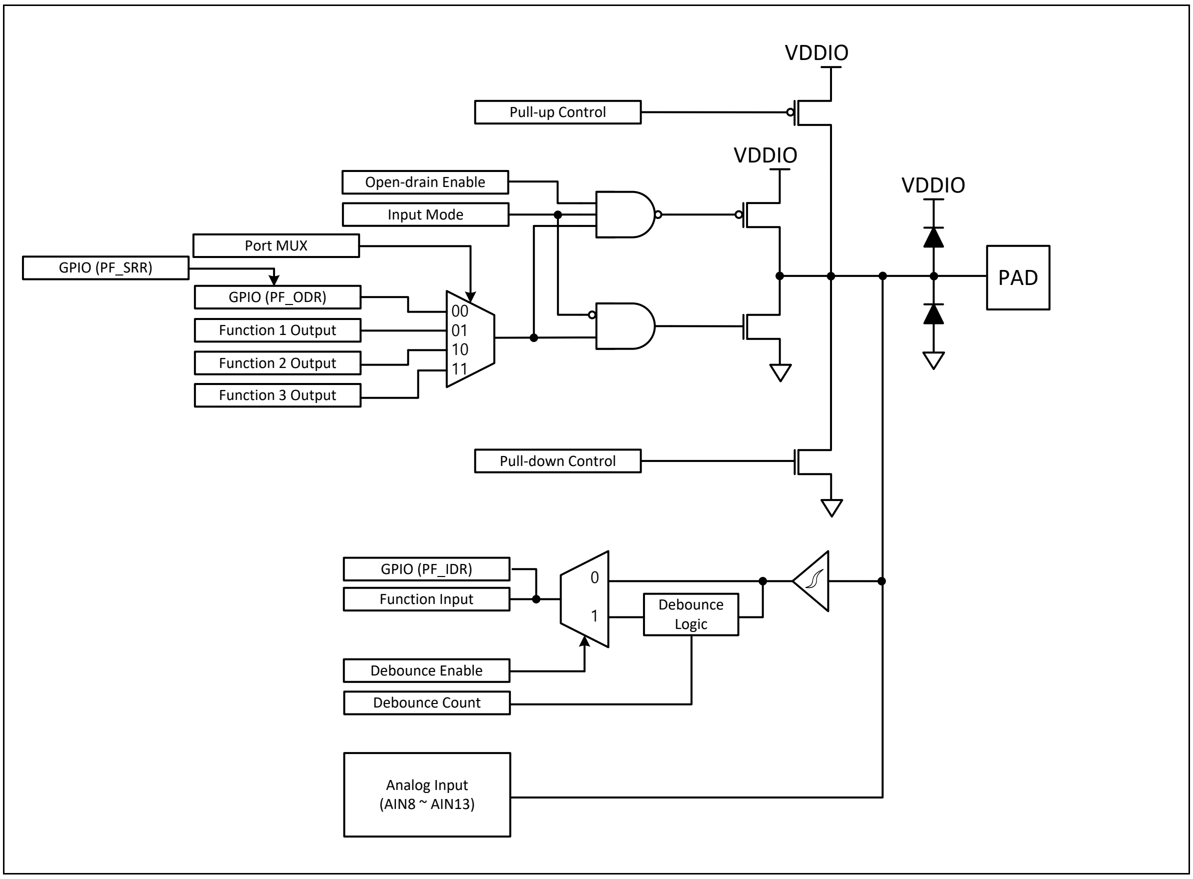


Figure 5.7. Block diagram of PF port

5.3 Pin Configuration

Most of the I/O pins of the A33G52x are multifunctional pins that perform more than two functions per pin. GPIO (General Purpose Input Output) output or input can be used after setting the pin function and direction of GPIO in PMC (Port Map Controller) block.

Table 5.2. The pin configuration for each port

Port	Pins	Pin Name	Remark
PA	16	PA0 - PA15	Output or Schmitt-trigger input
PB	16	PB0 - PB15	Output or Schmitt-trigger input
PC	16	PC0 - PC15	Output or Schmitt-trigger input
PD	16	PD0 - PC15	Strong output or CMOS input
PE	15	PE0 - PE9, PE11 - PE15	Output or Schmitt-trigger input
PF	6	PF0 - PF5, PF7 – PF11	Output or Schmitt-trigger input

5.4 Register Map

The base address of the General Purpose Input Output (GPIO) block is 0x4000_0300 and is connected to the APB bus that a slow bus connecting peripherals such as I / O on the Advanced Microcontroller Bus Architecture (AMBA) bus. The register map is shown in the table below.

Table 5.3. The Address map for each port

Port	Address
PA	0x4000_0300
PB	0x4000_0310
PC	0x4000_0320
PD	0x4000_0330
PE	0x4000_0340
PF	0x4000_0350

Table 5.4. Register map of GPIO block

Register	Offset	Access Type	Description	Initial Value	Ref
PA_ODR	0x00	RW	Port A Output Data Register	0x00000000	5.5.1
PA_IDR	0x04	RO	Port A Intput Data Register	0x----	5.5.2
PA_SRR	0x08	WO	Port A Pin Set/Reset Register	0x----	5.5.3
PB_ODR	0x10	RW	Port B Output Data Register	0x00000000	5.5.1
PB_IDR	0x14	RO	Port B Intput Data Register	0x----	5.5.2
PB_SRR	0x18	WO	Port B Pin Set/Reset Register	0x----	5.5.3
PC_ODR	0x20	RW	Port C Output Data Register	0x00000000	5.5.1
PC_IDR	0x24	RO	Port C Intput Data Register	0x----	5.5.2
PC_SRR	0x28	WO	Port C Pin Set/Reset Register	0x----	5.5.3
PD_ODR	0x30	RW	Port D Output Data Register	0x00000000	5.5.1
PD_IDR	0x34	RO	Port D Intput Data Register	0x----	5.5.2
PD_SRR	0x38	WO	Port D Pin Set/Reset Register	0x----	5.5.3
PE_ODR	0x40	RW	Port E Output Data Register	0x00000000	5.5.1
PE_IDR	0x44	RO	Port E Intput Data Register	0x----	5.5.2
PE_SRR	0x48	WO	Port E Pin Set/Reset Register	0x----	5.5.3
PF_ODR	0x50	RW	Port F Output Data Register	0x00000000	5.5.1
PF_IDR	0x54	RO	Port F Intput Data Register	0x----	5.5.2
PF_SRR	0x58	WO	Port F Pin Set/Reset Register	0x----	5.5.3

5.5 Register Description

5.5.1 Pn_ODR Port n Output Data Register

When the bit of the corresponding port is set as the output, the value of this register is output to each port. This register outputs a signal when the corresponding pin is set to GPIO.

PA_ODR=0x4000_0300, PB_ODR=0x4000_0310, PC_ODR=0x4000_0320

PD_ODR=0x4000_0330, PE_ODR=0x4000_0340, PF_ODR=0x4000_0350

31	30	29	28	27	26	25	24	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																ODR															
-																00000000 00000000															
-																RW															

15 0	**ODR**	If the port is output mode, ODR value is output.

5.5.2 Pn_IDR Port n Input Data Register

This register indicates the current input state of the port and indicates the input when the GPIO and pins are set to logic inputs of other functions. When the pin is set to a special function such as ADC or clock, '1' is always displayed.
If the level interrupt function is enabled on all ports via the PMC_ISR register, the level value can be checked by reading the Pn_IDR register value for the input level value.

PA_IDR=0x4000_0304, PB_IDR=0x4000_0314, PC_IDR=0x4000_0324

PD_IDR=0x4000_0334, PE_IDR=0x4000_0344, PF_IDR=0x4000_0354

31	30	29	28	27	26	25	24	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																**IDR**															
-																**XXXXXXXX XXXXXXXX**															
-																**RO**															

15 **0**	**IDR**	Indicates the current status of a port.

Note) If the external pin of the MCU is set as an input when it is in the floating state, it will be in an unknown state that recognizes the level as 0 or 1 regardless of the value of the input signal. In this floating state, determining the level value recognized from the Pn_IDR register and executing the program to perform certain functions may malfunction. Therefore, It is recommended to design a pull-up or pull-down for pins that receive signals from the outside.

5.5.3 Pn_SRR Port n Set/Reset Register

This register can be set or reset for the value of each pin of the corresponding port. This register can only change the PnODR value to a specific bit. If the values of BRR and BSR are '1' at the same time, the current output value is inverted and output.

PA_SRR=0x4000_0308, PB_SRR=0x4000_0318, PC_SRR=0x4000_0328

PD_SRR=0x4000_0338, PE_SRR=0x4000_0348, PF_SRR=0x4000_0358

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR																BSR															
XXXXXXXX XXXXXXXX																XXXXXXXX XXXXXXXX															
WO																WO															

31 **16**	**BRR**	If the bit is '1', the corresponding bit resets.
15 **0**	**BSR**	If the bit is '1', the corresponding bit sets.

Note) When BSR and BRR are set to '1' at the same time, the output of the corresponding pin is inverted.

5.6 Functional Description

All pins other than A33G52x function pins can be set to GPIO (General Purpose Input Output). To use GPIO, first set the pin MUX register on the port map controller (PMC) of that port to GPIO.

5.6.1 GPIO Interrupt

The GPIO block has six interrupt sources, and each port can be an interrupt source. To use each port of the GPIO as an interrupt source, set the edge or level and the polarity according to the interrupt generation method, and then set the interrupt enable register. Interrupts can be set for each pin by level or edge method. When an interrupt occurs, the corresponding bit of the port status register (PnISR) can be set to '1' to clear the interrupt flag.

Table 5.5 The Interrupt Pins of the GPIO block

Interrupt Source	Corresponding Pin
PA	Port A[15:0]
PB	Port B[15:0]
PC	Port C[15:0]
PD	Port D[15:0]
PE	Port E[15:0]
PF	Port F[11:0]

5.6.2 Setting Example

<Example 1> Function of PD0 pin, input / output direction, pull-up / pull-down resistor setting

PD_MR<P0[1:0]> = “00”	: Set PD0 Pin to GPIO (General Purpose I/O).
PD_CR<P0[1:0]> = “00”	: Set the directionof PD0 pin to push-pull output.
PD_PCR<P0>= “1”	: Enable the pull-up/pull-down function of PD0 pin.
PD_PCR<D0>= “0”	: Select pull-up resistor of PD0 pin.

<Example 2> Output data setting of PD0 pin

PD_MR<P0[1:0]> = “00”	: Set PD0 Pin to GPIO (General Purpose I/O).
PD_CR<P0[1:0]> = “00”	: Set the directionof PD0 pin to push-pull output.
PD_ODR<ODR[15:0]> = “00000000 00000001”	: Set the output signal of PD0 pin to high.

<Example 3> PD0 value set / reset

PD_SRR<BSR[15:0]> = “00000000 00000001”	: When the BSR bit of the PD0 port is '1', the bit value 'set'
PD_SRR<BRR[31:16]> = “00000000 00000001”	: When the BRR bit of the PD0 port is '0', the bit value 'reset'

CHAPTER 6. 32-bit WDT

Table 6.1. Operation Summary

Item	Pin Name	Remark
Clock usage	RingOSC, IOSC16, PCLK, Main XTAL, Sub XTAL	Select by PMU_PCSR Built-in Prescler
Reset Source	Set by PMU_PER	
Reset Generation	When WDT counter value reaches '0'	Set by WDT_CON
Interrupt Generation	When WDT counter value reaches '0' (WDT(3))	Set by WDT_CON
Interrupt Clear Method	Writng non-'0' value to the WDT_CVR	WDT_CVR

6.1 Overview

WDT (Watschdog Timer) monitors the operation of the MCU and is typically used to detect the occurrence of software errors. When MCU is out of control due to a malfunction, the watchdog timer generates reset to exit from the out of control. A33G52x Watchdog Timer consists of a 32-bit down counter. If the watchdog timer was selected as a reset source, when current downcount value register counts down to '0', the MCU restarts. If the MCU monitoring function is not used, the WDT can be used as a periodic timer with interrupt.

Features of WDT (Watchdog Timer)

- 32-bit down-counting timer
- Reset by underflow of the WDT
- Interrupt with periodic timer or underflow
- Selectable input clock of the WDT
  - PCLK
  - Selected clock source in PMU_PCSR<WDTCS[1:0]> : Main XTAL, IOSC16, SXOSC, RINGOSC
- Support 8-Step prescaler for the WDT input clock
- Selectable operation mode of the WDT in debug-mode

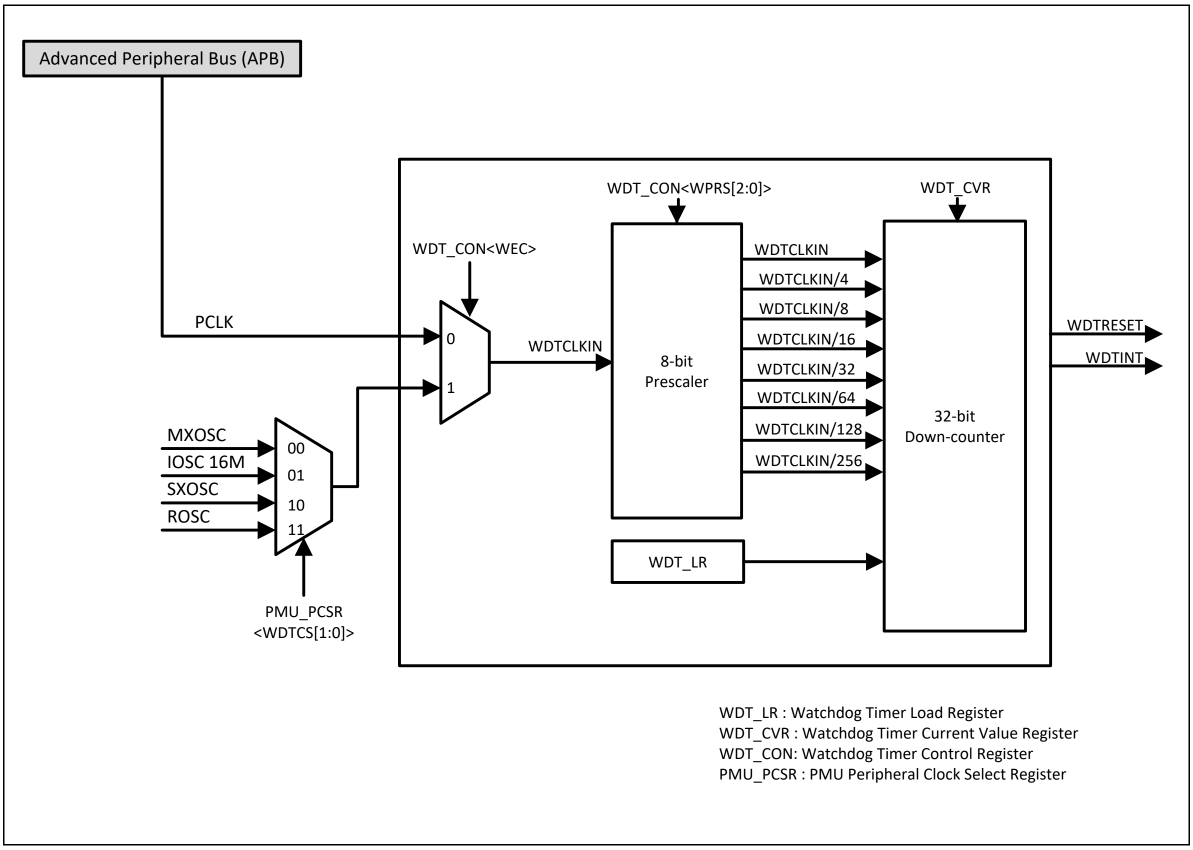


Figure 6.1. Block diagram

6.2 Register Map

The Refference Address of the WDT is 0x4000_0400. Overall Memory Map is shown as Table 6.2.

Table 6.2. Register map of the WDT

Name	Offset	R/W	Description	Initial value	Ref.
WDT_LR	0x00	RW	Watchdog load register	0x00000000	6.3.1
WDT_CVR	0x04	RO	Watchdog current value register	0x0000FFFF	6.3.2
WDT_CON	0x08	RW	Watchdog control register	0x00000047	6.3.3

6.3 Register Description

6.3.1 WDT_LR Watchdog Timer load register

The WDT_LR register is a register for changing of the WDT_CVR value. To change of the WDT_CVR value, 2 conditions must be met. First, WEN(Watchdog enable) of WTD_CON should be ‘1’, and second, WDT_LR should be written. If WEN of WDT_CON is '0', it is not reflected in WDT_CVR even if WDT_LR value was written, but when WEN becomes '1', the value is reflected in WDT_CVR.

When using WDT (Watchdog Timer) as a reset source, a value of WDT_LR must be written before WDT_CVR value goes ‘0’ to prevent reset.

The event of WDT is generated in transition state of the WDT_CVR value from ‘1’ to ‘0’. Therefore, when WDT is used as reset mode, the count value is written to WDTLR as it is, but when WDT is used as interrupt mode, the writing value should be subtracted 1 from the calculated value.

WDT_LR=0x4000_0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTLR																															
00000000 00000000 00000000 00000000																															
RW																															

31 0	**WDTLR**	WDT (Watchdog Timer) Load Value

6.3.2 WDT_CVR Watchdog Timer Curret Value Register

WDT_CHR is a 32bits-read-only register to indicate the curret value of WDT. To change the value, WDT_LR should be written when WEN is ‘1’. If WDT is used as reset source, reset is generated when WDT_CVR becomes ‘0’ and if WDT is used as periodic timer, interrupt ist generated when WDT_CVR becomes ‘0’. To use WDT for reset, WRE of WDT_CON and WDTRSTE(Bit 3) of PMU_RSER should be set to ‘1’

WDT_CVR=0x4000_0404

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTCVR																															
00000000 00000000 11111111 11111111																															
RO																															

31 0	**WDTCVR**	WDT (Watchdog Timer) Current Count Value

6.3.3 WDT_CON Watchdog Timer Control Register

WDT_CON is a register to control overall behavior of Watchdog Timer

WDT_CON=0x4000_0408

31	30	29	28	27	26	25	24	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																WDH	Reserved						WOF	WIE	WRE	WEN	Reserved	WEC	WPRS		
-																0	-						0	0	1	0	-	0	111		
-																RW	-						RO	RW	RW	RW	-	RW	RW		

15	**WDH**	0	Enable counter in Watchdog Timer in debug-mode
		1	Disable counter in Watchdog Timer in debug-mode
8	**WOF**	0	Disable underflow event
		1	Enable underflow event(Cleard by writing a non-‘0’ value to WDTCVR via WDTLR)
7	**WIE**	0	Disable underflow interrupt
		1	Enable underflow interrupt
6	**WRE**	0	Deactivation reset when underflow
		1	Activation reset when unterflow
5	**WEN**	0	Stop Watchdog Timer counter
		1	Run Watchdog Timer counter
3	**WEC**	0	Using PCLK as clock source
		1	Using a clock source which is set by PMU_PCSR
2 **0**	**WPRS**	Select a prescaler value of WDT clock	
		000	WDTCLKIN
		001	WDTCLKIN/4
		010	WDTCLKIN/8
		011	WDTCLKIN/16
		100	WDTCLKIN/32
		101	WDTCLKIN/64
		110	WDTCLKIN/128
		111	WDTCLKIN/256

Note) If prescaler value of watch-dog timer counter or input frequency is set low, interrupt be cleared per period of prescaled input clock.

If WDT (Watchdog Timer) operates as periodic timer and if the value of WDT_CVR is 0xFFFF(65535), the period of Interrupt can be calculated as fallows. In this case the PCLK is assumed to be 16MHz.

- Pulse period : 1/16MHz = 0.0625 µsec (WPRS=000)
- Interrupt period : 0.0625 µsec x (65535+1) = 4.096 msec

Interrupt periods according to value of WPRS are as fallows

Table 6.3. Underflow cycle of counter by operating frequency according to WPRS

WPRS	Prescaling	Interrupt period (PCLK=16MHz)	Interrupt period (PCLK=75MHz)
000	WDTCLKIN	4.096 msec	87.3 usec
001	WDTCLKIN/4	16.384 msec	3.4 msec
010	WDTCLKIN/8	32.768 msec	6.9 msec
011	WDTCLKIN/16	65.536 msec	13.9 msec
100	WDTCLKIN/32	131.072 msec	27.9 msec
101	WDTCLKIN/64	266.144 msec	55.9 msec
110	WDTCLKIN/128	524.288 msec	111.8 msec
111	WDTCLKIN/256	1.048 sec	223.6 msec

6.4 Operation Description

6.4.1 Control Downcounter of Watchdog Timer

If WDT(Watchdog Timer) is activated, WEN in WDT_CON register is set to ‘1’ and 32bits Downcounting with WDT_CVR register starts. In this case, WDT_CVR register must be set to prevent immediate reset or interrupt occurrence.
To chang the value of WDT_CVR, non-zero value should be written in WDT_LR when WEN is ‘1’, and the value of counter is reset if the value of counter is reloaded using WDT_LR register during operation of downcounter. However, the operating downcounter value must be greater than ‘0’.

6.4.2 Watchdog Timer Reset Mode

To reset WDT(Watchdog Timer), WRE of WDT_CON register and WDTRSTE(Bit 3) must be set to ‘1’. The reset by WDT(Watchdog Timer) to detect software errors occurs in transition state of downcounter from ‘1’ to ‘0’

6.4.3 Watchdog Timer Interrupt Mode

Watchdog Timer (WDT) interrupts can be used when certain safety tasks or data logging needs to be performed before the actual reset is generated.
Watchdog Timer interrupts is activated when WIE bit of WDT_CON register is set to ‘1’.
Interrupt ist generated when value of WDT_CVR register becomes ‘0’ (in trasition sate from ‘1’ to ‘0’), and WOF bit of WDT_CON register is set to ‘1’. To clear the WOF bit, non-zero value should be written in WDT_CVR with WDT_LR.

6.4.4 Using Watchdog Timer in Debug Mode

If the WDH bit of WDT CON register is set to ‘1’, watchdog timer counter in debug state can be stopped.
It can be checked that the downcounting of WDT(Watchdog Timer) is stopped when the debugging is paused after debugging run in debug mode, and it can be checked that the value of down counter decreases by 1 after debugging re-run.
Conversely, if the WDH bit is clear to '0', the Watchdog Timer counter continues to operate in the debug mode without stopping. When debugging is paused after debugging run, the value of the downcounter seems to be stopped, but it can be checked that the value is decreased after debugging re-run because of in operation of the downcounter.

6.4.5 Setting Example

<Example 1> WDT (Watchdog Timer) periodic interrupt mode - PCLK 74MHz, 1ms Period

```
WDT_CON<WDH> = “0”                                   : Initializing Control Register of Watchdog Timer
WDT_CON<WOF> = “0”
WDT_CON<WIE> = “0”
WDT_CON<WRE> = “0”
WDT_CON<WEN> = “0”
WDT_CON<WEC> = “0”
WDT_CON<WPRS[2:0]> = “000”
WDT_LR<WDTLR[31:0]>   =   “00000000   00000000        : Initializing Load Register of Watchdog Timer
00000000 00000000”

WDT_CON<WDH> = “1”                                   : Activation Debug of Watchdog Timer
WDT_CON<WIE> = “1”                                   : Activation Interrupt of Watchdog Timer
WDT_CON<WRE> = “1”                                   : Deactivation Reset of Watchdog Timer
WDT_CON<WEC> = “0”                                   : Set clock source as PCLK(74MHz)
WDT_CON<WPRS[2:0]>=”000”                             : Set Prescaler Value for Clock
WDT_LR<WDTLR[31:0]>                                  : when 1 tick = (1/74)us, set period = 1ms
= ”00000000_00000001_00100001_00010000”              → 1ms / (1/74)us = 74000

WDT_CON<WEN> = ”1”                                   : Run Watchdog Timer

NVICICER[0]<CLRENA[31:0]>                            : Clear Interrupt Register of NVIC Watchdog Timer
= “00000000_00000000_00000000_00001000”
NVICICPR[0]<CLRPEND[31:0]>                           : Clear Pending Bits of NVIC Watchdog Timer
= “00000000_00000000_00000000_00001000”
NVICIP[3]<PRI_3[31:24]> = “11100000”                 : Set Prioriy of NVIC Watchdog Timer
NVICISER[0]<SETPEND[31:0]>                           : Activation Interrupt of NVIC Watchdog Timer
= “00000000_00000000_00000000_00001000”
```

<Example 2> WDT (Watchdog timer) reset mode - RINGOSC 1MHz, 1s Period

```
PMU_RSSR<WDTRST> = “0”                            : Clear the status of watch-dog timer’s reset source
PMU_RSER<WDTRSTE> = “1”                           : Enable reset function of watch-dog timer
PMU_PCSR<WDTCS[1:0]> = “11”                       : Set the clock source of watch-dog timer (RingOSC, 1MHz)

WDT_CON<WDH> = “0”                                : Initializing control register of watch-dog timer
WDT_CON<WOF> = “0”                                : Initializing load register of watchdog timer
WDT_CON<WIE> = “0”
WDT_CON<WRE> = “0”
WDT_CON<WEN> = “0”
WDT_CON<WEC> = “0”
WDT_CON<WPRS[2:0]> = “0”
WDT_LR<WDTLR[31:0]> = “0”

WDT_CON<WDH> = “1”                                : Activation the flag to operate watch-dog timer in debug mode
WDT_CON<WIE> = “0”                                : Disable the interrupt of watch-dog timer
WDT_CON<WRE> = “1”                                : Activation the reset function of watch-dog timer
WDT_CON<WEC> = “1”                                : Select the clock source that is input to the WDT from PMU_PCSR.
WDT_CON<WPRS[2:0]> = “000”                        : Set a prescaler value of watch-dog timer clock

WDT_LR<WDTLR[31:0]>                               : 1tick : (1/1)us
=”00000000_00001111_01000010_010000000”           → period = 1s    1s / 1us = 1,000,000

WDT_CON<WEN> = “1”                                : Enable watch-dog timer operation
```

CHAPTER 7. 32-bit FRT (32-bit Free-run Timer)

Table 7.1. Operation Summary

Item	Pin Name	Remark
Clock usage	RINGOSC, IOSC16, MXOSC, SXOSC PCLK	Set by PMU_PCSR or FRT_CON Built-in prescaler
Reset Source	Set by PMU_PER	
Reset Generation	None	
Interrupt Generation	When FNT_CNT=FRT_PRD, (FRT(4))	Interval mode
Interrupt Clear Method	Writing ‘0’ to the FMF and FOF bit	FRT_CON

7.1 Overview

The A33G52x has a built-in 32-bit up-count timer for FRT (Free-run Timer). This FRT (Free-run Timer) can perform overflow interrupts or match interrupts according to the setting period of user application.

Features of FRT (Free-run Timer)

- 32-bit up-count timer
  - Periodic timer mode accoding to setting period
  - Free-run timer mode
- FRT (Free-run Timer) interrupts
  - Overflow interrupt
  - Match interrupt
- FRT (Free-run Timer) intput clocks
  - PCLK
  - Selected clock source by PMU_PCSR<FRTCS[3:2]> : Main XTAL, IOSC16, SXOSC, RINGOSC
- 8-steps FRT prescaler support for FRT (Free-run Timer) input clock

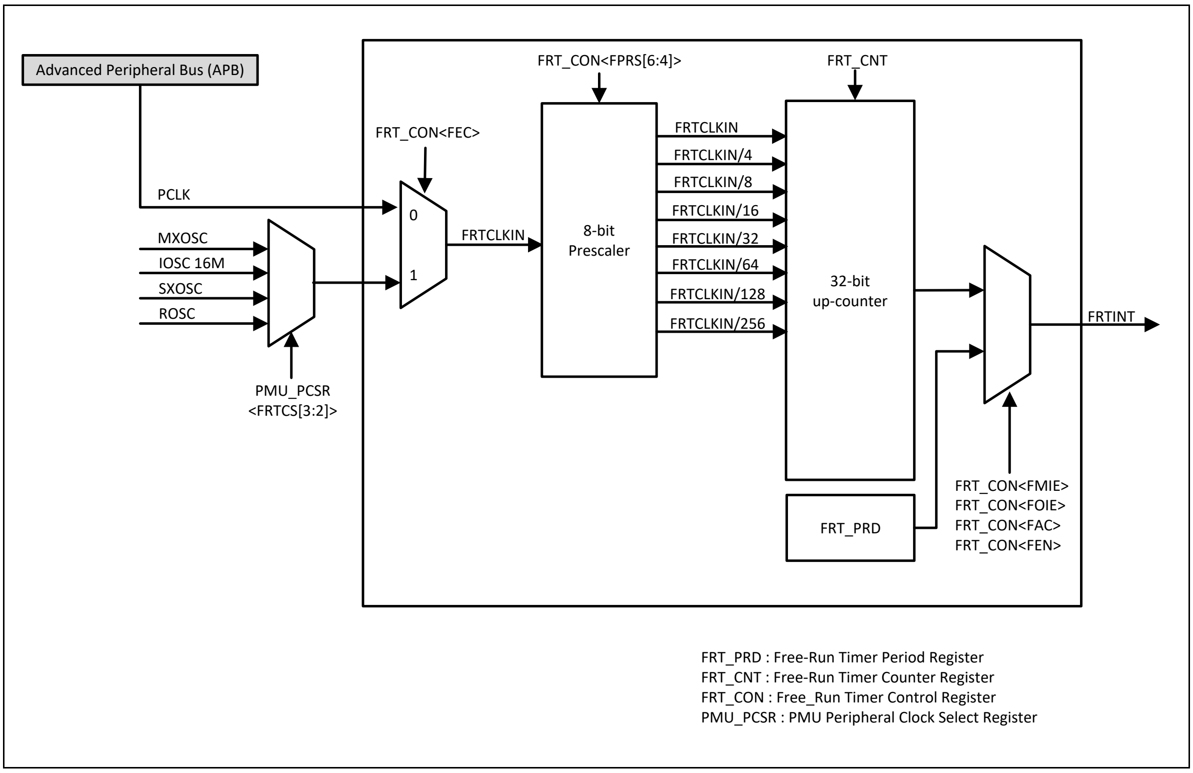


Figure 7.1. Block diagram of FRT (Free-run Timer)

7.2 Register Map

The base address of the FRT block is x4000_0500 and the register list is shown in the following table.

Table 7.2. The Register List of FRT (Free-run Timer)

Name	Offset	Access Type	Description	Initial value	Ref
FRT_PRD	0x00	RW	FRT (Free-run Timer) Period Register	0x00000000	7.3.1
FRT_CNT	0x04	RW	FRT (Free-run Timer) Counter Register	0x00000000	7.3.2
FRT_CON	0x08	RW	FRT (Free-run Timer) Control Register	0x00000000	7.3.3

7.3 Register Description

7.3.1 FRT_PRD FRT (Free-run Timer) Period Register

This register is for setting the period register when FRT(Free-run Timer) operates in periodic timer mode.
In the free-run mode of the A33G52x, when the value of the FRT_CNT register increases and becomes equal to the FRT_PRD value, a match interrupt occurs when the match interrupt bit is enabled. And match interrupt was occurred, FRTCNT will be ‘0’ and the FRT will restart. And match interrupt was occurred, FRT_CNT will be ‘0’ and the FRT will be restart.
The value calculated by subtracting 1 from the FRT period value must be entered to FRT_PRD.

FRT_PRD=0x4000_0500

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRTPRD																															
00000000 00000000 00000000 00000000																															
RW																															

31 **0**	**FRTPRD**	Set the period value of FRT (Free-Run Timer).

7.3.2 FRT_CNT FRT (Free-run Timer) Counter Register

FRT_CNT represents the current value of the timer as a free-running timer (FRT) and provides a 32-bit register that can access with write or read. This FRT_CNT register is an up-count timer in which the count value increases according to the FRT clock.
if FEN bit in the FRT_CON register is set to '1', FRT_CNT register can write or read access. When FEN = '0' the count value is not reflected in FRT_CNT but is reflected immediately in FRT_CNT when FEN = '1'.

FRT_CNT=0x4000_0504

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRTCNT																															
00000000 00000000 00000000 00000000																															
RW																															

31 **0**	**FRTCNT**	Current count value of FRT (Free-run Timer) When FRT_CON<FEN> = ‘0’, the counter value is not reflected. When FRT_CON<FEN>= ’1’, the counter value is reflected.

7.3.3 FRT_CON FRT (Free-run Timer) Control Register

This register controls overall operations of FRT (Free-run Timer). FRT_CON register offers match and overflow interrupt of free-run timer, and the function to set the operating mode and speed of free-run timer.
If match interrupt is used, FRTO_FMF
When using FRT match interrupts, you can set PF11 pin to FRTO_FMF by setting pin mux. this function enables match interrupts by activating FRT_CON <FMIE> and then executes FRT. Next, when the FRT is counted and a match interrupt event occurs, the output signal of FRTO_FMF pin outputs 'H' when FRT_CON <FMF> = ‘1’. On the contrary, if FRT_CON <FMF> = ‘0’ is written to clear the match interrupt event, the output signal of the FRTO_FMF pin becomes 'L'.

FRT_CON=0x4000_0508

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						**FMF**	**FOF**	**FEC**	**FPSR**			**FMIE**	**FOIE**	**FAC**	**FEN**
-																						0	0	0	000			0	0	0	0
-																						RW	RW	RW	RW			RW	RW	RW	RW

9	**FMF**	0	No match event
		1	Match event has occurred (write ‘0’ to clear)
8	**FOF**	0	No overflow event
		1	Overflow event has occurred (write ‘0’ to clear)
7	**FEC**	0	Select PCLK as a clock source
		1	Select a clock source with PMUPCSR
6	**FPRS**	000	FRTCLKIN
4		001	FRTCLKIN/4
		010	FRTCLKIN/8
		011	FRTCLKIN/16
		100	FRTCLKIN/32
		101	FRTCLKIN/64
		110	FRTCLKIN/128
		111	FRTCLKIN/256
3	**FMIE**	0	Disable Match Interrupt
		1	Enable Match Interrupt
2	**FOIE**	0	Disable Overflow Interrupt
		1	Enable Overflow Interrupt
1	**FAC**	0	Free-run mode
		1	Periodic mode
0	**FEN**	0	Stop FRT counter
		1	Enable FRT counter

Note) When the interrupt is used by setting the FRT (Free-run Timer) prescaler value or the input frequency to be slow, the FRT interrupt flag is cleared according to the interval of divided input clock.

7.4 Functional Description

The operation mode of A33G52x FRT (Free-run Timer) is determined by FRT_CON<FAC> bit value.
When FRT_CON<FAC> value is set to '0', FRT (Free-run Timer) runs in free-runs mode and when FRT_CON<FAC> value is set to '1', FRT (Free-run Timer) runs in periodic mode.
When FRT runs in free-run timer mode, FRT_CON<FOIE> bit determines whether overflow interrupt occurs. When FRT runs in periodic timer, FRT_CON<FMIE>bit determines whether match interrupt occurs.

7.4.1 Free-run Mode

When FRT (Free-run Timer) operates free-run mode, FRT_CNT value increases from 0 to 0xFFFF_FFFF regardless of FRT_PRD value. If FRT_CNT value reaches to 0xFFFF_FFFF, the next value is set to 0. In this case, FRT_CON<FOIE> bit was set to '1', an overflow interrupt occurs when FRT_CNT value changes from 0xFFFF_FFFF to 0.

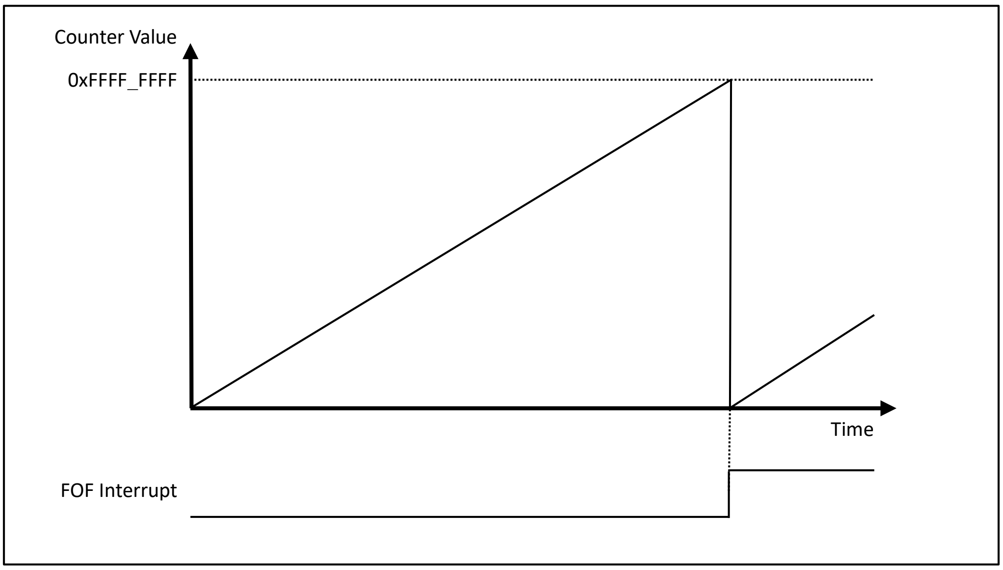


Figure 7.2. A Operating Timing in Free-run mode

7.4.2 Periodic timer mode

When the FRT (Free-Run Timer) is executed in the periodic timer mode, the FRT_CNT value is initialized to 0 and counted up again when the FRT_CNT value becomes equal to the FRT_PRD value. At this moment, If the FRT_CON<FMIE> bit was set to '1', a match interrupt occurs.

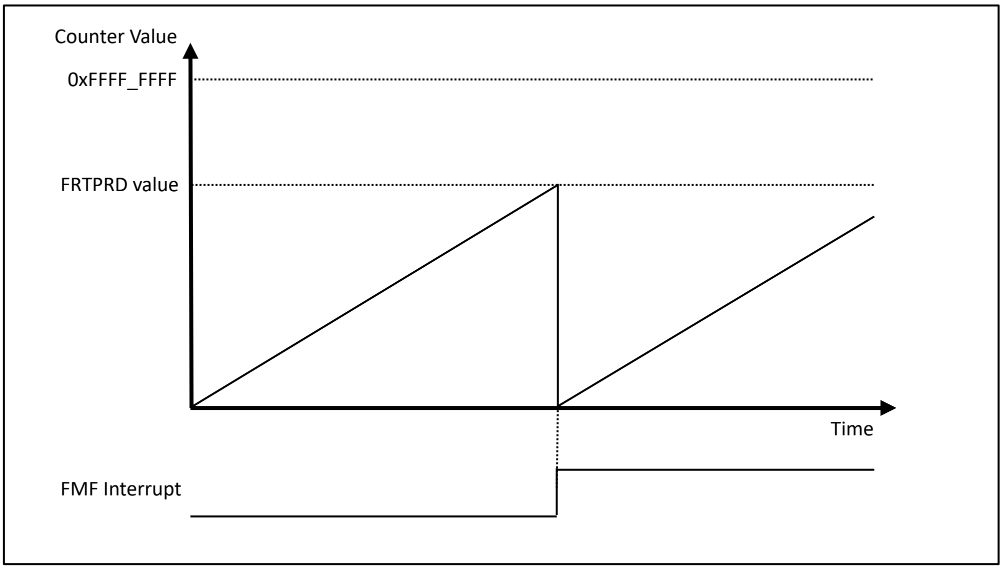


Figure 7.3. A Operating Example in Periodic mode-1

If the FRT_CNT value is set to a value larger than FRT_PRD, FRT_CNT initializes 0xFFFF_FFFF, and the next operation is the same as the periodic timer mode. In this case, Although FRT_CON<FOIE> bit was set to 1, match interrupt does not occur when FRT_CNT value changes from 0xFFFF_FFFF to 0. In this case, if the FRT_CNT value changes from 0xFFFF_FFFF to 0, even though the FRT_CON <FOIE> bit is set to 1, no match interrupt occurs.

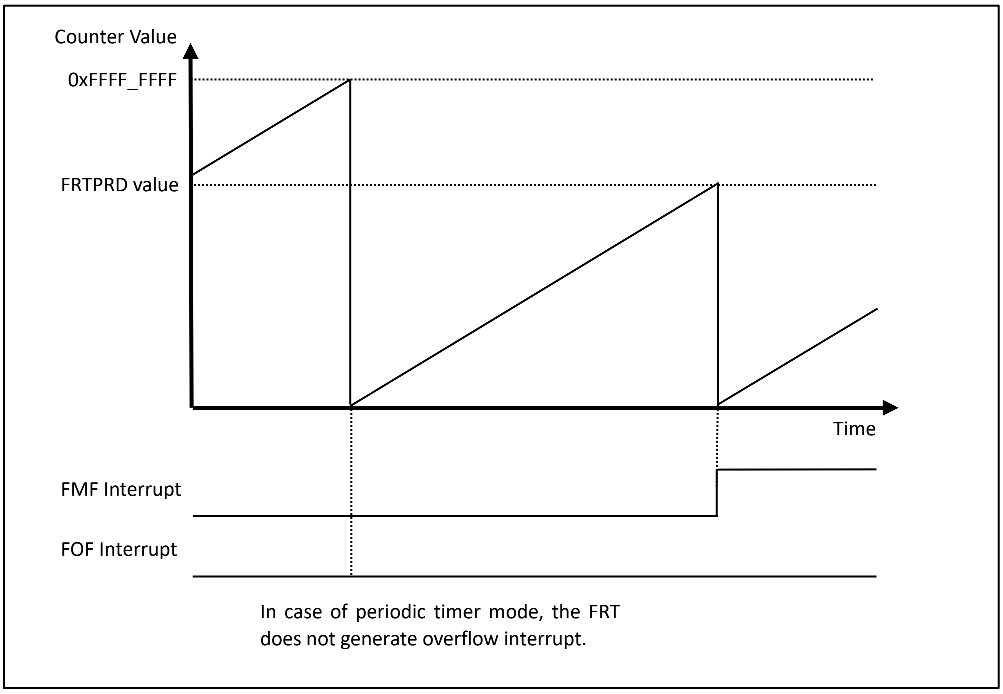

Counter Value
0xFFFF_FFFF
FRTPRD value
Time
FMF Interrupt
FOF Interrupt
In case of periodic timer mode, the FRT does not generate overflow interrupt.

Figure 7.4. A operating example in periodic mode-2

7.4.3 Timing Calculation of FRT (Free-run Timer)

FRT interval timer mode, the timer is assumed to calculate the timing. Clock source PCLK 16MHz, FRTPRD value is 0x4000 (16384), FPRS = 000 is assumed.

1-clock period : 1/16MHz = 0.0625 µsec
Interrupt interval : 0.0625 µsec x 16384 = 1.024 msec

The table below shows the interval of interrupt depending on the FPRS values.

Table 7.3. Interrupt intervals according to the FPRS value(PCLK=16MHz, FRTPRD=0x4000)

FRT_CON<FPRS[6:4]>	Clock Source	Interval of Interrupt
000	FRTCLKIN	1.024 msec
001	FRTCLKIN/4	4.096 msec
010	FRTCLKIN/8	8.192 msec
011	FRTCLKIN/16	16.384 msec
100	FRTCLKIN/32	32.768 msec
101	FRTCLKIN/64	65.536 msec
110	FRTCLKIN/128	131.072 msec
111	FRTCLKIN/256	262.144 msec

7.4.4 Setting Example

<Example 1> Setting the free-run mode of FRT (8MHz XTAL, Period = 1s)

```
PMU_PCSR<FRTCS[3:2]> = “00”                          : Selects FRT Clock source as an 8MHz external crystal.
FRT_CON<FEC> = “1”                                   : Uses FRT clock source set by PMU_PCSR (FRTCLKIN)
FRT_CON<FPRS[6:4]> = “010”                           : Uses FRT Clock(FRTCLKIN) is divided 8.
FRT_CON<FAC> = “0”                                   : Sest FRT operation mode
FRT_CNT<FRTCNT[31:0]>                                : Initializes FRT Counter Value to zero
=”00000000_00000000_00000000_00000000”
FRT_PRD<FRTPDR[31:0]>                                : Set FRT operating clock (FRTCLKIN/8 = 1MHz)
=”00000000_00001111_01000010_00111111”                 1 period : 1s -> 1s / 1us = 1,000,000
FRT_CON<FMIE> = “1”                                  : Enables FRT match interrupt event
FRT_CON<FOIE> = “1”                                  : Enables FRT overflow interrupt event

NVICICER[0]<CLRENA[31:0]>                            : Clears NVIC FRT interrupt bit
= “00000000_00000000_00000000_00010000”
NVICICPR[0]<CLRPEND[31:0]>                           : Clears NVIC FRT pending bit
= “00000000_00000000_00000000_00010000”
NVICIP[4]<PRI_4[7:0]> = “11100000”                   : Sest priority of NVIC FRT
NVICISER[0]<SETPEND[31:0]>                           : Enables interrupt of NVIC FRT
= “00000000_00000000_00000000_00010000”

FRT_CON<FEN> = “1”                                   : Starts the operation of FRT counter
```

<Example 2> setting the periodic timer mode of FRT (MXOSC = 8MHz, 1s)

```
PF_MR<PF11[23:22]> = “10”                            : Set PF11 pin MUX to FRT match flag output.
PF_CR<P11[23:22]> = “00”                             : Set PF11 pin direction to push-pull output
PF_PCR<P11> = “0”                                    : disable the pull-up/pull-down resistor of PF11 pin

PMU_PCSR<FRTCS[3:2]> = “00                           : Use 8MHz MXOSC as FRT clock source
FRT_CON<FEC> = “1”                                   :  Use the clock defined by PMU_PCSR as the clock source of FRT.
FRT_CON<FPRS[6:4]> = “010”                           : Set prescaler value of FRT input clock
FRT_CON<FAC> = “1”                                   : Use the FRT as Periodic timer mode

FRT_PRD<FRTPDR[31:0]>                                : 1 tick : (1/1)us
=”00000000_00001111_01000010_00111111”                 period : 1s -> 1s / 1us = 1,000,000

FRT_CON<FMIE> = “1”                                  : Enable FRT match interrupt event

NVICICER[0]<CLRENA[31:0]>                            : Set NVIC FRT interrupt
= “00000000_00000000_00000000_00010000”                - IRQ number : Free-run Timer IRQ(4)
NVICICPR[0]<CLRPEND[31:0]>                             - Priority: 1
= “00000000_00000000_00000000_00010000”                - Lower priority: 3
NVICIP[4]<PRI_4[7:0]> = “11100000”                   - : Enable interrupt of NVIC FRT
NVICISER[0]<SETPEND[31:0]>
= “00000000_00000000_00000000_00010000”

FRT_CON<FEN> = “1”                                   : Starts the operation of FRT counter
FRT_CNT<FRTCNT[31:0]>                                : Initializes FRT Counter Value to zero
=”00000000_00000000_00000000_00000000”
```

CHAPTER 8. 16-bit TIMER

Table 8.1. Operation Summary

Item	Pin Name	Remark
Clock usage	RINGOSC, IOSC16, MXOSC, SXOSC PCLK, TnC	Set by PMU_PCSR Set by TIMERn_CON Built-in prescaler
Reset Source	Set by PMU_PER	
Reset Generation	None	
Interrupt Generation	Timer mode: Match/Clear/Overflow Capture mode : Capture Input (TIMER0(5) ~ TIMER9(14))	Set by TIMERn_CON
Interrupt Clear Method	Writing ‘1’ to the interrupt bit	TIMERn_CON

8.1 Overview

The A3352x MCU has a 16-bit timer with 10 channels. The clock source of 16-bit timer can be inputted by peripheral system clock (PCLK) or by external clock source. And it has a built-in 10-bit prescaler that can generate various timer clocks. . This 16-bit timer supports four types of operating modes: periodic timer mode, PWM mode, one shot mode, and capture mode. If the 16-bit timer operates in the periodic mode, a certain periodic interrupt set by the user may occur. When the 16-bit timer operates in PWM mode, the PWM signal can be output by adjusting the period and duty. In one-shot mode and PWM mode, one PWM signal is output. One shot and PWM mode is a mode that can output one PWM waveform. Capture mode can able to measure the time of signal accoding to condition of external input signalWhen using the capture mode, the time interval of the signal can be measured according to the set condition of the external input signal. You can also export output to the outside to control other devices.

Main features of 16-bit TIMER

- 16-bit upcount timer
- 4 types of operating modes
  - Periodic timer mode
  - One-shot timer mode
  - PWM mode
  - Capture mode
- Timer Interrupts
  - Match interrupt [1:0] / Overflow interrupt
- Timer Input clocks
  - 4-steps (1/2, 1/4, 1/8, 1/16) Prescaler of PCLK
  - Clock source selected by PMU_PCSR : Main XTAL, IOSC16, SXOSC, RINGOSC
  - Timer clock source through the input of TnC port
- 10-bit prescaler for timer input clock division

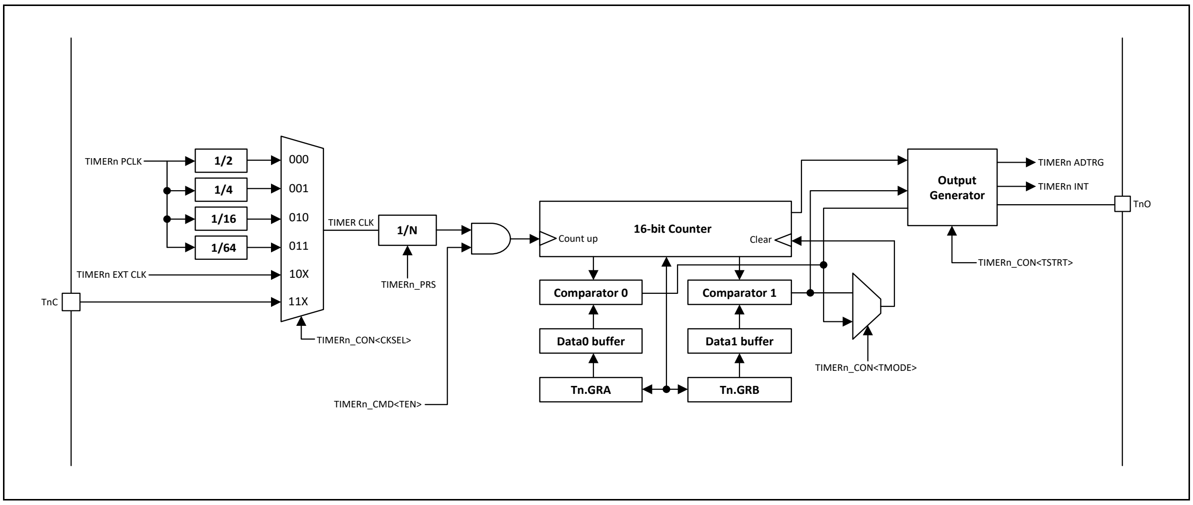


Figure 8.1. Block diagram of 16-bit Timer

8.2 Pin Configuration

Table 8.2. The external pin of 16-bit timer

Pin Name	Type	Description	Support by package		
			A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM	A33G526MM A33G526ML A33G524MM A33G524ML	A33G527VQ A33G527VL A33G526VQ A33G526VQ
T0C	I	Timer0 capture input signal / external clock input	O	O	O
T1C	I	Timer1 capture input signal / external clock input	O	O	O
T2C	I	Timer2 capture input signal / external clock input	O	O	O
T3C	I	Timer3 capture input signal / external clock input	O	O	O
T4C	I	Timer4 capture input signal / external clock input	O	O	O
T5C	I	Timer5 capture input signal / external clock input	O	O	O
T6C	I	Timer6 capture input signal / external clock input	O	O	O
T7C	I	Timer7 capture input signal / external clock input	-	-	O
T8C	I	Timer8 capture input signal / external clock input	O	O	O
T9C	I	Timer9 capture input signal / external clock input	O	O	O
T0O	O	Timer0 Timer/PWM/One-shot Output	O	O	O
T1O	O	Timer1 Timer/PWM/One-shot Output	O	O	O
T2O	O	Timer2 Timer/PWM/One-shot Output	O	O	O
T3O	O	Timer3 Timer/PWM/One-shot Output	O	O	O
T4O	O	Timer4 Timer/PWM/One-shot Output	O	O	O
T5O	O	Timer5 Timer/PWM/One-shot Output	O	O	O
T6O	O	Timer6 Timer/PWM/One-shot Output	O	O	O
T7O	O	Timer7 Timer/PWM/One-shot Output	O	O	O
T8O	O	Timer8 Timer/PWM/One-shot Output	-	O	O
T9O	O	Timer9 Timer/PWM/One-shot Output	-	O	O

8.3 Register Map

The A3352x MCU has a 16-bit timer with 10 channels. The base address of 16-bit timer is 0x4000_0C00 and the register list is shown in the following table.

Table 8.3. The Addresses for each channel

Channel	Address
TIMER0	0x4000_0C00
TIMER1	0x4000_0C20
TIMER2	0x4000_0C40
TIMER3	0x4000_0C60
TIMER4	0x4000_0C80
TIMER5	0x4000_0CA0
TIMER6	0x4000_0CC0
TIMER7	0x4000_0CE0
TIMER8	0x4000_0D00
TIMER9	0x4000_0D20

Each register of 16-bit timer is as follows.

Table 8.4. Register list of 16-bit timer

Register Name	Offset	Access Type	Description	Initial Value	Ref
TIMERn_CON	0x00	RW	Timer Control Register	0x00000000	8.4.1
TIMERn_CMD	0x04	RW	Timer Command Register	0x00000000	8.4.2
TIMERn_GRA	0x08	RW	Timer General Purpose Register A	0x00000000	8.4.3
TIMERn_GRB	0x0C	RW	Timer General Purpose Register B	0x00000000	8.4.4
TIMERn_PRS	0x10	RW	Timer Prescaler Register	0x00000000	8.4.5
TIMERn_CNT	0x14	RW	Timer Counter Register	0x00000000	8.4.6

8.4 Register Description

8.4.1 TIMERn_CON Timer n Control Register

It consists of a 32-bit register that controls all timer functions such as operating mode, timer clock source, and interrupts. The timer's operation mode and interrupts can be set according to the purpose of the user application.

TIMER0_CON=0x4000_0C00,TIMER1_CON=0x4000_0C20, TIMER2_CON=0x4000_0C40,

TIMER3_CON=0x4000_0C60, TIMER4_CON=0x4000_0C80, TIMER5_CON=0x4000_0CA0,

TIMER6_CON=0x4000_0CC0, TIMER7_CON=0x4000_0CE0, TIMER8_CON=0x4000_0D00,

TIMER9_CON=0x4000_0D20

31	30	29	28	27	26	25	24	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																	IOVF	TIF1	TIF0	Reserved	TOVE	TIE1	TIE0	TSTRT	TCS			CAPM	Reserved	TMODE	
-																	0	0	0	-	0	0	0	0	000			0	-	0	
-																	RC	RC	RC	-	RW	RW	RW	RW	RW			RW	-	RW	

14	**TOVF**	The flag of timer overflow interrupt	
		0	No overflow occurred
		1	Over flow occurred (Write ‘1’ to clear this flag)
13	**TIF1**	The flag of match 1 interrupt	
		0	No match1 interrupt occurred
		1	Match 1 interrupt occurred (Write ‘1’ to cleare this flag)
12	**TIF0**	The flag of match 0 interrupt	
		0	No match 0 interrupt occurred
		1	Match 0 interrupt occurred (Write ‘1’ to cleare this flag)
10	**TOVE**	Enable / disable overflow interrupt	
		0	Disable overflow interrupt
		1	Enable overflow interrupt
9	**TIE1**	Enable / Disable match 1 interrupt	
		0	Disable match 1 interrupt
		1	Enable match 1 interrupt
8	**TIE0**	Enable / Disable match 0 interrupt	
		0	Disable match 0 interrupt
		1	Enable match 0 interrupt
7	**TSTRT**	Setting Initial output in Periodic/PWM/One-shot mode	
		0	Initial output ‘L’ upon counter being cleared
		1	Initial output ‘H’ upon counter being cleared
6 **4**	**TCS**	Select the clock source of timer	
		000	TCLK/2
		001	TCLK/4
		010	TCLK/16
		011	TCLK/64
		10x	Select the clock source by PMU_PCSR
		11x	Select external TnC input as timer clock source
3	**CAPM**	Select clear timing of internal counter	
		0	Rising-edge clear
		1	Falling-edge clear
1 **0**	**TMODE**	Timer operation mode	
		00	Periodic mode
		01	PWM mode
		10	One-shot mode
		11	Capture mode

8.4.2 TIMERn_CMD Timer n Command Register

This register controls timer operation such as timer running or stop. The timer counter can also be initialized by writing '1' to the TIMERn_CMD<TCLR> bit.

TIMER0_CMD=0x4000_0C04, TIMER1_CMD=0x4000_0C24, TIMER2_CMD=0x4000_0C44
TIMER3_CMD=0x4000_0C64, TIMER4_CMD=0x4000_0C84, TIMER5_CMD=0x4000_0CA4
TIMER6_CMD=0x4000_0CC4, TIMER7_CMD=0x4000_0CE4, TIMER8_CMD=0x4000_0D04
TIMER9_CMD=0x4000_0D24

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

15	TOUT	The status of external output pin	
1	TCLR	0	No effect
		1	Initialize Timer. - Clear counter register - Set output TOUT as TSTRT value - Set internal buffer with TnGRA/B - Initialize capture signal
0	TEN	Select the operation of timer	
		0	Stop timer
		1	Run timer

8.4.3 TIMERn_GRA Timer n General Purpose Register A

This register is a General purpose register A of a timer. The operations for each mode are explained as below.

TIMER0_GRA=0x4000_0C08, TIMER1_GRA=0x4000_0C28, TIMER2_GRA=0x4000_0C48

TIMER3_GRA=0x4000_0C68, TIMER4_GRA=0x4000_0C88, TIMER5_GRA=0x4000_0CA8

TIMER6_GRA=0x4000_0CC8, TIMER7_GRA=0x4000_0CE8, TIMER8_GRA=0x4000_0D08

TIMER9_GRA=0x4000_0D28

31	30	29	28	27	26	25	24	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																GRA															
-																00000000 00000000															
-																RW															

15 0	GRA	**[Timer n General Purpose Register A]** This register works as below depend on the mode: **[Periodic mode / PWM mode / one-shot mode]** This register is used as target count comparator. When the counter value is equal to this register, the counter will be cleared and the timer will restart in periodic mode and PWM mode, or will stop in one-shot mode. When the timer restarts from '0', the output of TOUT becomes TSTRT. If the counter reaches this register value, then TIF0 interrupt will occur. On the TIF0 interrupt or the clear-operation. This register value is copied onto internal data buffer 0. $$Period = \frac{TCLKIN}{TnGRA}$$ **[Capture mode:]** In the rising edge clear mode, capture occurs upon falling edge In the falling edge clear mode, capture occurs upon rising edge

8.4.4 TIMERn_GRB Timer n General Purpose Register B

This register is a General purpose register B of a timer. The operations for each mode are explained as below.

TIMER0_GRB=0x4000_0C0C, TIMER1_GRB=0x4000_0C2C, TIMER2_GRB=0x4000_0C4C

TIMER3_GRB=0x4000_0C6C, TIMER4_GRB=0x4000_0C8C, TIMER5_GRB=0x4000_0CAC

TIMER6_GRB=0x4000_0CCC, TIMER7_GRB=0x4000_0CEC, TIMER8_GRB=0x4000_0D0C

TIMER9_GRB=0x4000_0D2C

31–16	15–0
Reserved	GRB
-	00000000 00000000
-	RW

15 0	GRB	**[Timer n General Purpose Register B]** This register works as blow depend on the mode: **[Periodic mode]** Not used **[PWM mode, one-shot mode]** This register is used as target count value. When the counter value is equal to this register, then TIF1 interrupt will occur. On the TIF1 interrupt or the clear-operation. This register value is copied onto internal data buffer 1. **[Capture mode]** In the rising edge clear mode, capture occurs upon rising edge (inverse condition of GRA) In the falling edge clear mode, capture occurs upon falling edge (inverse condition of GRA)

8.4.5 TIMERn_PRS Timer n Prescaler Register

This register can set the prescaler value that divides the timer n input frequency. It consists of a 10-bit register. By applying a prescaler to the timer n clock source selected in the TIMERn_CON register, you can generate more accurate and different timer base clocks.

TIMER0_PRS=0x4000_0C10, TIMER1_PRS=0x4000_0C30, TIMER2_PRS=0x4000_0C50

TIMER3_PRS=0x4000_0C70, TIMER4_PRS=0x4000_0C90, TIMER5_PRS=0x4000_0CB0

TIMER6_PRS=0x4000_0CD0, TIMER7_PRS=0x4000_0CF0, TIMER8_PRS=0x4000_0D10

TIMER9_PRS=0x4000_0D30

31	30	29	28	27	26	25	24	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																						PRS									
-																						0 00000000									
-																						RW									

9 0	PRS	Timer n prescaler value of input clock $TCLK = \frac{PCLK}{(PRS + 1)}$

8.4.6 TIMERn_CNT Timer n Counter Register

This is timer n counter register that is read or write accessible. It counts up to the specified period as the timer input speed set by the user.

TIMER0_CNT=0x4000_0C14, TIMER1_CNT=0x4000_0C34, TIMER2_CNT=0x4000_0C54

TIMER3_CNT=0x4000_0C74, TIMER4_CNT=0x4000_0C94, TIMER5_CNT=0x4000_0CB4

TIMER6_CNT=0x4000_0CD4, TIMER7_CNT=0x4000_0CF4, TIMER8_CNT=0x4000_0D14

TIMER9_CNT=0x4000_0D34

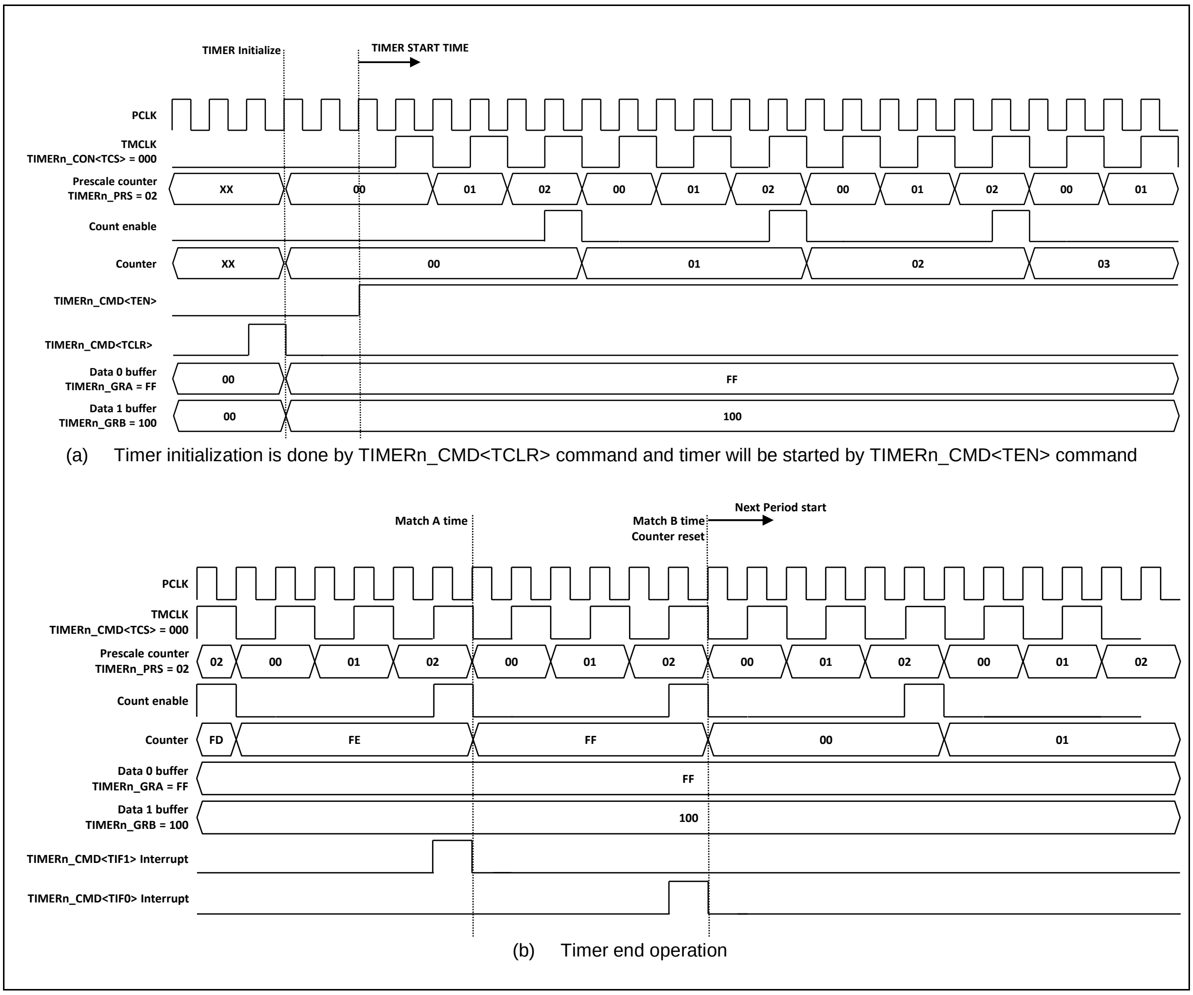
<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">CNT</td></tr>
<tr><td colspan="16">-</td><td colspan="16">00000000 00000000</td></tr>
<tr><td colspan="16">-</td><td colspan="16">RW</td></tr>
</table>

15 **0**	**CNT**	Current count value

8.5 Functional Description

8.5.1 The default behavior of 16-bit timer

TMCLK in Figure 8.2 is the reference clock for timer operation. This clock can be run counting clock by dividing prescalervalue of timer This clock can be operated by counting the value set in the prescaler register. The figure below shows the start and end points of the counter in the timer's periodic mode.



(a) Timer initialization is done by TIMERn_CMD<TCLR> command and timer will be started by TIMERn_CMD<TEN> command

(b) Timer end operation

Figure 8.2 Timer base start and match actions

The timer count period can be obtained by the following formula:
Period = TMCLK Period * TIMERn_GRA Value
Match A Interrupt Timing = TMCLK Period * TIMERn_GRA Value

It is recommended to write the value first in the TCLR of TIMERn_CMD before changing the TEN bit value of TIMERn_CMD when changing timer setting or restarting with new setting value.

8.5.2 Periodic mode of 16-bit timer

Figure8.3 is shown the timing of 16-bit timer’s periodic mode. The TIMERn_GRA value defines the period of the timer, but the TIMERn_GRB value does not affect the period of the timer.

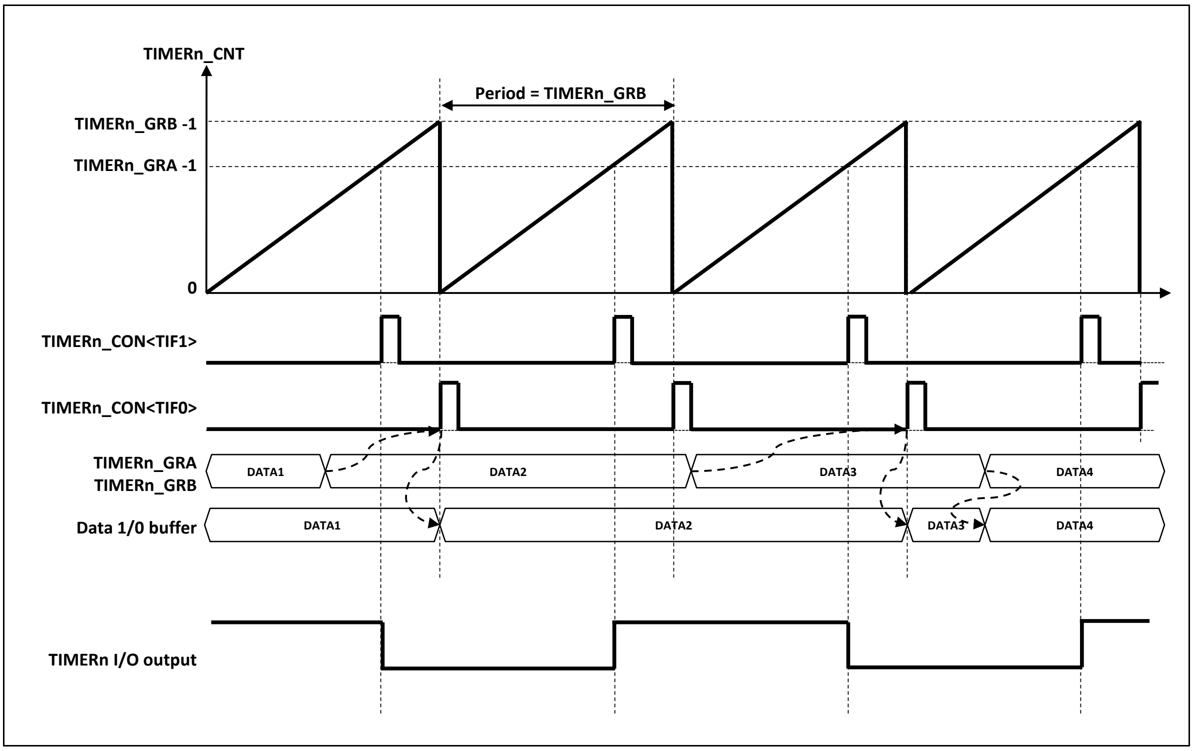


Figure 8.3 The periodic mode operation of 16-bit timer

The timer count period in periodic mode can be obtained by the following formula:

Period = TMCLK Period * TIMERn_.GRA Value
Match A Interrupt Timing = TMCLK Period * TIMERn_GRA Value

If TIMERn_GRA value is '0', it means that the timer period is '0'. In this case, the TIMERn_CMD <TEN> bit is set to '1' to execute the timer, but the timer does not work.
When a load condition occurs, the TIMERn_GRA and TIMERn_GRB values are loaded into the internal comparison data 0 buffer. In periodic mode, the TCLR behavior of the TIMERn_CMD register is loaded into the data buffer and the next GRA match event is loaded into the data buffer.

8.5.3 One-shot mode of 16-bit timer

Figure8.4 is shown the timing of 16-bit timer's one-shot mode.
The TIMER_nGRB value determines the one-shot period, and the other comparison point is provided by TIMERn_GRA.

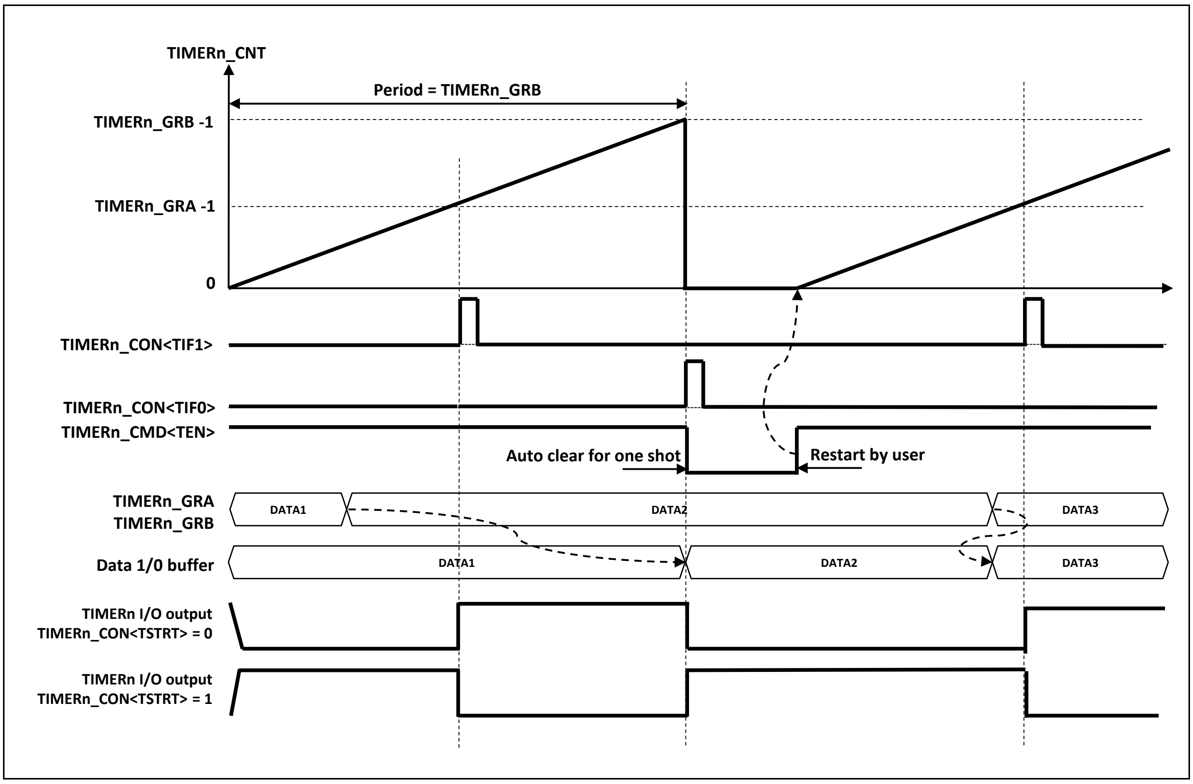


Figure 8.4 The one-shot mode operation of 16-bit timer

The timer count period in one-shot mode can be obtained by the following formula:

Period = TMCLK Period * TIMERn_GRB Value
Match A Interrupt Timing = TMCLK Period * TIMERn_GRA Value

If TIMERn_GRB value is '0', it means that the timer period is '0'. In this case, the TIMERn_CMD <TEN> bit is set to '1' to execute the timer, but the one-shot timer does not work.
When a load condition occurs, the TIMERn_GRA and TIMERn_GRB values are loaded into the internal compare data 0 buffer and data 1 buffer. In one-shot mode, the TCLR write behavior of the TIMERn_CMD register is loaded into the data buffer and the next GRB match event is loaded into the data buffer.
The signal output from the TnO pin is the same as in PWM mode. The TIMERn_GRB value defines the period of the output pulse and the TIMERn_GRA value defines the duty of the one-shot pulse.

8.5.4 PWM mode of 16-bit timer

Figure8.5 is shown the timing of 16-bit timer's PWM mode. The TIMERn_GRB value determines the PWM pulse period and the TIMERn_GRA value, which is a comparison point, determines the PWM pulse duty.

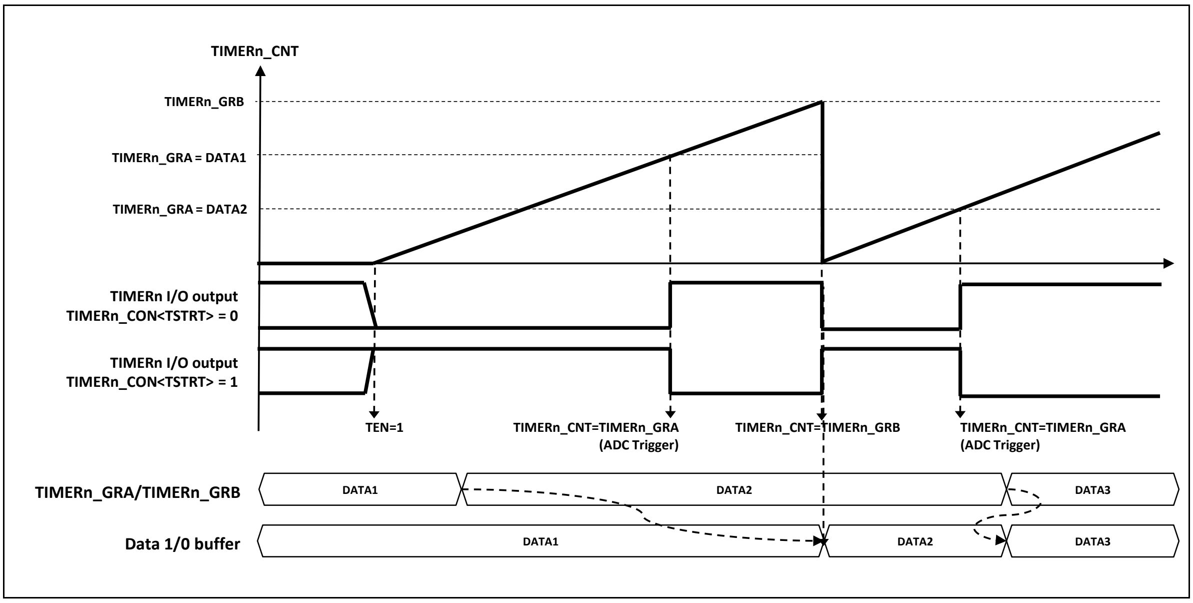

TIMERn_CNT
TIMERn_GRB
TIMERn_GRA = DATA1
TIMERn_GRA = DATA2
TIMERn I/O output TIMERn_CON<TSTRT> = 0
TIMERn I/O output TIMERn_CON<TSTRT> = 1
TEN=1
TIMERn_CNT=TIMERn_GRA (ADC Trigger)
TIMERn_CNT=TIMERn_GRB
TIMERn_CNT=TIMERn_GRA (ADC Trigger)
TIMERn_GRA/TIMERn_GRB
DATA1
DATA2
DATA3
Data 1/0 buffer
DATA1
DATA2
DATA3

Figure 8.5 The PWM mode operation of 16-bit timer

The timer count period in PWM mode can be obtained by the following formula:

Period = TMCLK Period * TIMERn_GRB Value
Match A Interrupt Timing = TMCLK Period * TIMERn_GRA Value

If TIMERn_GRB value is '0', it means that the timer period is '0'. In this case, the TIMERn_CMD <TEN> bit is set to '1' to execute the timer, but the PWM timer does not work.
When a load condition occurs, the TIMERn_GRA and TIMERn_GRB values are loaded into the internal compare data 0 buffer and data 1 buffer. In one-shot mode, the TCLR write behavior of the TIMERn_CMD register is loaded into the data buffer and the next GRB match event is loaded into the data buffer.
The signal output from the TnO pin is PWM. The TIMERn_GRB value defines the period of the output pulse and the TIMERn_GRA value defines the duty of the one-shot pulse. The level of the voltage to be output when this PWM starts operation can be controlled by the value of the TSTRT bit in the TIMERn_CON register.
The ADC trigger generator is enabled with Match interrupt time.

8.5.5 Capture mode of 16-bit timer

Figure8.6 is shown the timing of 16-bit timer's capture mode.
TnC is used for pulse acquisition of the input signal. Counter values can be captured according to the capture conditions such as rising or falling edges.

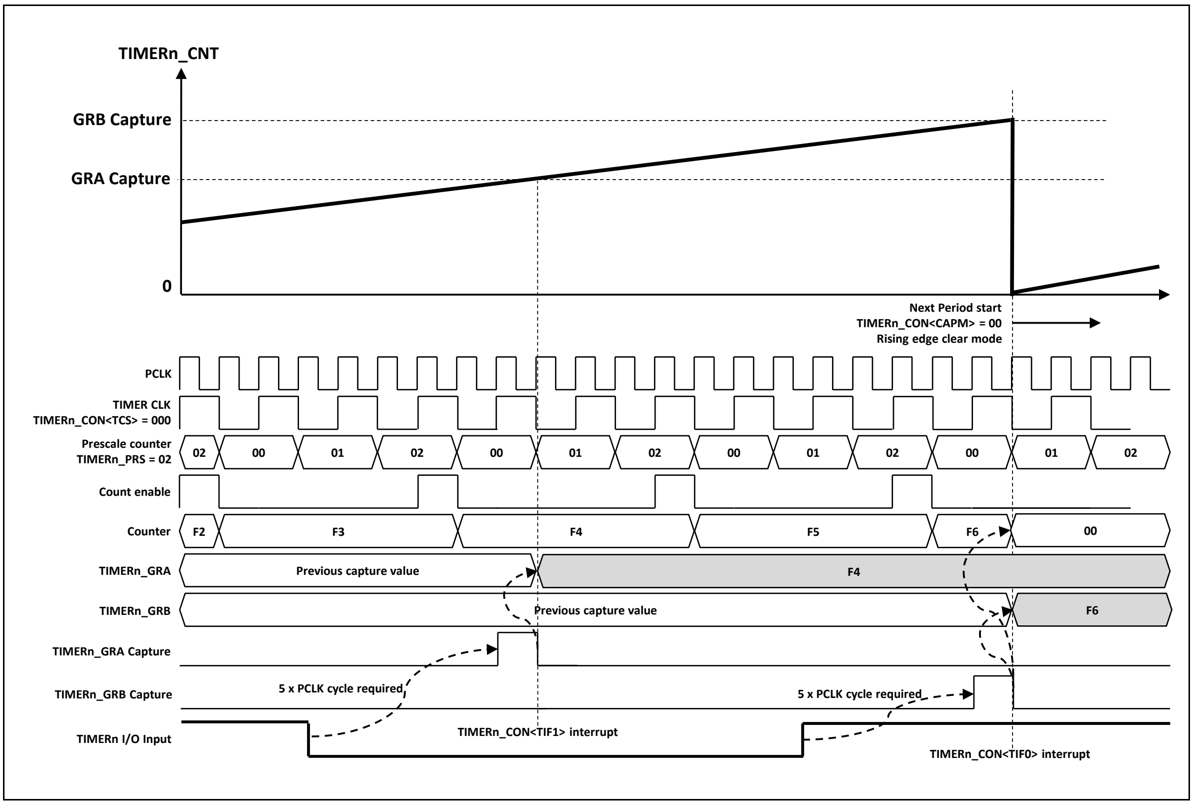


Figure 8.6 The capture mode operation of 16-bit timer

Timer capture internally requires a minimum of 5 PCLK clock cycles, and the actual capture point is the rising or falling edge of the TnC input signal after 5 clock cycles. The internal counter value can be cleared by the rising edge or falling edge with the CAPM bit in the TIMERn_CON register.

8.5.6 ADC Trigger mode of 16-bit timer

A33G52x's 16-bit timer can generate the ADC start trigger signal and can be the trigger source for a ADC block. The ADC trigger source control of timer can be set in ADC control register (ADC_CON)

Figure8.6 is shown the timing of 16-bit timer's ADC trigger mode. To use a timer as the ADC trigger source, the ADC conversion rate must be less than the timer period. Otherwise, an overrun situation will occur. The ADC ACK is not needed because the trigger signal is automatically cleared after 3 PCLK clock pulses.

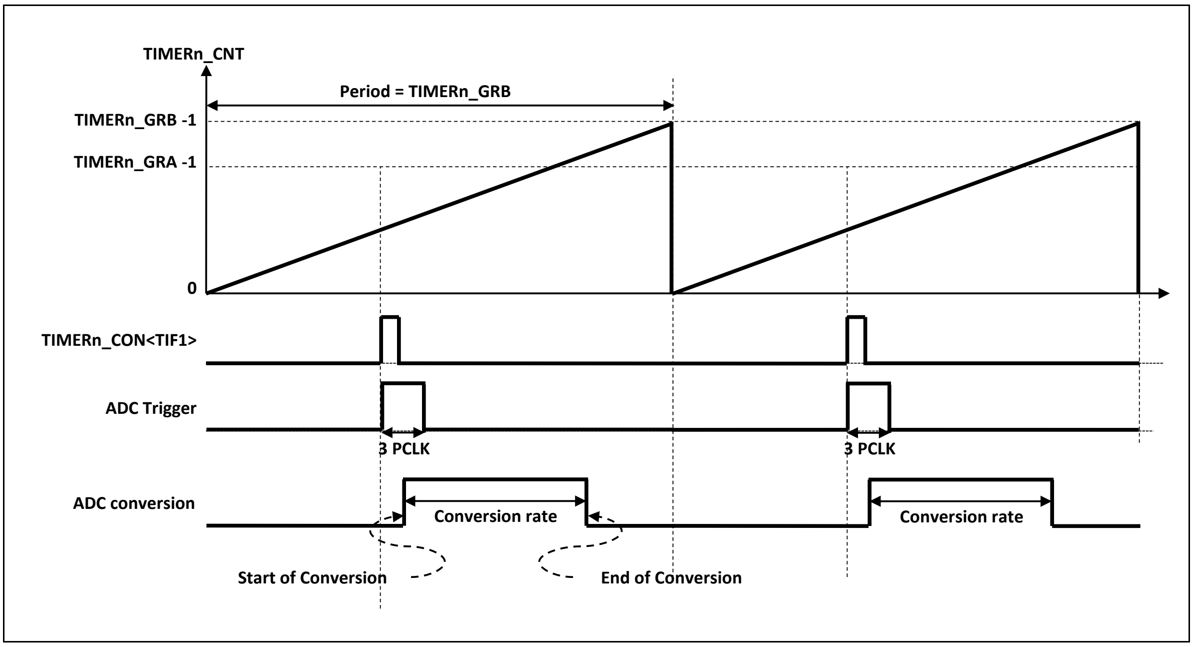


Figure 8.7 Block Diagram for ADC Trigger Timing of 16-bit timer

8.5.7 Setting Example

<Example 1> Setting the periodic mode of timer (PCLK = 16MHz, Period = 1ms)

```
TIMERn_CMD<TCLR> = “1”                                : Initializes the timer by writing ‘1’ to clear counter value
TIMERn_CMD<TCLR> = “0”
TIMERn_CON<TMODE[1:0]> = “00”                         : Sets the periodic mode of timer.
TIMERn_CON<TCS[6:4]> = “011”                          : Selects the timer clock source.(16MHz/64) = 4us
TIMERn_PRS<PRS[9:0]> = “0x00”                         : Sets the prescaler value for dividing timer input frequency.

TIMERn_GRA<GRA[15:0]> = “00000000_11111010”           : Sets the value ‘250’ of GRA (250kHz/ 250) =1ms
TIMERn_GRB<GRB[15:0]> = “00000000_00000000”           : Initializes the value of GRB.
TIMERn_CNT<CNT[15:0]> = “00000000_00000000”           : Initializes the counter value of timer

TIMERn_CON<TOVE> = “0”                                : Disables the overflow interrupt flag of timer
TIMERn_CON<TIE1> = “0”                                : Disables the flag of match 1 interrupt of timer
TIMERn_CON<TIE0> = “0”                                : Disables the flag of match 0 interrupt of timer

TIMERn_CON<TOVF> = “1”                                : Writes ‘1’ to clear overflow interrupt flag of timer
TIMERn_CON<TIF1> = “1”                                : Writes ‘1’ to clear match 1 interrupt flag of timer
TIMERn_CON<TIF0> = “1”                                : Writes ‘1’ to clear match 0 interrupt flag of timer
TIMERn_CON<TIE0> = “1”                                : Enables match 0 interrupt of periodic timer mode

TIMERn_CMD<TEN> = “1”                                 : Starts the operation of periodic timer
```

CHAPTER 9. PWM Generator

Table 9.1. Operation Summary

Item	Pin Name	Remark
Clock usage	PCLK	Built-in prescaler
Reset Source	Set by PMUPER	
Reset Generation	When PWM counter is cleared to ‘0’	Set by PWMn_CTRL
Interrupt Generation	When PWM counter is cleared to ‘0’ (PWM0(24) ~ PMW7(31))	Set by PWMn_CTRL
Interrupt Clear Method	When writing non-‘0’ value ot PRF	PWMn_CTRL

9.1 Overview

The A33G52x has eight built-in PWM (Pulse-Width Modulation) generators that can output PWM (Pulse-Width Modulation) waveforms. This PWM (Pulse-Width Modulation) generator has a 16-bit resolution, four channels form a 1-unit, and each unit has a built-in prescaler. 이 PWM (Pulse-Width Modulation) It can be used as square waveform signal required for user application such as LED, motor, inverter, etc. through PWM related register setting.

Main features of PWM (Pulse-Width Modulation)

- PWM (Pulse-Width Modulation) output with 16-bit resolution
- PWM output 8 channels
- PWM invert signal output channels
  - 100-pin (PWM0~PWM7)
  - 80pin (PWM1~PWM7)
- PWM duty setting function by setting counter and period register.
- PWM input clocks
  - PCLK
  - 4 steps (1/2, 1/4, 1/8, 1/16) prescaler
- Integrated 1-unit 8-bit prescaler for PWM input clock division
- In multichannel PWM operation, timing synchronization based on the PWM signal with the shortest cycle

9.2 Block Diagram

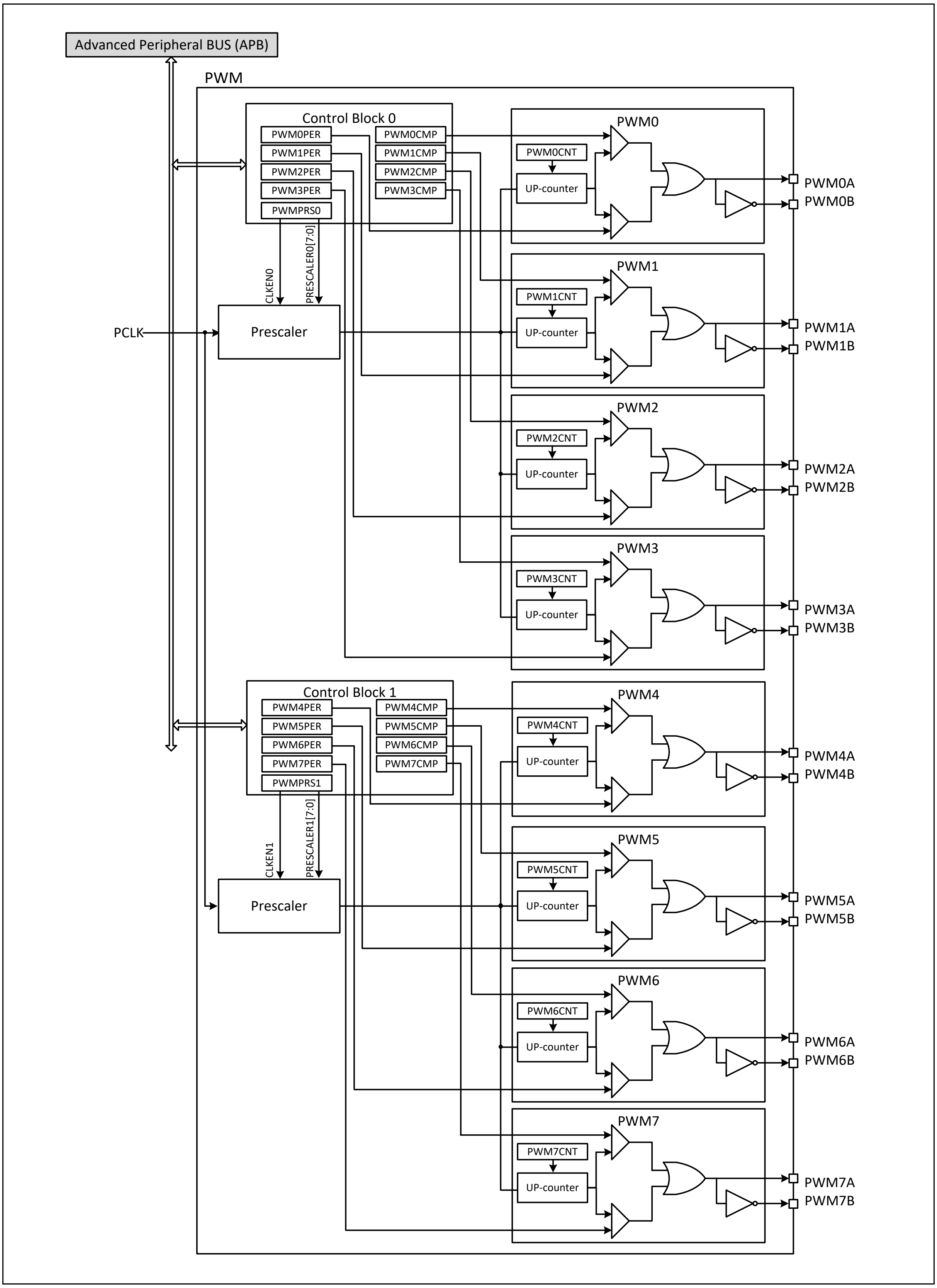


Figure 9.1. The block diagram of PWM

9.3 Pin configuration

PWM (Pulse-Width Modulation) pin configuration of A33G52x series is as follows. The 80-pin and 100-pin products are each provided with an inverted output signal mode for PWM0-7.

Table 9.2. The external pins of PWM

Pin Name	Type	Description	Support by package		
			A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM	A33G526MM A33G526ML A33G524MM A33G524ML	A33G527VQ A33G527VL A33G526VQ A33G526VQ
PWM0A	O	Output of PWM0	O	O	O
PWM0B	O	Reverse output of PWM0	-	-	O
PWM1A	O	Output of PWM1	O	O	O
PWM1B	O	Reverse output of PWM1	-	O	O
PWM2A	O	Output of PWM2	O	O	O
PWM2B	O	Reverse output of PWM2	-	O	O
PWM3A	O	Output of PWM3	O	O	O
PWM3B	O	Reverse output of PWM3	-	O	O
PWM4A	O	Output of PWM4	O	O	O
PWM4B	O	Reverse output of PWM4	-	O	O
PWM5A	O	Output of PWM5	O	O	O
PWM5B	O	Reverse output of PWM5	-	O	O
PWM6A	O	Output of PWM6	-	O	O
PWM6B	O	Reverse output of PWM6	O	O	O
PWM7A	O	Output of PWM7	-	O	O
PWM7B	O	Reverse output of PWM7	O	O	O

9.4 Register Map

The base address of the PWM block is 0x4000_0700 and the address for each channel is as follows :

Table 9.3. The base address for each PWM (Pulse-Width Modulation) channel

Channel	Base Address
PWM0	0x4000_0700
PWM1	0x4000_0720
PWM2	0x4000_0740
PWM3	0x4000_0760
PWM4	0x4000_0780
PWM5	0x4000_07A0
PWM6	0x4000_07C0
PWM7	0x4000_07E0
PWMPRS0	0x4000_077C
PWMPRS1	0x4000_07FC

The following table shows register list of each channel.

Table 9.4. The register list of PWM (Pulse-Width Modulation)

Register Name	Offset	Access Type	Description	Initial Value	Ref
PWM_PRS0	0x077C	RW	PWM0-3 Prescaler Register 0	0x00000000	9.5.1
PWM_PRS1	0x07FC	RW	PWM4-7 Prescaler Register 1	0x00000000	9.5.2
PWMn_CTRL	0x00	RW WC	PWM Timer n Control Register	0x00000000	9.5.3
PWMn_CNT	0x04	RW	PWM Timer n Counter Register	0x00000000	9.5.4
PWMn_PER	0x08	RW	PWM Timer n Period Register	0x00000000	9.5.5
PWMn_CMP	0x0C	RW	PWM Timer n Compare Register	0x00000000	9.5.6

9.5 Register Description

9.5.1 PWM_PRS0 PWM0-3 Prescaler Register 0

This register is used to set Unit 0 of the PWM (Pulse-Width Modulation) generator. This register provides a prescaler for dividing the PWM input clock of unit 0 and the input clock set in this register is applied to the PWM0 to PWM3 channels.

PWM_PRS0=0x4000_077C

31	30	29	28	27	26	25	24	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																CLKEN0	Reserved							PRESCALER0							
-																0	-							00000000							
-																RW	-							RW							

15	CLKEN0	Operation control of prescaler 0 clock	
		0	Stop prescaler 0 clock
		1	Run prescaler 0 clock
7 0	PRESCALER0	The prescaler value of unit 0 $$\mathrm{PWMCLK0} = \frac{PCLK}{PRESCALER0 + 1}$$	

9.5.2 PWM_PRS1 PWM4-7 Prescaler Register 1

This register is used to set Unit 1 of the PWM (Pulse-Width Modulation) generator. This register provides a prescaler for dividing the PWM input clock of unit 1 and the input clock set in this register is applied to the PWM4 to PWM7 channels.

PWM_PRS1=0x4000_07FC

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td>CLKEN1</td><td colspan="7">Reserved</td><td colspan="8">PRESCALER1</td></tr>
<tr><td colspan="16">-</td><td>0</td><td colspan="7">-</td><td colspan="8">00000000</td></tr>
<tr><td colspan="16">-</td><td>RW</td><td colspan="7">-</td><td colspan="8">RW</td></tr>
</table>

15	CLKEN1	Operation control of prescaler 1 clock	
		0	Stop prescaler 1 clock
		1	Run prescaler 1 clock
7 0	PRESCALER1	The prescaler value of unit 1 $$\mathrm{PWMCLK1} = \frac{PCLK}{PRESCALER1 + 1}$$	

9.5.3 PWMn_CTRL PWM timer n Control Register

PWM (Pulse-Width Modulation) This register is used to control channel n.
This register has a function to specify PWM clock and control settings for each PWM output channel.

PWM0_CTRL=0x4000_0700, PWM1_CTRL=0x4000_0720

PWM2_CTRL=0x4000_0740, PWM3_CTRL=0x4000_0760

PWM4_CTRL=0x4000_0780, PWM5_CTRL=0x4000_07A0

PWM6_CTRL=0x4000_07C0, PWM7_CTRL=0x4000_07E0

31	30	29	28	27	26	25	24	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																			PRF	Reserved			PRIE	CKSEL			Reserved	INVA	Reserved	SYNC	STRT
-																			0	-			0	000			-	0	-	0	0
-																			WC	-			RW	RW			-	RW	-	RW	RW

12	PRF	Display the PWM periodic interrupt flag	
		0	No periodic interrupt
		1	Periodic interrupt occurs (write '1' to clear)
8	PRIE	Enable/Disable PWM periodic interrupt	
		0	Disable PWM periodic interrupt
		1	Enable PWM periodic interrupt
7 5	CKSEL	Select a input clock source of PWM	
		000	divided by 2
		001	divided by 4
		010	divided by 8
		011	divided by 16
		1xx	Disable input clock
3	INVA	Selection for PWM Inversion Output	
		0	Normal PWM output A (Not inverted)
		1	Inverted PWM output A
1	SYNC	Synchronize the current PWM channel to another channel's timing.	
		0	Do not synchronize this channel with the others
		1	Synchronize this channel to the other channels (Synchronized to the timing of the PWM channel with the smallest PRD value among the several PWM output channels.) (Refer to 9.6.4)
0	STRT	Control the operation of PWM	
		0	Stop counter
		1	Run counter

9.5.4 PWMn_CNT PWM Timer n Counter Register

This register is the count value register of the PWM channel n. This register consists of 16 bits.

PWM0_CNT=0x4000_0704, PWM1_CNT=0x4000_0724

PWM2_CNT=0x4000_0744, PWM3_CNT=0x4000_0764

PWM4_CNT=0x4000_0784, PWM5_CNT=0x4000_07A4

PWM6_CNT=0x4000_07C4, PWM7_CNT=0x4000_07E4

31	30	29	28	27	26	25	24	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															
-																00000000 00000000															
-																RW															

15 **0**	**CNT**	Current PWM counter value

9.5.5 PWMn_PER PWM Timer n Counter Period Register

This register specifies the period of PWM channel n counter and consists of 16 bits. The value of the counter reaches the value specified in this register, the counter value is reset.

PWM0_PER=0x4000_0708, PWM1_PER=0x4000_0728

PWM2_PER=0x4000_0748, PWM3_PER=0x4000_0768

PWM4_PER=0x4000_0788, PWM5_PER=0x4000_07A8

PWM6_PER=0x4000_07C8, PWM7_PER=0x4000_07E8

31	30	29	28	27	26	25	24	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																PERIOD															
-																00000000 00000000															
-																RW															

15 0	PERIOD	The period value of PWM channel n $\text{PWM Period} = \frac{PWMCLK}{PERIOD+1}$

9.5.6 PWMn_CMP PWM Timer n Compare Register

This register specifies the value of PWM (Pulse-Width Modulation) comparator A for output A of channel n and consists of 16 bits. When the counter is reset, the output A is output as 'L'. If the value of this register matches the value of the PWM (Pulse-Width Modulation) counter, the output A is set to 'H'.

PWM0_CMP=0x4000_070C, PWM1_CMP=0x4000_072C

PWM2_CMP=0x4000_074C, PWM3_CMP=0x4000_076C

PWM4_CMP=0x4000_078C, PWM5_CMP=0x4000_07AC

PWM6_CMP=0x4000_07CC, PWM7_CMP=0x4000_07EC

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="16">CMP</td></tr>
<tr><td colspan="16">-</td><td colspan="16">00000000 00000000</td></tr>
<tr><td colspan="16">-</td><td colspan="16">RW</td></tr>
</table>

15 0	CMP	The comparator value of PWM channel n

9.6 Functional Description

9.6.1 Setting the input clock of PWM (Pulse-Width Modulation) Generator

The source clock for PWM (Pulse-Width Modulation) is determined by the system peripheral clock and the prescaler in the PWM_PRSn register or the prescaler in the PWMn_CTRL register. PWM (Pulse-Width Modulation) source clock can be obtained by the formula below.

$$\mathbf{PWMCLKn} = \frac{\boldsymbol{PCLK}}{\boldsymbol{PRESCALER}\mathbf{n} + \mathbf{1}}$$

The first way to get the PWM source clock is to obtain the PWM input clock PWMCLKn by activating PWMCLKn with the PWMPRSn [15] bits in the PWM_PRSn register and setting the PRESCALERn [7: 0] value to the system peripheral input clock. PWMCLK0 is the clock applied to PWM0-3 group, and PWMCLK1 is the clock applied to PWM4-7 group.
PWMCLKn applied to each PWMn can be set to the input clock of each PWM (Pulse-Width Modulation) channel by dividing 2, 4, 8, or 16 by the value set in CKSEL [7: 5] through the clock divider.

9.6.2 Setting the period and PWM output signal

Each PWMn channel has a PWMn_PER register for setting the PWM frequency.
PWM period can be obtained by below fomula with 16-bit width value.
The PWM (Pulse-Width Modulation) period can be obtained by applying the following formula with 16-bit PWMn_PER value and the PWM input clock.

$$\mathbf{PWM\ Period} = \frac{\boldsymbol{PWMCLK}}{\boldsymbol{PERIOD+1}}$$

Each PWMn channel can set PWM duty as well as period. PWM duty can be set to a value in the range of 0 ~ PERIOD in PWMn_CMP[15:0].
The default output of PWMnA is 'L', and the PWMnA output changes to 'H' when the PWMn_CNT register value reaches the PWMn_CMP value.
You can also invert the starting output level of PWMnA by entering a value in the INVA bit in the PWMn_CTRL register. If you enter 0 as the default value, the start output value will be Low. If you input 1, High will be output. PWMnB is output when the PWMnA waveform is inverted.

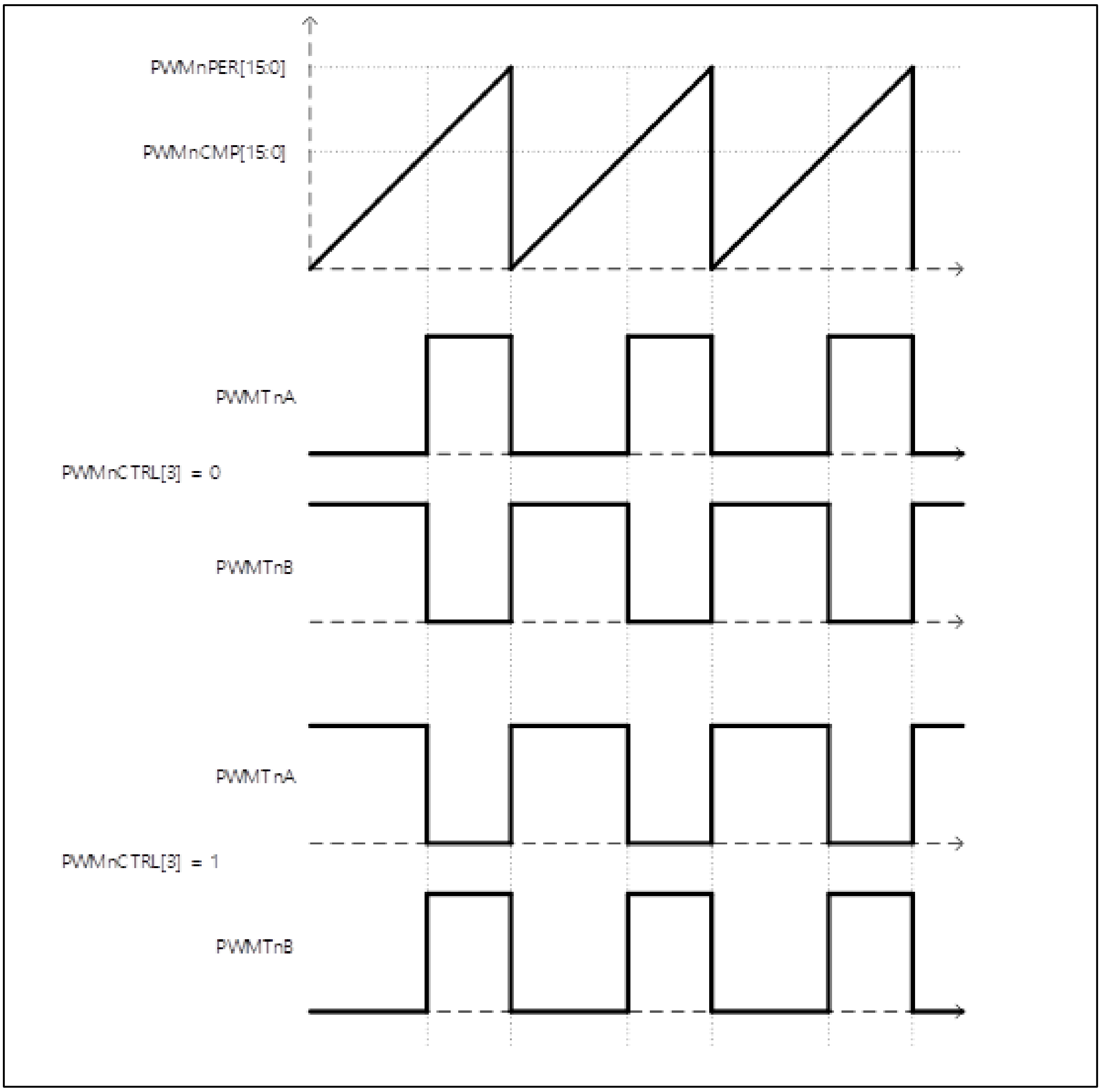


Figure 9.2. The block diagram of PWM output timing

After each of the above functions are set, if 1 is input to the STRT bit in the PWMn_CTRL register, PWM is output. If 0 is input, PWM output is stopped.

9.6.3 PWM (Pulse-Width Modulation) Interrupts

Each of PWMn channels can be configured periodic interrupt. If the PRIE bit in the PWMn_CTRL register is set to ‘1’, the PWM periodic interrupt is enabled. If the PRIE bit in the PWMn_CTRL register is set to ‘0’, the periodic interrupt is disabled. Interrupt vector is supported for each channel. If an interrupt occurs in each PWMn channel, it is possible to check whether an interrupt is generated by reading the PRF bit in the PWMn_CTRL register. If a periodic interrupt occurs, the PRF bit must be cleared by writing 1 to it.

9.6.4 Synchronization of PWM channels

The PWM (Pulse-Width Modulation) block has a function to set synchronization between channels. if the synchronization function is enabled through the control registers of each channel in the PWM0-3 group synchronizes the counting of the active PWM channels. The timing of the synchronized PWM channels is set based on the peirod of the shortest PWM channel. Similarly, PWM4-7 groups operate in the same way as PWM0-3 groups. However, the synchronization function of PWM0-3 group and PWM4-7 group is independent and does not affect the operation between PWM0-3 and PWM4 ~ 7 groups.

Note) In a PWM group (PWM0-3 or PWM4-7), when the enabled synchronous (SYNC = 1) channel and asynchronous (SYNC = 0) channel are used at the same time, the period of synchronous PWM channels is effected by asynchronous PWM channel. Therefore, if the application requires independent PWM operation, it is recommended to use PWM channel of other PWM group or to use TIMER with PWM mode.

9.6.5 Setting Example

<Example 1> Setting PWM0, PWM4 output in polling mode (PCLK = 16MHz, 1ms period, 50% duty)

PWMPRS0<PRESCALER0[7:0]>=”1111”	: Sets the prescaler value of PWM0-3 (16MHz / (15+1)) = 1MHz
PWMPRS0<CLKEN> = “1”	: Enables the output of PWM0-3 prescaler
PWMPRS1<PRESCALER1[7:0]> = “1111”	: Sets the prescaler value of PWM4-7 (16MHz / (15+1)) = 1MHz
PWMPRS1<CLKEN> = “1”	: Enables the output of PWM4-7 prescaler
PWM0CTRL<CKSEL[7:5]> = “100”	: Disables the clock divider of PWM0 channel
PWM4CTRL<CKSEL[7:5]> = “100”	: Disables the clock divider of PWM4 channel
PWM0PER<PERIOD[15:0]> = “00000011 11101000”	: Sets the value of PMW0 period to 1000 (1us*1000) = 1ms
PWM0CMP<CMP[15:0]> = “00000001 11110100”	: Sets the value of PWM0 duty count to 500.
PWM4PER<PERIOD[15:0]> = “00000011 11101000”	: Sets the value of PMW4 period to 1000 (1us*1000) = 1ms
PWM4CMP<CMP[15:0]> = “00000001 11110100”	: Sets the value of PWM4 duty count to 500.
PWM0CTRL<INVA> = “0”	: Sets the start output level of PWM0 channel to Low.
PWM0CTRL<STRT> = “1”	: Starts PWM0 operation.
PWM4CTRL<INVA> = “0”	: Sets the start output level of PWM4 channel to Low.
PWM4CTRL<STRT> = “1”	: Starts PWM4 operation.

<Example 2> Setting PWM0, PWM4 output in interrupt mode (PCLK = 16MHz, 1ms period, 50% duty)

PWMPRS0<PRESCALER0[7:0]> = “1111”	: Sets the prescaler value of PWM0-3 (16MHz / (15+1)) = 1MHz
PWMPRS0<CLKEN> = “1”	: Enables the output of PWM0-3 prescaler.
PWMPRS1<PRESCALER1[7:0]> = “1111”	: Sets the prescaler value of PWM4-7 (16MHz / (15+1)) = 1MHz
PWMPRS1<CLKEN> = “1”	: Enables the output of PWM4-7 prescaler.
PWM0CTRL<CKSEL[7:5]> = “100”	: Disables the clock divider of PWM0 channel
PWM4CTRL<CKSEL[7:5]> = “100”	: Disables the clock divider of PWM4 channel
PWM0PER<PERIOD[15:0]> = “00000011_11101000”	: Sets the value of PMW0 period to 1000 (1us*1000) = 1ms
PWM0CMP<CMP[15:0]> = “00000001_11110100”	: Sets the value of PWM0 duty count to 500.
PWM4PER<PERIOD[15:0]> = “00000011_11101000”	: Sets the value of PMW4 period to 1000 (1us*1000) = 1ms
PWM4CMP<CMP[15:0]> = “00000001_11110100”	: Sets the value of PWM4 duty count to 500.
PWM0CTRL<PRIE>=’1’	: Enables the periodic interrupt of PWM0 channel.
PWM4CTRL<PRIE>=’1’	: Enables the periodic interrupt of PWM4 channel
PWM0CTRL<INVA> = ‘0’	: Sets the start output level of PWM0 channel to Low.
PWM0CTRL<STRT> = ‘1’	: Starts PWM0 operation.
PWM4CTRL<INVA> = ‘0’	: Sets the start output level of PWM4 channel to Low.
PWM4CTRL<STRT> = ‘1’	: Starts PWM4 operation.

CHAPTER 10. I^{2}C Interface

Table 10.1. Operation Summary

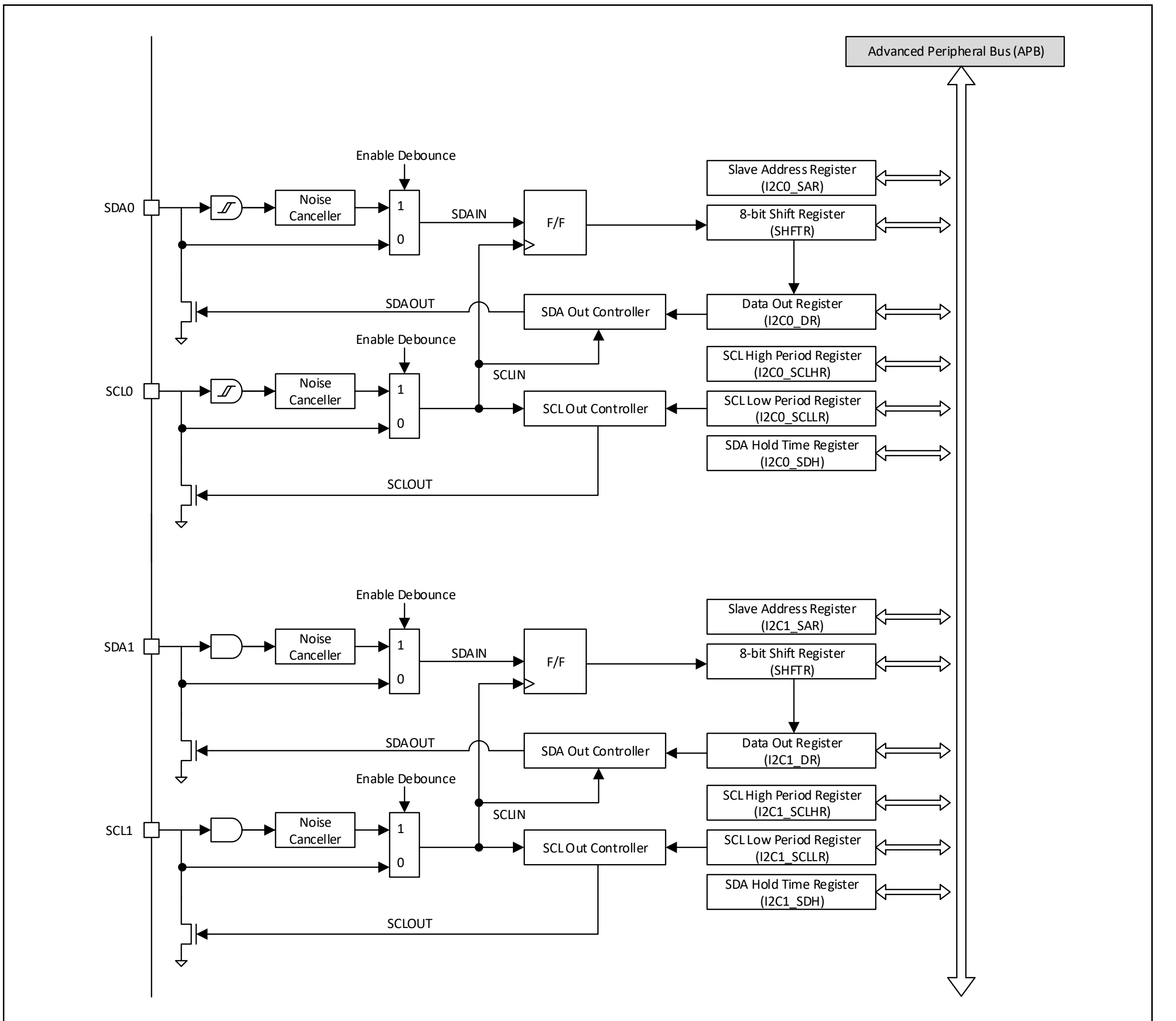
Item	Pin Name	Remark
Clock usage	PCLK	Transper rate
Reset Source	Set by PMU_PER Software reset function	I2Cn_CR
Reset Generation	None	
Interrupt Generation	GCALL, End of transmit, End of receive ACK/STOP reception, SLAVE selected (I2C0(36), I2C1(37))	I2Cn_SR
Interrupt Clear Method	Write 0xFF to the I2Cn_SR	I2Cn_SR

10.1 Overview

The I2C interface on the A33G52x meets the standard I2C (Inter Integrated Circuit) communication interface and is used for serial communication with internal and external devices of the same serial bus format. It supports both master and slave mode with 2 channels and can send or receive data per byte-based interrupt or polling method. I2C (Inter Integrated Circuit) blocks are used to communicate with various peripherals with the same bus format. A33G52x To use the I2C (Inter Integrated Circuit) function, it is recommended to set SCL and SDA as open drain and then connect the external pull-up resistor and use it as 'HIGH' state.

Main features of I2C (Inter Integrated Circuit) Interrface

- Standard I2C (Inter Integrated Circuit) interface communication
  - 2 channels support
- Master and slave operation mode setting
- Single master – multi-slave operation support
  - 1: 1 and N: N (up to 1008) slave device support
- The setting function of i2c transmission rate
  - Up to 400kHz communication speed setting support
- I2C (Inter Integrated Circuit) interrupt
- 7-bit slave addressing
- Delay setting for SCL High or Low waveforms
- time setting function for SDA previous value data



Advanced Peripheral Bus (APB)
Enable Debounce
SDA0
Noise Canceller
1
0
SDA IN
F/F
Slave Address Register (I2C0_SAR)
8-bit Shift Register (SHFTR)
SDA OUT
SDA Out Controller
Data Out Register (I2C0_DR)
Enable Debounce
SCL IN
SCL0
Noise Canceller
SCL Out Controller
SCL High Period Register (I2C0_SCLHR)
SCL Low Period Register (I2C0_SCLLR)
SDA Hold Time Register (I2C0_SDH)
SCLOUT
Enable Debounce
SDA1
Noise Canceller
SDA IN
F/F
Slave Address Register (I2C1_SAR)
8-bit Shift Register (SHFTR)
SDA OUT
SDA Out Controller
Data Out Register (I2C1_DR)
Enable Debounce
SCL IN
SCL1
Noise Canceller
SCL Out Controller
SCL High Period Register (I2C1_SCLHR)
SCL Low Period Register (I2C1_SCLLR)
SDA Hold Time Register (I2C1_SDH)
SCLOUT

NOTES)
1. SDA0, SCL0 of I2C0 channel is Schmitt-trigger input type
2. SDA1, SCL0 of I2C1 channel is CMOS input type

Figure 10.1. The block diagram of I2C Interface

10.2 Pin Configuration

The I2C (Inter Integrated Circuit) has two signals: SCL and SDA. For communication, these signals must be set to the internal or external pull-up state of the open-drain. However, if only internal pull-up is connected without external pull-up, the communication speed may be limited due to high resistance value.

Table 10.2. The List of I2C Pins

Pin Name	Type	Description	Support by package		
			A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM	A33G526MM A33G526ML A33G524MM A33G524ML	A33G527VQ A33G527VL A33G526VQ A33G526VQ
SCL0	I/O	Serial Clock Signal (open-drain)	O	O	O
SDA0	I/O	Serial Data Signal (open-drain)	O	O	O
SCL1	I/O	Serial Clock Signal (open-drain)	O	O	O
SDA1	I/O	Serial Data Signal (open-drain)	O	O	O

Notes)

1. SDA0, SCL0 Schmitt-trigger input type
2. SDA1, SCLK1 CMOS input Type

10.3 Register Map

The base address of the I^{2}C (Inter Integrated Circuit) interface module is 0x4000 0A00 and the each channel address is shown in the table below.

Table 10.3. The List of the Channel Address

Channel	Address
I2C0	0x40000A00
I2C1	0x40000A80

Table 10.4. I^{2}C The Register List of I2C Interface

Register Name	Offset	Access Type	Description	Initial Value	Ref
I2Cn_DR	0x00	RW	I2C n Tx/Rx Data Register	0x0000000FF	10.4.1
I2Cn_SR	0x08	RW	I2C n Status Register	0x00000000	10.4.2
I2Cn_SAR	0x0C	RW	I2C n Slave Mode Address	0x00000000	10.4.3
I2Cn_CR	0x14	RO RW WO	I2C n Configuration Register	0x00000000	10.4.4
I2Cn_SCLL	0x18	RW	I2C n SCL Low Period Register	0x0000FFFF	10.4.5
I2Cn_SCLH	0x1C	RW	I2C n SCL High Period Register	0x0000FFFF	10.4.6
I2Cn_SDH	0x20	RW	I2C n SDA Hold Time Register	0x00007FFF	10.4.7

10.4 Register Description

10.4.1 I2Cn_DR I^{2}C n Tx/Rx Data Register

I2Cn_DR is an 8-bit register that has read / write access and stores data to be transferred and received data.

I2C0_DR=0x4000_0A00, I2C1_DR=0x4000_0A80

31	30	29	28	27	26	25	24	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																								ICDR							
-																								11111111							
-																								RW							

7 0	ICDR	**[Transmission mode]** The data stored in this register is output via the serial data line (SDA) and transmitted. **[Receive mode]** The value entered through SDA is saved and the received data can be read by reading this register.

10.4.2 I2Cn_SR I^{2}C n Status Register

I2Cn_SR is an 8-bit register that stores the status of various events occurring during I2C serial communication. By reading the value stored in this register, each flag can be checked for i2c communication status.

I2C0_SR=0x4000_0A08, I2C1_SR=0x4000_0A88

31	30	29	28	27	26	25	24	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																								GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMODE	RXACK
-																								0	0	0	0	0	0	0	0
-																								RW	RW	RW	RW	RW	RW	RW	RW

7	**GCALL**	General call detection flag	
		0	No general call detected
		1	General call detected or acquired mastership
6	**TEND**	1 byte transmission completion flag	
		0	Transmission NOT completed
		1	1 byte transmission end
5	**STOP**	Detection the end of transmission	
		0	Not detected stop condition
		1	Detected Stop condition
4	**SSEL**	Slave selection status flag	
		0	NOT selected as slave
		1	Selected as slave
3	**MLOST**	Mastership status flags	
		0	I2C gets bus mastership
		1	Lost bus mastership
2	**BUSY**	I2C bus BUSY status flag	
		0	Bus is idle
		1	Bus is busy
1	**TMODE**	Transmitting mode status	
		0	Receive mode
		1	Transmission mode
0	**RXACK**	RX ACK reception status flag	
		0	NOT received ACK
		1	Received ACK

10.4.3 I2Cn_SAR I^{2}C n Slave Mode Address Register

I2Cn_SAR is a register capable of 8-bit read / write access. The upper 7 bits stores the address selected when the device operates as a slave, and the lower 1 bit can select whether to receive the general call address from the I2C master bus.

I2C0_SAR=0x4000_0A0C, I2C1_SAR=0x4000_0A8C

31	30	29	28	27	26	25	24	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																								SVAD							GCALL
-																								0000000							0
-																								RW							RW

7 1	**SVAD**	7-bit slave address	
0	**GCALL**	General call control	
		0	Ignore general call address
		1	Enable general call detection

10.4.4 I2Cn_CR I²C n Rx/Tx Configuration Register

I2Cn_CR is an 8-bit read/write access register that provides settings for transmit/receive modes, interrupts, and internal delays via the serial bus.

I2C0_CR=0x4000_0A14, I2C1_CR=0x4000_0A94

31	30	29	28	27	26	25	24	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																						INTERVAL		IIF	Reserved	SOFTRESET	IINTEN	ACKEN	Reserved	STOP	START
-																						00		0	-	0	0	0	-	0	0
-																						RW		RO	-	WO	RW	RW	-	RW	RW

9 **8**	**INTERVAL**	The Options that add Internal delay to the i2c bus	
		00	(I2Cn_SCLL * 1) delay added
		01	(I2Cn_SCLL * 2) delay added
		10	(I2Cn_SCLL * 4) delay added
		11	(I2Cn_SCLL * 8) delay added
7	**IIF**	This flag indicates that an interrupt has occurred. To clear the IIF bit, the relevant interrupt flag must be cleared in the I2Cn_SR register.	
		0	NOT occurred interrupt
		1	Occurred interrupt
5	**SOFTRESET**	Initializing the internal register of I2C device.	
		0	No reset
		1	Reset
4	**IINTEN**	Enable I2C interrupt request	
		0	Disable interrupt
		1	Enable interrupt
3	**ACKEN**	Issue ACK signal in receive mode	
		0	Do NOT issue ACK signal after receiving data
		1	Issue ACK signal after receiving data
1	**STOP**	This field is used to stop transmission. If this bit is set to '1' during data transmission/receive, forced termination is performed during transmission. Even if an ACK signal is received after data transmission in transmission mode, if this bit is set to '1', transmission is terminated without transmitting the next data.	
		0	Continue communication
		1	Stop communication
0	**START**	This field is used to start in mater mode.	
		0	Wait in slave mode
		1	Start communication in master mode (At the end of the address phase, this field is automatically cleared.)

10.4.5 I2Cn_SCLL I^{2}C n SCL Low Period Register

I2Cn_SCLL is a 16-bit read / write register that can set the low period of the SCL signal when the I2C channel is operated in master mode.

I2C0_SCLL=0x4000_0A18, I2C1_SCLL=0x4000_0A98

31	30	29	28	27	26	25	24	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**															
-																**11111111 11111111**															
-																**RW**															

15 **0**	**SCLL**	Determine the period of the SCL low signal for I2C transmit rate in master mode. $Timimg = \frac{(1/\mathrm{PCLK})}{(\mathrm{SCLL} + 1)}$

10.4.6 I2Cn_SCLH I2C n SCL High Period Register

I2Cn_SCLH is a 16-bit read / write register that can set the high period of the SCL signal when the I2C channel is operated in master mode.

I2C0_SCLH=0x4000_0A1C, I2C1_SCLH=0x4000_0A9C

31	30	29	28	27	26	25	24	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															
-																11111111 11111111															
-																RW															

15 0	SCLH	Determine the period of the SCL high signal for I2C transmit rate in master mode. $Timimg = \frac{(1/\mathrm{PCLK})}{(\mathrm{SCLH}+3)}$

10.4.7 I2Cn_SDH I²C n SDA Hold Time Register

I2Cn_SDH is a 15-bit read / write access register that allows you to set a value to count how long the SDA value should remain at the previous value after the level of SCL falls from High to Low.

I2C0_SDH=0x4000_0A20, I2C1_SDH=0x4000_0AA0

31	30	29	28	27	26	25	24	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																	SDH														
-																	1111111 11111111														
-																	RW														

14 **0**	**SDH**	Set SDA Hold time $Timimg = \frac{(1/\mathrm{PCLK})}{(\mathrm{SDH}+4)}$

10.5 Functional Description : I2C Protocol

10.5.1 I^{2}C (Inter Integrated Circuit) Bit Transfer

The data on the SDA line must be stable during HIGH period of the clock, SCL. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

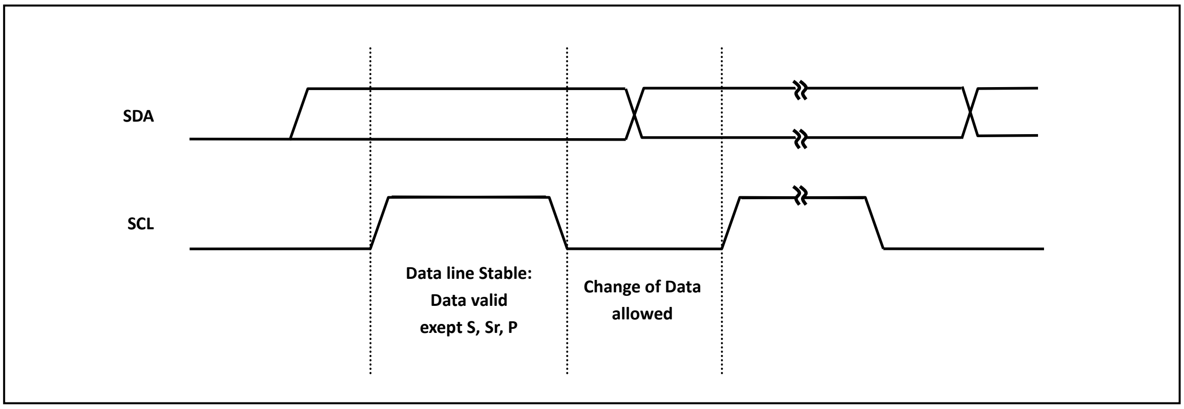


Figure 10.2. Bit Transfer on the I2C Interface Bus

10.5.2 START / Repeated START / STOP

One master device can send a START (S) signal to another device to communicate. When communication is complete, the master sends a STOP (P) signal to release the bus in use and make it available to other devices.

The conditions for determining START (S) and STOP (P) in I2C communication are as follows.
START (S) condition is that SDA line signal changes from 'H' to 'L' during SCL level is 'H'.
STOP (P) condition is that SDA line signal changes from 'L' to 'H' during SCL level is 'H'.

The START(S) and STOP(P) condition can be generated by the master at any time. After START (S) signal, the staus of the bus is busy. When the master sends a START (S) signal, the bus goes into the BUSY state. When the STOP signal is transmitted by the master, the bus is released. Then the state between START (S) and STOP (P) becomes BUSY.
For example, when the bus is between START and STOP, the bus remains in the BUSY state. If the condition is repeated START (Sr) instead of STOP, the state of the bus will be BUSY. START (S) and repeat START (Sr) are functionally the same.
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, ie, 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.

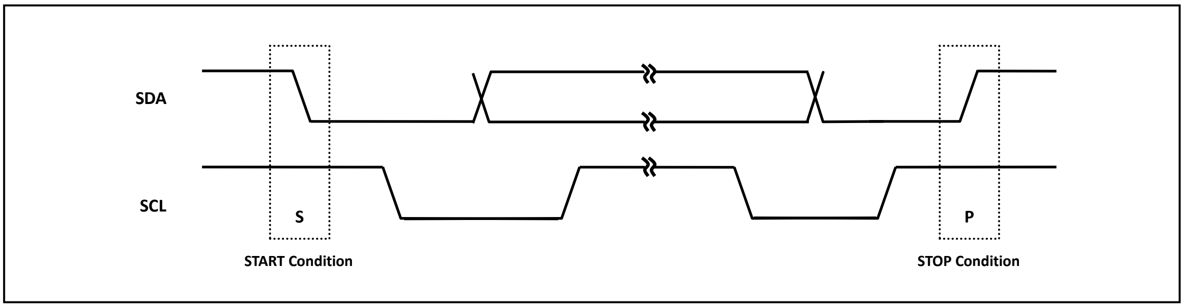


Figure 10.3. START and STOP Condition

10.5.3 Data Transfer

All data sent to the SDA line is configured in 8-bit units, and the number of bytes transferred is unlimited. SDA line transmits ACK bits after every data transmission. The 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 SCL 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 SCL.

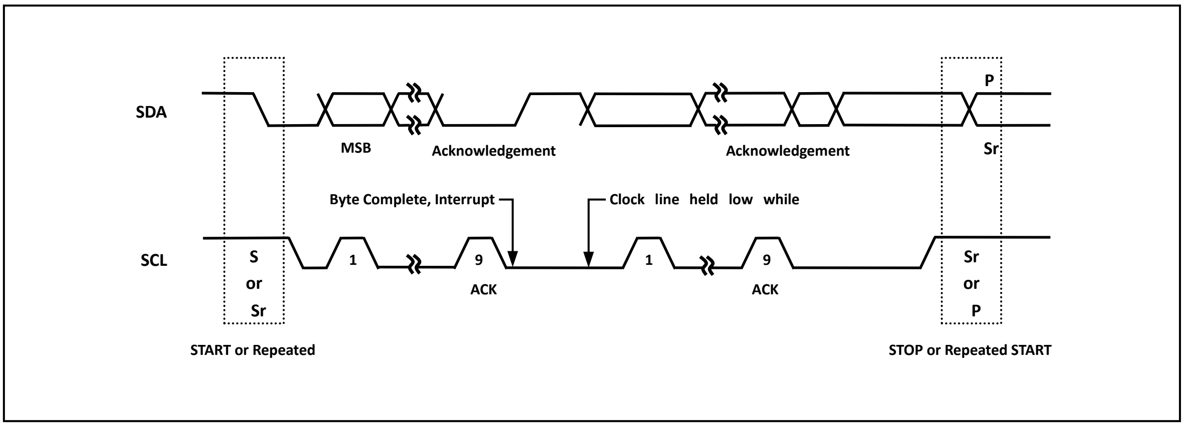


Figure 10.4. Data Transfer on the I²C bus

10.5.4 Acknowledge (ACK)

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA 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 SDA line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.
If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

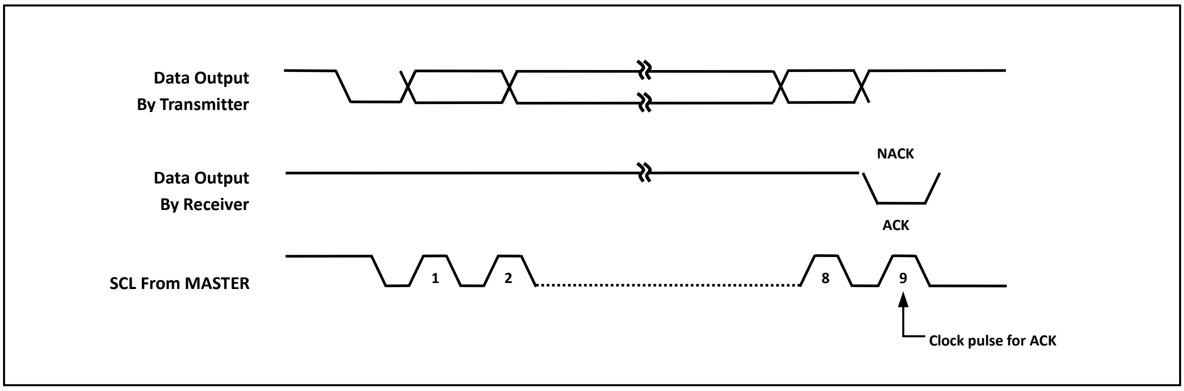


Figure 10.5. Acknowledge on the I²C Bus

10.5.5 Synchronization / Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and it will hold the SCL 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 SCL line if another clock is still within its LOW period. In this way, a synchronized SCL 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 SDA line, while the SCL 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.

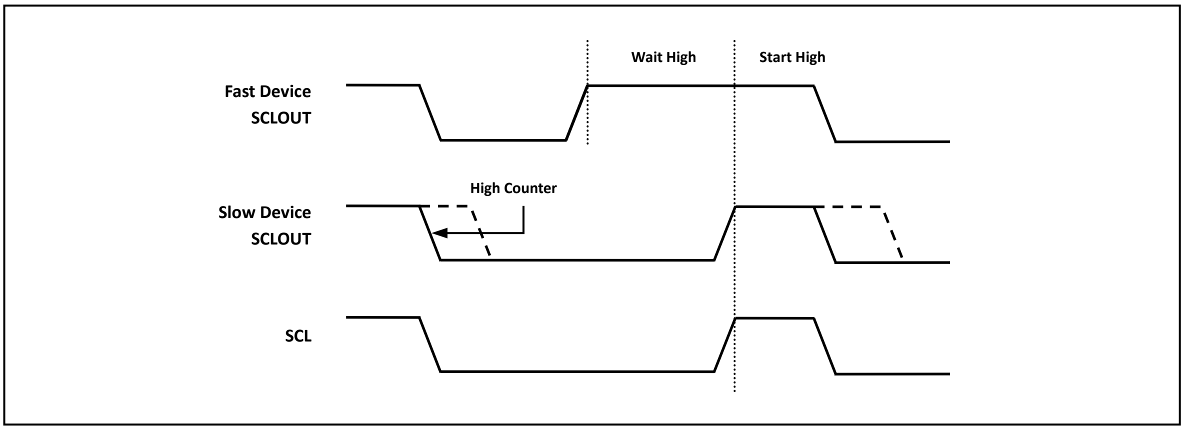


Figure 10.6. I²C clock synchronization during arbitration procedure

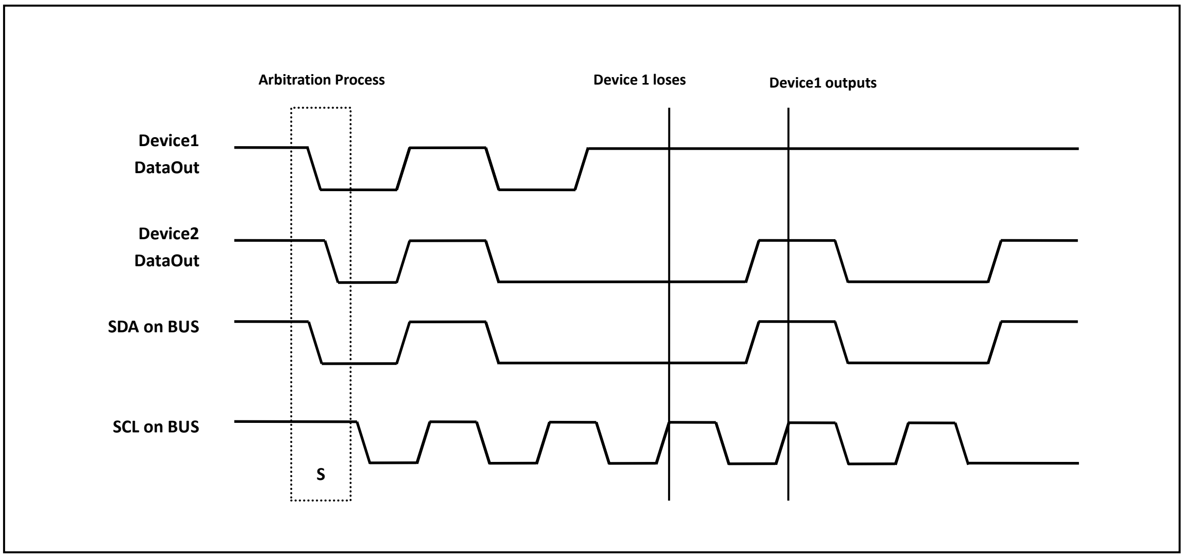


Figure 10.7. Arbitration precedure of two I²C masters

10.5.6 Debounce function for I2C

I2C is one of the standard serial communication protocols, which is widely used in industry. The I2C uses 2 bus lines such as Serial Data Line (SDAn) and Serial Clock Line (SCLn) to exchange data.
I2C features the followings (n = 0 and 1):

Table 10.5. The List of the I2C Channel

Channel	Pin Name		Description
I2C0	PB14	SCL0	Output or Schmitt-trigger input
	PB15	SDA0	Output or Schmitt-trigger input
I2C1	PD14	SCL1	Output or CMOS input
	PD15	SDA1	Output or CMOS input

For the input pins of SCL1, SDA1 CMOS type, it is recommended to debounce. The debounce settings of I2C1 channel is introduced in figure 10.8.
Each of SCD1 pin and SCA1 pin can use the debounce to prevent reset malfunction which is due to noise. This is configured in registers PD_DER (0x4000_026C) and the PD_DPR (0x4000_027C), and must be applied to the pin ports of SCL1 and SDA1. First, a user needs to set the debounce clock (prescaler) and enable the debounce filter by configuring the PD_DPR. When the filter is enabled, input signals in less than 4 clocks is considered as noise and ignored.

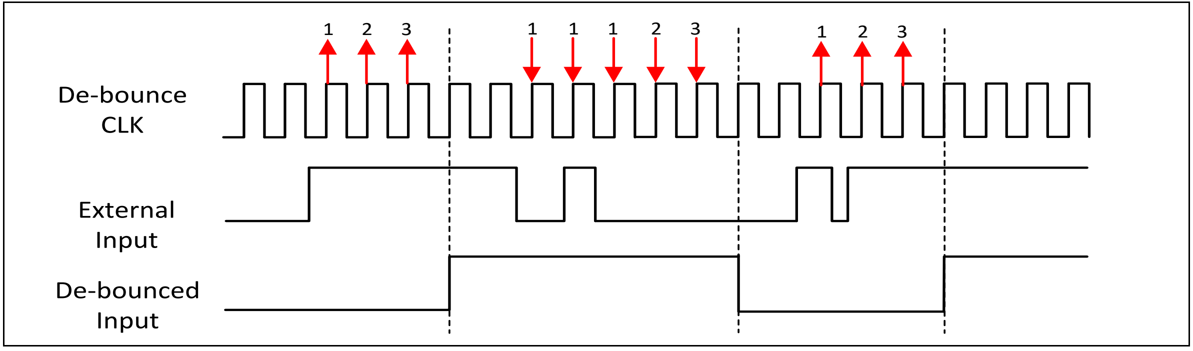


Figure 10.8. I²C clock synchronization during arbitration procedure

Assuming that input time of the PD_DPR is 16us and PCLK = 20MHz, the debounce filtering time is calculated as shown below (f = debounce clock frequency, T = debounce clock period).

$$4 \times T \geq 16\text{us}$$

$$T \geq 4us$$

$$\frac{1}{f} \geq 4us$$

$$\text{f} = \frac{PCLK}{2^x} \leq \frac{1}{4us} = \frac{1}{4}us$$

$$2^x \geq \frac{PCLK}{\frac{1}{4}MHz} = \frac{20MHz}{\frac{1}{4}MHz} = 80$$

$$\text{x} \geq 7$$

For Example,

$$f = \frac{PCLK}{2^7} = \frac{20MHz}{2^7} = \frac{20}{128}us$$

$$T = \frac{128}{20}us$$

$$Filtering\ time = 4T = 4 \times \frac{128}{20}us = 25.6$$

After the calculation, the result is '7'. Add the value to the PD_DPR to get 25.6us as the filtering time. For detailed information and relationship between the PD_DPR, PCLK, and debounce clock, please refer to spec. 4.5.15.
Example code shown below can be added during initialization process to complete software debounce processing for a PD pin.

```
PD_DPR <DPR>    = "7"
PD_DER <P14>    = "1"
```

10.6 Functional Description : 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 a I2C byte transfer.

Note that when a I2C interrupt is generated, IIF flag in I2CMR register is set, it is cleared by writing an arbitrary value to I2CSR. When I2C interrupt occurs, the SCL line is hold LOW until writing any value to I2CSR. When the IIF flag is set, the I2CSR contains a value indicating the current state of the I2C bus. According to the value in I2CSR, 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.

10.6.1 Master Transmitter

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

1. Enable I2C by setting IICEN bit in I2CMR. This provides main clock to the peripheral.
2. Load SLA+W into the I2CDR 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 I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLLR and I2CSCLHR for the Low and High period of SCL line.
4. Configure the I2CSDAHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLLR to the I2CSDAHR.
5. Set the START bit in I2CMR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the START bit is set, 8-bit data in I2CDR 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 whzther the slave acknowledged or not in the 9th high period of SCL. 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 MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLSOT bit in I2CSR is set, the ACKEN bit in I2CMR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL 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.

   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 I2CDR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOP bit in I2CMR.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2CDR and set START bit in I2CMR.

After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR and if transfer direction bit is '1' go to master receiver section.

7. 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 SCL LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOST bit in I2CSR is set. If then, I2C waits in idle state. When the data in I2CDR is transmitted completely, I2C generates TEND interrupt. I2C can choose one of the following cases regardless of the reception of ACK signal from slave.

   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 I2CDR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOP bit in I2CMR.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2CDR and set the START bit in I2CMR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR, 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 I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The next figure depicts above process for master transmitter operation of I2C

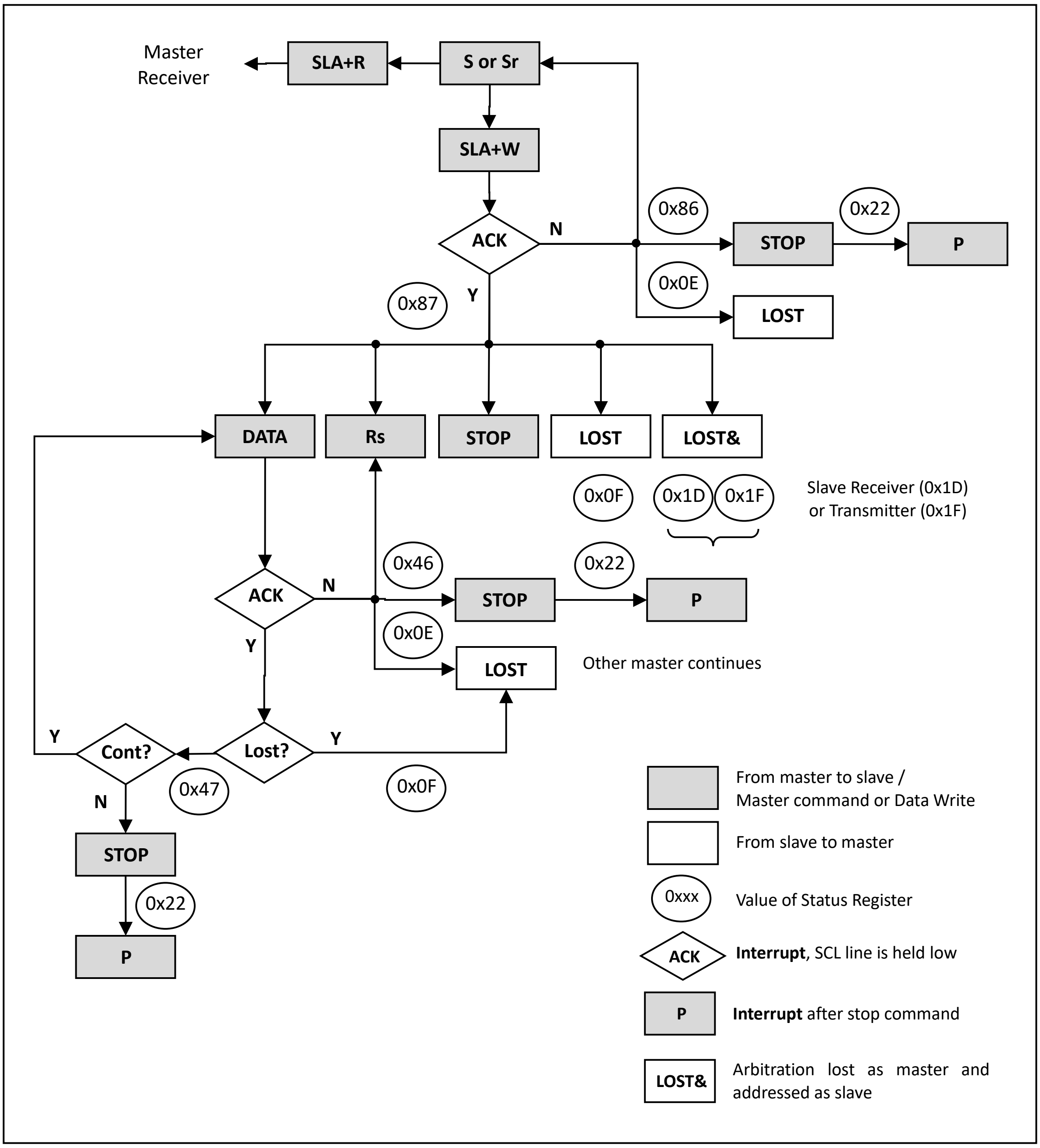


Figure 10.9. Formats and States in the master transmitter mode

10.6.2 MasterReceiver

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

1. Enable I2C by setting IICEN bit in I2CMR. This provides main clock to the peripheral.
2. Load SLA+R into the I2CDR 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 I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLLR and I2CSCLHR for the Low and High period of SCL line.
4. Configure the I2CSDAHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLLR to the I2CSDAHR.
5. Set the START bit in I2CMR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the START bit is set, 8-bit data in I2CDR 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 SCL. 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 MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLSOT bit in I2CSR is set, the ACKEN bit in I2CMR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL 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.
   1) Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKEN bit in I2CMR to decide whether I2C ACKnowledges the next data to be received or not.
   2) Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOP bit in I2CMR.
   3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA+R/W into the I2CDR and set START bit in I2CMR.

After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR and if transfer direction bit is '0' go to master transmitter section.

7. Byte of data is being received.
8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCL LOW. When 1-Byte of data is received completely, I2C generates TEND interrupt.

   I2C can choose one of the following cases according to the RXACK flag in I2CSR.
   1) Master continues receiving data from slave. To do this, set ACKEN bit in I2CMR to ACKnowledge the next data to be received.
   2) Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKEN bit in I2CMR.
   3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOP bit in I2CMR.
   4) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLA+R/W into the I2CDR and set the START bit in I2CMR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1) and 2), move to step 7. In case of 3), move to step 9 to handle STOP interrupt. In case of 4), move to step 6 after transmitting the data in I2CDR, 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 I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The processes described above for master receiver operation of I2C can be depicted as the following figure.

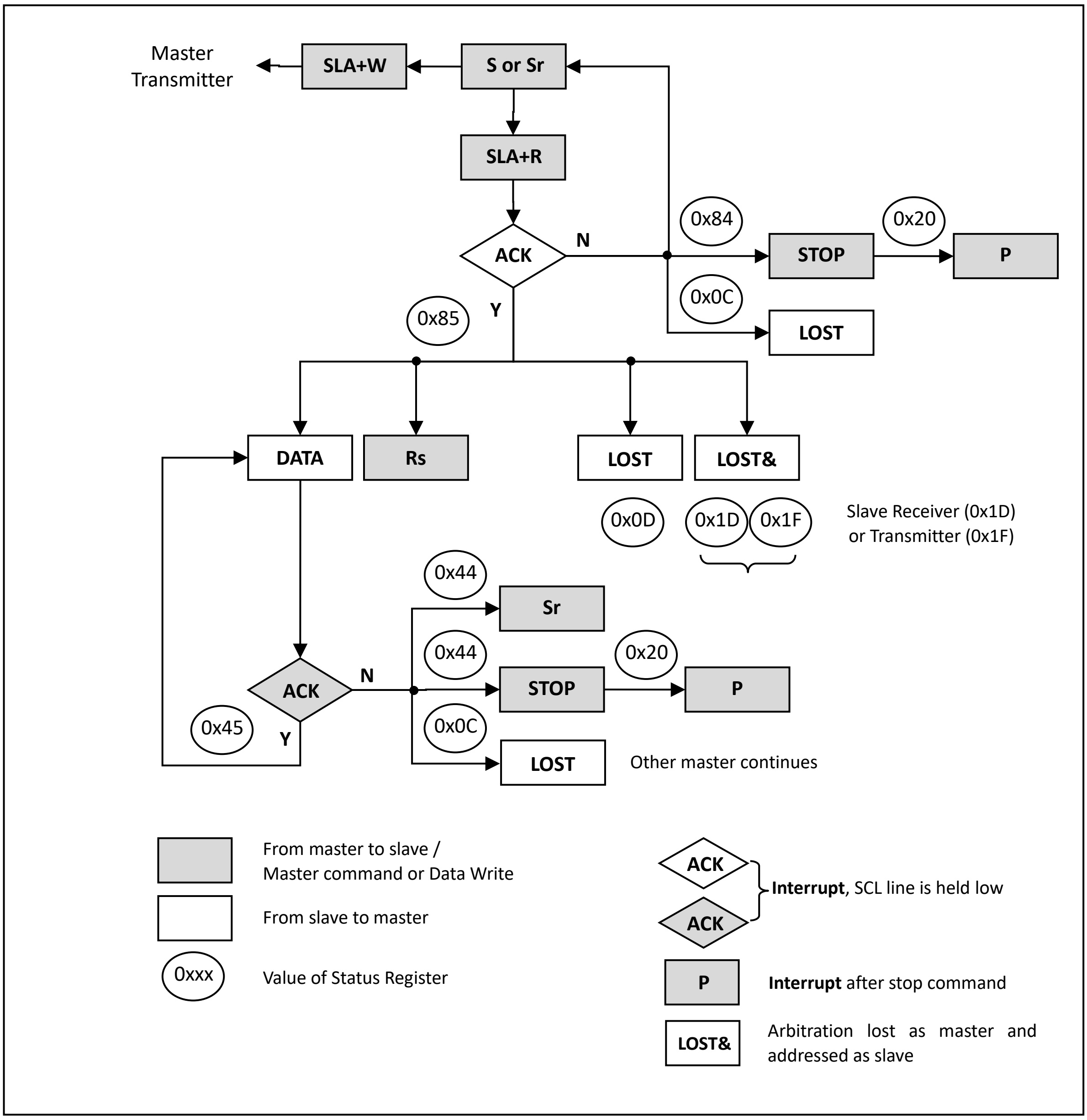


Figure 10.10. Formats and states in the master receiver mode

10.6.3 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 SCL, load value 0x00 into I2CSDAHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDAHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICEN bit and INTEN bit in I2CMR. 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 I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and the SCL line is held LOW. Note that even if the address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs, load transmit data to I2CDR and write arbitrary value to I2CSR to release SCL line.
5. Byte of data is being transmitted.
6. In this step, I2C generates TEND interrupt and holds the SCL line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   1) No ACK signal is detected and I2C waits STOP or repeated START condition.
   2) ACK signal from master is detected. Load data to transmit into I2CDR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The next figure shows flow chart for handling slave transmitter function of I2C.

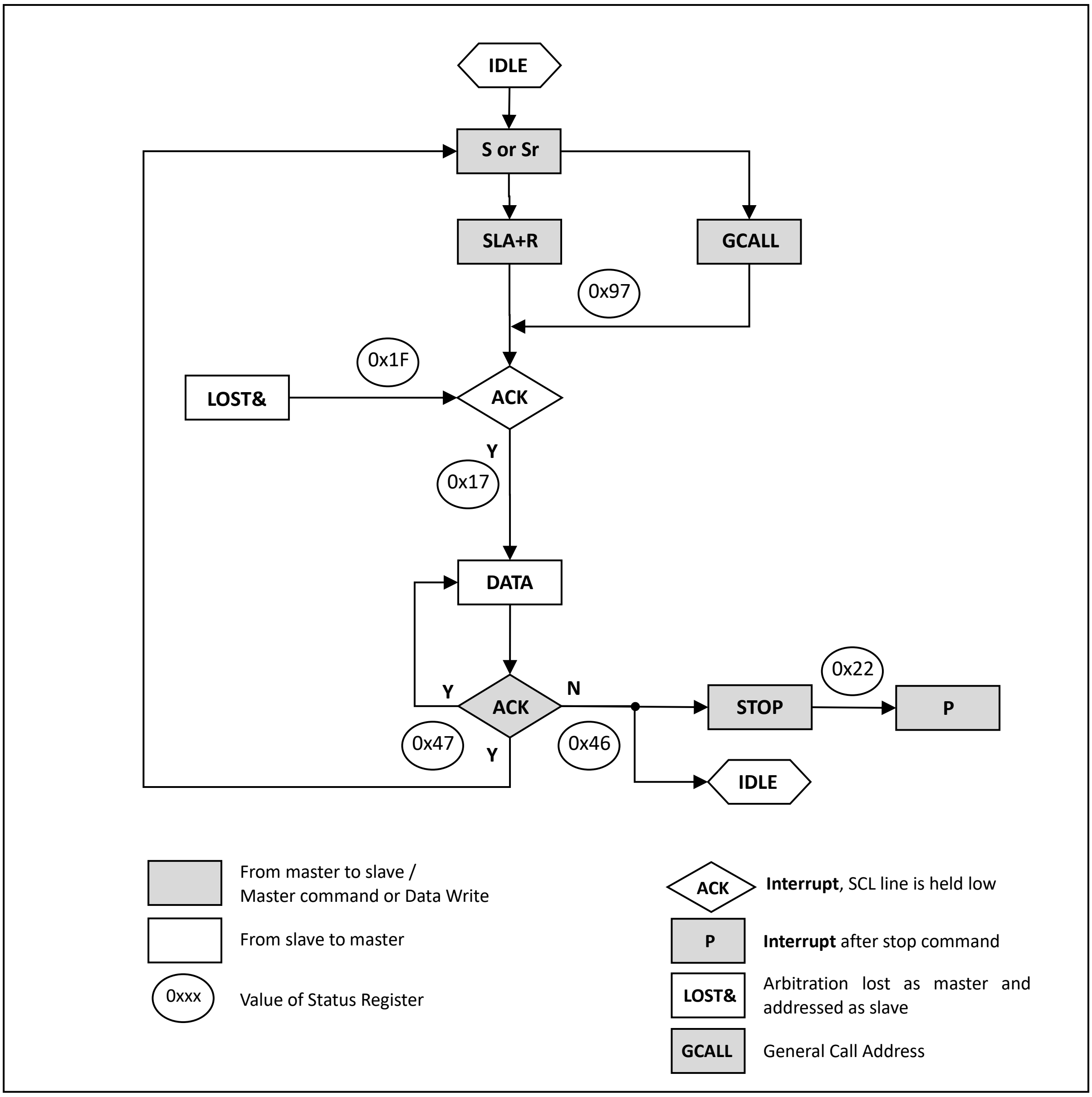


Figure 10.11. Formats and states in the slave transmitter mode

10.6.4 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 SCL, load value 0x00 into I2CSDAHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDAHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICEN bit and INTEN bit in I2CMR. 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 I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and the SCL line is held LOW. Note that even if the address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs and I2C is ready to receive data, write arbitrary value to I2CSR to release SCL line.
5. Byte of data is being received.
6. In this step, I2C generates TEND interrupt and holds the SCL line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.
   1) No ACK signal is detected (ACKEN=0) and I2C waits STOP or repeated START condition.
   2) ACK signal is detected (ACKEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The process can be depicted as following figure when I2C operates in slave receiver mode.

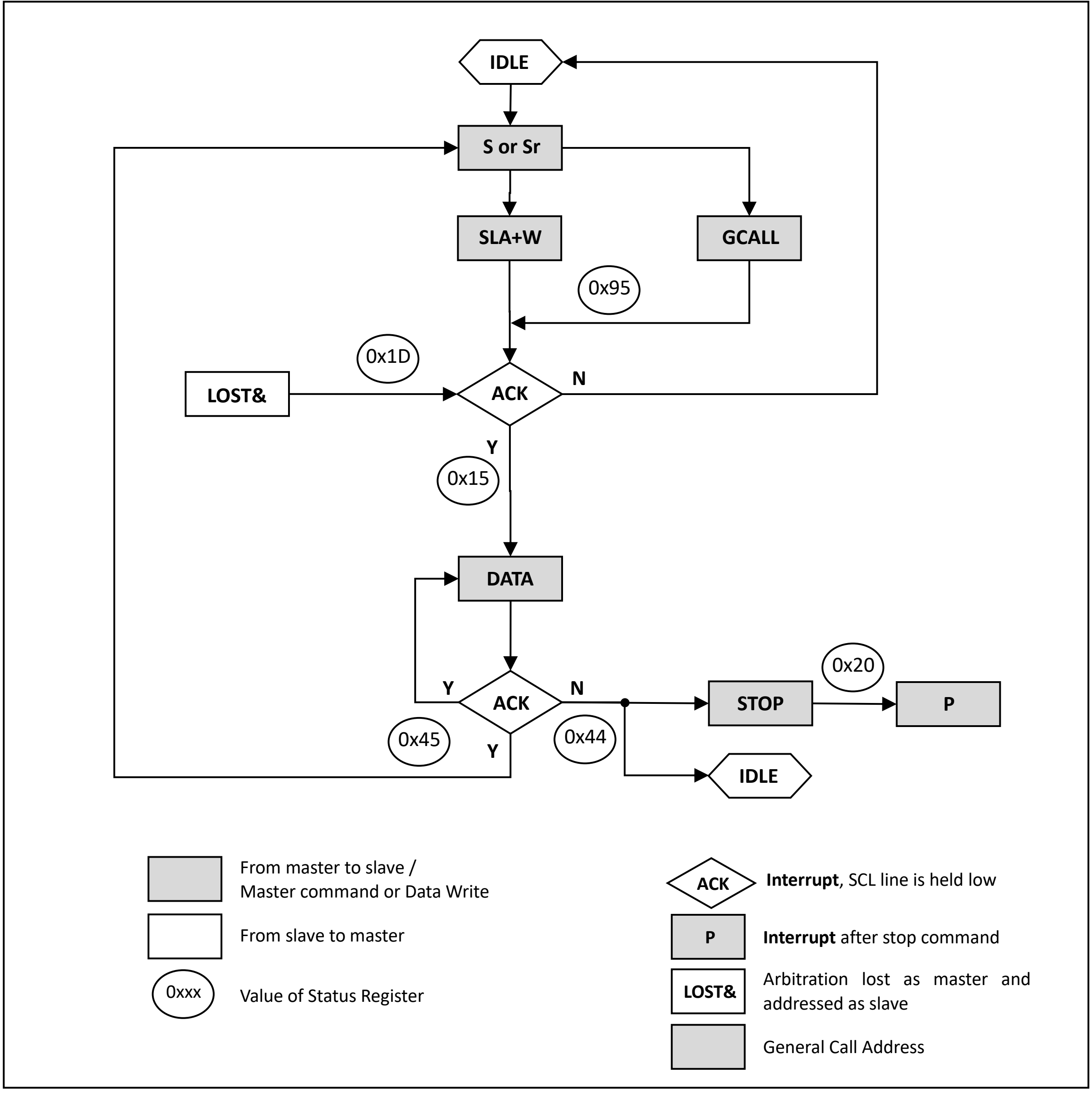


Figure 10.12. Formats and states in the slave receiver mode

10.6.5 Setting Example

<Example 1> Initialization of I2C0 Master

PMU_PER<I2C0> = “1”	: Enables the I2C0 channel
PMU_PCCR<I2C0> = “1”	: Enables the clock supply of I2C0 channel
PB_MR<P15[31:30]> = “01”	: Selects the function of PB15 pin as SDA0
PB_MR<P14[29:28]> = “01”	: Selects the function of PB14 pin as SCL0
PB_CR<P15[31:30]> = “01”	: Selects SDA0(PB15) pin as open drain
PB_CR<P14[29:28]> = “01”	: Selects SCL0(PB14) pin as open drain
PB_PCR<P15> = “0”	: Disables pull-up/pull-down function of SDA0(PB15) pin
PB_PCR<P14> = “0”	: Disables pull-up/pull-down function of SCL0(PB14) pin
I2C0_CR<SOFTRESET> = “1”	: Initializes the internal registers of I2C0 serial device
I2C0_CR<INTERVAL[9:8]> = “01”	: Configures the internal delay option value of I2C0
I2C0_CR<IINTEN> = “1”	: Enables the interrupt of I2C0 channel
I2C0_SCLL<SCL[31:0]> = ”0”	: Initializes SCL0 low period register
I2C0_SCLH<SCLH[31:0]> = “0”	: Initializes SCL0 high period register
I2C0_SDH<SDH[31:0]> = “0”	: Initializes SDA0 hold time register
I2C0_CR<ACKEN> = “1”	: Sets ACK signal output function after receiving data
I2C0_SAR<SVAD[7:1]> = “010 0000”	: Sets 7 bits slave address '0x20'

CHAPTER 11. UART (Universal Asynchronous Receiver / Transmitter)

Table 11.1. Operation Summary

Item	Description	Remark
Clock usage	PCLK	Transfer rate
Reset Source	Set by PMU_PER	
Reset Generation	None	
Interrupt Generation	Rx data, Tx data Line status Error (UART0(38),UART1(39),UART2(40),UART3(50))	UARTn_IER, UARTn_IIR
Interrupt Clear Method	Line status interrupt: read UARTn_LSR Rx interrupt: read UARTn_RBR Tx interrupt: read UARTn_IIR or write to UARTn_THR	UARTn_IIR, UARTn_LSR, UARTn_RBR, UARTn_THR

11.1 Overveiw

The A33G52x has 4 channels of UART (Universal Asynchronous Receiver / Transmitter) compatible with 16C550/16C450. Among these, UART0 and UART1 channels are 16550 with FIFO type, and UART2 and UART3 channels are 16450 with double buffer type. All UART channels operate in the same data mode (no FIFO mode) as 16450 when they are initialized. After initialization, FIFO mode setting is possible only for UART0 and UART1. In the FIFO mode, up to 16 bytes of data can be stored in the transmit/receive FIFO.

The built-in UART (Universal Asynchronous Receiver / Transmitter) can read out the data of the set configuration and the received data or the current UART status. UART status information can be checked not only for the type and conditions of transmission and reception by UART (Universal Asynchronous Receiver / Transmitter) but also for errors (parity, overrun, framing, break interrupt) that occur when data is received.

Each Universal Asynchronous Receiver / Transmitter (UART) has a programmable baud rate generator to divide the prescaled clock into values from 1 to 65535. This divided clock is again divided into 16 clocks to create a clock that drives the UART's internal transmit / receive blocks.

Communication with the UART can also be controlled by an interrupt using a user-programmable interrupt function. However, the general purpose 16450/16550 has a modem control signal and associated registers, but not the A33G52x.

Main features of UART (Universal Asynchronous Receiver / Transmitter)

- ■ 16550/16450 compatible asynchronous serial communication port 4 channels
  - 16550 compatible device with FIFO type 2 channels: UART0, UART1
  - 16450 compatible devices in double buffer type 2 channels: UART2, UART3
- ■ Serial interface settings
  - 5-bit, 6-bit, 7-bit or 8-bit data
  - Even, odd, or no-parity bit generation and detection
  - bit, 1.5-bit or 2-stop bit generation and detection
- ■ Add/Remove standard asynchronous communication bits (start, stop, and parity)
- ■ Independently configurable transmit, receive, line status, interrupt
- ■ 16-bit Programmable baud-rate generator
- ■ Built-in decimal point divider to improve baud rate accuracy.
- ■ Single and multi sampling function for received data
- ■ Support boot program using UART0 channel when entering boot mode
- ■ Status check function through UART interrupt ID register and line status register.
  - Stop bit error detection
  - Display information about current line status
  - Line break generation and detection
  - Receive error diagnosis function
- ■ Loop-back control
- ■ Priority-based interrupt system

11.2 Block Diagram

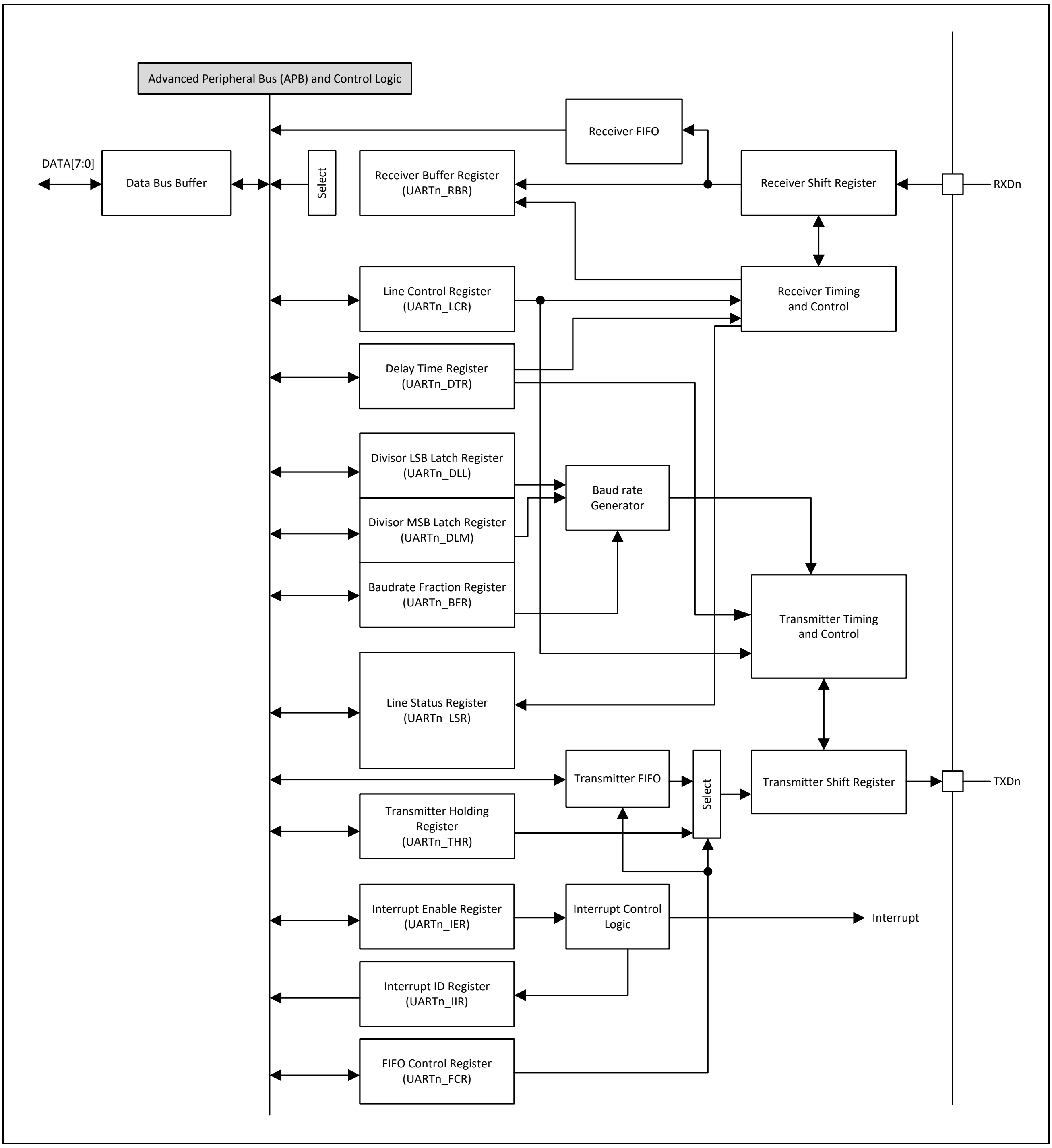

Advanced Peripheral Bus (APB) and Control Logic
DATA[7:0]
Data Bus Buffer
Select
Receiver FIFO
Receiver Buffer Register (UARTn_RBR)
Receiver Shift Register
RXDn
Line Control Register (UARTn_LCR)
Receiver Timing and Control
Delay Time Register (UARTn_DTR)
Divisor LSB Latch Register (UARTn_DLL)
Divisor MSB Latch Register (UARTn_DLM)
Baud rate Generator
Baudrate Fraction Register (UARTn_BFR)
Transmitter Timing and Control
Line Status Register (UARTn_LSR)
Transmitter FIFO
Select
Transmitter Shift Register
TXDn
Transmitter Holding Register (UARTn_THR)
Interrupt Enable Register (UARTn_IER)
Interrupt Control Logic
Interrupt
Interrupt ID Register (UARTn_IIR)
FIFO Control Register (UARTn_FCR)

Figure 11.1. The block diagram of UART

11.3 Pin Configuration

Table 11.2. The List of UART Pins

Pin Name	Type	Description	Support by package		
			A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM	A33G526MM A33G526ML A33G524MM A33G524ML	A33G527VQ A33G527VL A33G526VQ A33G526VQ
TXD0	O	Serial output of UART0	O	O	O
RXD0	I	Serial input of UART0	O	O	O
TXD1	O	Serial output of UART1	O	O	O
RXD1	I	Serial input of UART1	O	O	O
TXD2	O	Serial output of UART2	O	O	O
RXD2	I	Serial input of UART2	O	O	O
TXD3	O	Serial output of UART3	O	O	O
RXD3	I	Serial input of UART3	O	O	O

11.4 Register Map

The A33G52x has four UART channels. The base address of the UART module is 0x4000_0B00, and the address of each UART channel is as follows.

Table 11.3. The address list for each channel

Port	Address	Remark
UART0	0x4000_0B00	16-depth FIFO
UART1	0x4000_0B40	16-depth FIFO
UART2	0x4000_0B80	Double-buffer
UART3	0x4000_0BC0	Double-buffer

The register list of a UART is shown as follows

Table 11.4. Register list of a UART

Register	Offset	Access Type	Description	Initial Value	Ref
UARTn_RBR	0x00	RO	UART n Rx Buffer Register (DLAB=0)	0x00000000	11.5.1
UARTn_THR	0x00	WO	UART n Tx Data Hold Register (DLAB=0)	0x00000000	11.5.2
UARTn_DLL	0x00	RW	UART n Divisor Latch Register LSB (DLAB=1)	0x00000000	11.5.3
UARTn_DLM	0x04	RW	UART n Divisor Latch Register MSB (DLAB=1)	0x00000000	11.5.4
UARTn_IER	0x04	RW	UART n Interrupt Enable Register	0x00000000	11.5.5
UARTn_IIR	0x08	RO	UART n Interrupt ID Register	0x00000001	11.5.6
UARTn_FCR	0x08	WO	UART n FIFO Control Register	0x00000000	11.5.7
UARTn_LCR	0x0C	RW	UART n Line Control Register	0x00000000	11.5.8
UARTn_LSR	0x14	RO	UART n Line Status Register	0x00000060	11.5.9
UARTn_SCR	0x1C	RW	UART n Scratch pad Register	0x00000000	11.5.10
UARTn_BFR	0x24	RW	UART n Divisor Decimal Point Calculation Register	0x00000000	11.5.11
UARTn_DTR	0x28	RW	UART n Delay Time Register	0x00000000	11.5.12

11.5 Register Description

11.5.1 UARTn_RBR UART n Rx Buffer Register

The UART receive data buffer register can be read with a maximum length of 8 bits. Serial input data is stored in the receive buffer register. The last receive data is stored in this register until a new bit is received.
This register is a read-only register and can be accessed when DLAB (the seventh bit in the UARTn_LCR register) is set to '0'.

UART0_RBR=0x4000_0B00, UART1_RBR=0x4000_0B40

UART2_RBR=0x4000_0B80, UART3_RBR=0x4000_0BC0

31	30	29	28	27	26	25	24	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							
-																								XXXXXXXX							
-																								RO							

7 0	RBR	Received data byte

11.5.2 UARTn_THR UART n Tx Buffer Register

The data to be output to the serial communication port of the UART transmit data buffer register is written with a maximum length of 8 bits. This register is a write-only register and can be accessed when DLAB (7th bit in UARTn_LCR register) is '0'.

UART0_THR=0x4000_0B00, UART1_THR=0x4000_0B40

UART2_THR=0x4000_0B80, UART3_THR=0x4000_0BC0

31	30	29	28	27	26	25	24	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							
-																								XXXXXXXX							
-																								WO							

7 0	THR	Data byte to be transmitted

11.5.3 UARTn_DLL UART n Divisor Latch Register LSB

UART0_DLL=0x4000_0B00, UART1_DLL=0x4000_0B40

UART2_DLL=0x4000_0B80, UART3_DLL=0x4000_0BC0

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">Reserved</td><td colspan="8">DLL</td></tr>
<tr><td colspan="24">-</td><td colspan="8">00000000</td></tr>
<tr><td colspan="24">-</td><td colspan="8">RW</td></tr>
</table>

7 **0**	**DLL**	Dividor latch low byte

11.5.4 UARTn_DLM UART n Divisor Latch Register MSB

UART0_DLM=0x4000_0B04, UART1_DLM=0x4000_0B44

UART2_DLM=0x4000_0B84, UART3_DLM=0x4000_0BC4

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">Reserved</td><td colspan="8">DLM</td></tr>
<tr><td colspan="24">-</td><td colspan="8">00000000</td></tr>
<tr><td colspan="24">-</td><td colspan="8">RW</td></tr>
</table>

7 **0**	**DLM**	Dividor latch high byte

Each channel of the UART has a programmable baud rate generator that can receive the PCLK clock and divide it by a divide value from 2 to 65535. (If dividing value is '1', it is the highest baud rate.). The output clock frequency of the baud rate generate is 16 x Baud rate. In order for UART communication to work properly, the DLL and DLM registers value must be written to appropriate values before UART operation.

For more information on the baud rate calculation formula, please refer to 11.6.8.

11.5.5 UARTn_IER UART n Interrupt Enable Register

This register enables four types of UART interrupts. Each interrupt generates an interrupt output signal. Setting all bits in the interrupt enable register (IER) to 0 disables all interrupts in the UART and sets the corresponding bit in UARTn_IER to 1 to enable only certain interrupts. If the interrupt is disabled, it is not written to the UARTn_IIR register. Even if the corresponding interrupt condition occurs, the interrupt output signal is not generated. All other UART functions, such as the status display of the UARTn_LSR register, operate identically regardless of the UARTn_IER setting.

UART0_IER=0x4000_0B04, UART1_IER=0x4000_0B44

UART2_IER=0x4000_0B84, UART3_IER=0x4000_0BC4

31	30	29	28	27	26	25	24	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																												TEMTIE	RLSIE	THREIE	DRIE
-																												0	0	0	0
-																												RW	RW	RW	RW

3	**TEMTIE**	Disable/Enable transmission empty interrupt.	
		0	Disable
		1	Enable
2	**RLSIE**	Disable/Enable receive line status interrupt	
		0	Disable
		1	Enable
1	**THREIE**	Disable/Enable THR(Transmit Holding Register) empty interrupt	
		0	Disable
		1	Enable
0	**DRIE**	Disable/Enable data receive ready interrupt **[in double buffer mode]** If this bit is enabled in double buffer mode, data receive ready interrupt is enabled. **[in FIFO (First in, first out) mode]** If this bit is enabled in FIFO mode, data receive ready interrupt and character time-out interrupt are enabled.	
		0	Disable
		1	Enable

11.5.6 UARTn_IIR UART n Interrupt Identification Register

This register is an 8-bit read-only register that is displayed as an ID value for each interrupt operation set in the UART interrupt enable register (UARTn_IER) register. When CORE accesses the UARTn_IIR register, the UART locks all interrupts and the highest priority interrupt is read. Interrupts occur while CORE is accessed, but do not change state until the current access is completed.

UART0_IIR=0x4000_0B08, UART1_IIR=0x4000_0B48

UART2_IIR=0x4000_0B88, UART3_IIR=0x4000_0BC8

31	30	29	28	27	26	25	24	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																								XMITE		Reserved	XMITE	FID	IID		IPEN
-																								0		-	0	0	00		1
-																								RO		-	RO	RO	RO		RO

7 **6**	**FIFO**	Display the current UART operation mode	
		00	None-FIFO (Double buffer mode)
		11	FIFO mode
4	**XMITE**	0	NOT occurred transmit empty interrupt
		1	Occurred transmit empty interrupt
3 **1**	**IID/FID**	The ID value of interrupt (Figure 11.5)	
0	**IPEN**	Display the status of interrupt	
		0	Pending a interrupt to be processed
		1	None of interrupt

The UART supports interrupts with three levels of priority and displays the current highest priority interrupt ID in the current UARTn_IIR. Three levels of interrupts are configured with the following priorities:
1) Receive Line Status Interrupt
2) Receive data ready interrupt / Character time-out interrupt
3) Transmit holding register empty register

Detail description is below:

IPEN This bit indicates whether there are currently unprocessed interrupts. If IPEN = '0', it means that a specific interrupt condition has occurred. IID and FID of UARTn_IIR can be used as a pointer to the interrupt processing routine. If IPEN = '1', it indicates that an unhandled interrupt is not present.

IID This 2-bit value indicates the ID of the highest priority interrupt currently occurring. This is covered in detail in "Figure 11.5 Interrupt list and way for handling interrupt".

FID 16450 mode, this bit is always 0, and in FIFO mode, along with IID, indicates that the time-out interrupt is the highest priority interrupt currently generated.

FIFO These two bits become '1' when UARTn_FCR <FIFOEN> = 1.

Table 11.5. Interrupt list and way for handling interrupt

Priority Level	FIFO Mode Only	IIR			Interrupt occurrence and clear condition		
	Bit 3	Bit 2	Bit 1	Bit 0	Interrupt Name	Interrupt Source	Interrupt Clear
-	0	0	0	1	None	None	None
1	0	1	1	0	Receiver Line Status	Over-run, Parity, Framing, Break, etc.	Read UARTn_LSR register
2	0	1	0	0	Receiver Data Available	Recived data exists Or Recieve FIFO level is reached	Read receive buffer data Or Receive data is read until it is below FIFO level
2	1	1	0	0	Character Time-out Indication	Continuous data is NOT received for 4 character time after a data below FIFO level is received.	Read receive buffer data
3	0	0	1	0	Transmitter Holding Register Empty	THR (Transmitter Holding Register) is empty.	Read UARTn_IIR register or Write a value to UARTn_THR register
4	0	0	0	0	N.A	-	-

11.5.7 UARTn_FCR UART n FIFO Control Register

This register is a write-only 8-bit register that has the same location as UARTn_IIR, and is a register that can be used by the UART0 and UART1 channels only. (Conversely, UARTn_IIR is a read-only register). The UARTn_FCR register is used to enable the FIFO, clear the transmit/receive FIFO, and set the trigger level of the receive FIFO.

UART0_FCR=0x4000_0B08, UART1_FCR=0x4000_0B48

UART2_FCR=0x4000_0B88, UART3_FCR=0x4000_0BC8

31 30 29 28 27 26 25 24 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	FIFODEPTH	Reserved	FCR2	FCR1	FIFOEN
-	0	-	0	0	0
-	WO	-	WO	WO	WO

7 6	FIFODEPTH	Receiving FIFO trigger level	
		00	1 byte
		01	4 bytes
		10	8 bytes
		11	14 bytes
2	FCR2	0	Initial value
		1	Initialize transmit FIFO
1	FCR1	0	Initial value
		1	Initialize receive FIFO
0	FIFOEN	0	Disable FIFO
		1	Enable transmit/receive FIFO

The functions supported in the UARTn_FCR register are described below.

FIFOEN As writing “1” to this field, both transmit and receive FIFO are enabled and UART enters the FIFO mode. If UARTn_FCR<FIFOEN> is set to “0”, all data in the FIFO is erased while mode is changed from the FIFO to 16450. Although FIFOEN is set to “1” when UARTn_FCR<FIFOEN> is “0”, all data in FIFO is erased, too. If UARTn_FCR<FIFOEN> is not “1”, the others of the UARTn_FCR cannot be written.

FCR1 Writing “1” to the UARTn_FCR<FCR1> clears all data in the RCVR FIFO and reset its counter logic to “0”. (The receiver shift register is not cleared). Written data “1” is automatically cleared to “0”.

FCR2 Writing “1” to the UARTn_FCR<FRT2> clears all data in the XMIT FIFO and reset its counter logic to “0”. (Transmitter shift register is not cleared) Written data “1” is automatically cleared to “0”.

FIFODEPTH This bit is used to specify level for interrupt generation of the RCV FIFO.

11.5.8 UARTn_LCR UART n Line Control Register

This register can set the asynchronous data transmit / receive format of the UART line or set the divisor latch access bit (DLAB).
The UARTn_LCR register is able to read / write access. It is not necessary to separately store the current format of the asynchronous data transmission / reception line in the system memory.

UART0_LCR=0x4000_0B0C, UART1_LCR=0x4000_0B4C

UART2_LCR=0x4000_0B8C, UART3_LCR=0x4000_0BCC

31	30	29	28	27	26	25	24	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																								DLAB	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN	
-																								0	0	0	0	0	0	00	
-																								RW	RW	RW	RW	RW	RW	RW	

7	**DLAB**	0	Normal transmission
		1	Divisor latch access mode
6	**BREAK**	0	Normal transmission
		1	Send break
5	**STICKP**	0	Stick parity as ‘L’
		1	Stick parity as ‘H’
4	**PARITY**	0	Odd parity
		1	Even parity
3	**PEN**	0	Disable parity
		1	Enable parity
2	**STOPBIT**	0	1 stop bit
		1	1.5 / 2 stop bits
1 **0**	**DLEN**	00	5-bit data
		01	6-bit data
		10	7-bit data
		11	8-bit data

It is possible to make an asynchronous data transmit/receive configuration and set DLAB to write data on divisor latch register. LCR is r/w register, as using this register, it is not need to store the configuration of current asynchronous data transmit/receive. See the next description for details.

DLEN DLEN is used to specify the number of bit for a transmit/receive data word

STOPBIT This bit represents a type of stop bit which is inserted at the end of transmit/receive data. If this bit is “0”, 1-stop bit is inserted at the end of the transmit data. When this bit is”1”, if the word length is set to 5bit by DLEN, 1.5-stop bit is inserted at the end of the transmit data and stop bit will be 2-bit in the case of 6~8 bit of word length. Receiver checks the first bit of stop bit regardless of the number of stop bit selected by DLEN and STOP.

PEN This bit is parity enable bit. When PEN is “1”, parity bit is generated between last data bit and stop bit in transmitter, then receiver checks parity bit between last data bit and stop bit. (parity bit is generated to make the number of bits with value of “1” in a given set of data and parity to even or odd according to the configuration of PARITY and STICKP bits.)

PARITY This is a parity selection bit. If PEN is “1” and PARITY is “0”, the number of “1” should be odd. If PEN is “1” and PARITY is “1”, the number of “1” should be even. The number of “1” includes all “1” in data and parity.

STICKP This bit is Stick parity. If PEN, PARITY and STICKP are all “1”, parity bit always should be “0”. If PEN and STICKP is “1” and PARITY is “0”, parity bit always should be “1”. When STICKP is “0”, stick parity is disabled.

BREAK This is break control bit. If BREAK set to “1”, serial output (TxD) become”0” (spacing state). Break condition is cleared by setting BREAK bit to "0"

** Note : This function is used to warn on the opposite side in communication system. In the case that break function is used, transfer can stop by "break"

DLAB This bit is a divisor latch access bit to control an access to DLL/DLM register. In order to access divisor latch register to control baud generator, this bit should be "1". To access receive buffer, Transmit hold register and interrupt enable register, DLAB should be "0".

11.5.9 UARTn_LSR UART n Line Status Register

UARTn_LSR is a read-only register that shows information on the status of data transmission and reception.

UART0_LSR=0x4000_0B14, UART1_LSR=0x4000_0B54

UART2_LSR=0x4000_0B94, UART3_LSR=0x4000_0BD4

31	30	29	28	27	26	25	24	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																								FIFOE	TEMT	THRE	BI	FE	PE	OE	DR
-																								0	1	1	0	0	0	0	0
-																								RO	RO	RO	RO	RO	RO	RO	RO

7	**FIFOE**	0	FIFO is valid
		1	FIFO has invalid data
6	**TEMT**	0	Transmitter NOT empty
		1	Transmitter empty
5	**THRE**	0	THR NOT empty
		1	THR Empty
4	**BI**	0	No break detection
		1	Break interrupted
3	**FE**	0	No framing error
		1	Framing error
2	**PE**	0	No parity error
		1	Parity error
1	**OE**	0	No overrun error
		1	Overrun error
0	**DR**	0	No data received
		1	Received data ready

This register represents transmit/receive status. The next describes each field in details:

DR This bit represent that data is completely received and moved to receive FIFO. If receive buffer register or FIFO is read, DR becomes to “0”

OE This bit represents overrun error. OE will be set if next receive data is moved to receive buffer register before previous receive data is read. OE is cleared to “0” when LSR is read by CPU. In FIFO mode, if the next receive data is completely received to the shift register after 16 bytes of FIFO is filled to the full with received data, OE is set to “1”. At this time, the data in receive register is damaged but it is not moved to FIFO.

PE This is parity error indicator. PE is set to “1” when received data did not satisfy parity condition in LCR. PE is cleared to “0” when LSR is read by CPU. In FIFO mode, PE becomes “1” when parity error data is placed on the top of the FIFO.

FE This is framing error indicator that represents stop bit of received data is not correct. FE is set when stop bit that follows after the last data bit or parity bit is detected to “0”. FE is cleared to “0” as reading LSR by ARM7TDMI core. In FIFO mode, PE set to “1” when framing error data is placed on the top of the FIFO. In order to resynchronize the receiver again, UART starts next word receive with assumption that UART framing error is generated by start bit of next receive data.

BI BI is break interrupt bit. BI is set when received data input keeps “0” for more time than total data receive time(the total receive time for start bit, data bit, parity bit, and stop bit). BI is cleared as reading by ARM7TDMI core. In FIFO mode, PE is set to “1” when break error data is placed on the top of the FIFO. If Break condition occurs, one zero data is sent to the FIFO. Next input of data starts when new start bit is received after receiver becomes “1”.

Note) Bit 1, 2, 3, and 4 of LSR causes the receive line status interrupt when corresponding interrupt is enabled.

THRE This bit represents transmit holding register empty (THRE). THRE is a cause of interrupt when transmitter holding register empty interrupt (THREIE) is enabled and UART is ready to receive new data from CPU. THRE is set to “1” when transmitter holding register is possible to be written because transfer data is moved from transmitter holding register to transmitter shift register. If new data is written to transmitter holding register, it is cleared to “0”. In FIFO mode, THRE is set to “1” when there are not any data in XMIT FIFO, and THRE is set to “0” when there are one or more data in XMIT FIFO.

TEMT This bit represents transmit data empty. TEMP is set to “1” when transmit holding register (THR) and transmit shift register (TSR) are empty and TEMP is set to “0” when there are data in THR or TSR. In FIFO mode, TEMP is set to “1” when XMIT FIFO and TSR are all empty.

FIFOE In 16450 mode , this bit is always “0”. In FIFO mode, FIFOE is set to “1” when there is one or more event among parity error, framing error, or break detection. FIFOE is cleared to “0” when all data with error are read and LSR is read by Core.

Note) Line status register is read-only register.

11.5.10 UARTn_SCR UART n Scratch Register

This 8-bit register is independent of the UART operation and settings. This register is prepared for the software to be used as temporary space for storing data.

UART0_SCR=0x4000_0B1C, UART1_SCR=0x4000_0B5C

UART2_SCR=0x4000_0B9C, UART3_SCR=0x4000_0BDC

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">Reserved</td><td colspan="8">SCR</td></tr>
<tr><td colspan="24">-</td><td colspan="8">00000000</td></tr>
<tr><td colspan="24">-</td><td colspan="8">RW</td></tr>
</table>

7 **0**	**SCR**	Scratch byte data

11.5.11 UARTn_BFR UART n Divisor Decimal Point Calculation Register

This 8-bit register is used to increase the baud rate accuracy.
Calculate the UART baud rate using UARTn_DLM and UARTn_DLL registers, and if there is a decimal point value, multiply the decimal point part by '256' and set the integer value in UARTn_BFR register.

UART0_BFR=0x4000_0B24, UART1_BFR=0x4000_0B64

UART2_BFR=0x4000_0BA4, UART3_BFR=0x4000_0BE4

31	30	29	28	27	26	25	24	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**							
-																								**00000000**							
-																								**RW**							

7 **0**	**BFR**	Divisor Decimal Point Calculation Register

Table 11.6 below shows the BFR values for each baud rate of the UART channel based on the 75 MHz peripheral clock. You can set the baud rate more accurately by referring to the BFR value.

Table 11.6.Fractional counter configuration table (PCLK=75MHz)

Baud rate (bps)	Divider(BDR) (DLM:DLL)	Decimal Point Value	FCNT	BFR	Error
1200	1953.13	0.13	32.00	32	0.00%
2400	976.56	0.53	144.00	144	0.00%
4800	488.28	0.28	72.00	72	0.00%
9600	244.14	0.14	36.00	36	0.00%
19200	122.07	0.07	18.00	18	0.00%
38400	61.04	0.04	9.00	9	0.00%
57600	40.69	0.69	176.67	176	0.38%
115200	20.35	0.35	88.33	88	0.38%
230400	10.17	0.17	44.17	44	0.38%
460800	5.090	0.09	22.08	22	0.38%

FCNT= (Decimal point value) *256.
In this case, the decimal point value is derived from decimal point of the BDR that calculated the UARTn_DLM, UARTn_DLL, PCLK, and baud rate values.
For example, FCNT value can calculated above equiation. For example, the target baud rate is 2400 bps and UART_PCLK is 75MHz case, the BDR value is 976.56. The integer number 976 should be the BDR value and the floating number 0.56 will make the FNCT value as below.
FCNT = 0.56 * 256 = 144, So the FCNT value is 144.
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.

11.5.12 UARTn_DTR UART n Delay Time Register

This UARTn_DTR register adjusts the delay time between UART start bit or data bit frame to prevent start error due to noise during sampling or set inverted input or output of UART Rx/Tx signal.

UART0_DTR=0x4000_0B28, UART1_DTR=0x4000_0B68

UART2_DTR=0x4000_0BA8, UART3_DTR=0x4000_0BE8

31	30	29	28	27	26	25	24	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	RXINV	TXINV	Reserved	WAITVAL		
-																								0	0	0	0	-	000		
-																								RW	RW	RW	RW	-	RW		

7	**SMS**	0	Start bit sampling 1 time run
		1	Start bit sampling 3 time run (Start error prevention due to noise)
6	**DMS**	0	Data bit sampling 1 time run
		1	Data bit sampling 3 time run (Error prevention due to noise)
5	**RXINV**	0	No external RX input change
		1	External RX input inversion
4	**TXINV**	0	No TX output change
		1	TX output inversion
2 **0**	**WAITVAL**	To set the delay time between the output of the continuous data. $\text{Wait time } (W_T) = \frac{WAITVAL}{Baud\ rate}$	

SMS DTR register adjusts the delay time between the UART start bit or data bit to prevent start-up errors due to noise during sampling or to set the inverted input / output of the UART Rx/Tx signal. This bit is '1'. At this time, the value is output when all the acquired sample values are 1 or 0.
Note) If the channel is disabled in the PMU_PER register after powering on the MCU, the SMS bit value is automatically set to '1' when the UART n channel is enabled.

DMS If DMS value is '1', Sampling of received data bit is performed 3 times. At this time, the most frequent value of each sample value is output.

RXINV If the RXINV bit is '1', the external RX input signal is inverted and input to the inside.

TXINV When the TXINV bit is '1', the internal TX output signal is inverted and output to the outside.

WAITVAL By entering a value from 0 to 4095 in this WAITVAL field, the delay time between successive data outputs can be adjusted. At this time, the delay time (WT) can be obtained by the following formula.

$$\text{Wait time } (W_T) = \frac{WAITVAL}{Baud\ rate}$$

** Note : Sysem Parameter 1(0x0020_0000~0x0020_01FF) area contains commands related to the UARTn_DTR register.
If this area is cleared, the UARTn_DTR initial value will be '0x80',
if this area is not cleared, the UARTn_DTR initial value will be '0x00'.
It needs to be careful when using.

11.6 Functional Description

The A33G52x has a built-in UART that supports single-sampling and multisampling modes when the start bit or data bit is detected by the UARTn_DTR <SMS> and UARTn_DTR <DMS> bits. The user can select the sampling method according to the operating environment. This sampling method supports noise-resistant UART communication environment.
In addition, in order to reduce the interference error caused by noise introduced into the received data of the UART, it is necessary to activate the Debounce function of the PMC (Port Map Control).

11.6.1 Receiver Single Sampling

On detecting a falling edge of input signal, UART recognizes the falling edge as a START bit signal. From this edge, every bits of a character transmission are sampled by 16-multiple clock signal.
If the SMS or DMS bit is '0' of UARTnDTR reigster, UART receiver will determine the bit value by the signal value on the seventh clock edge from a bit window.

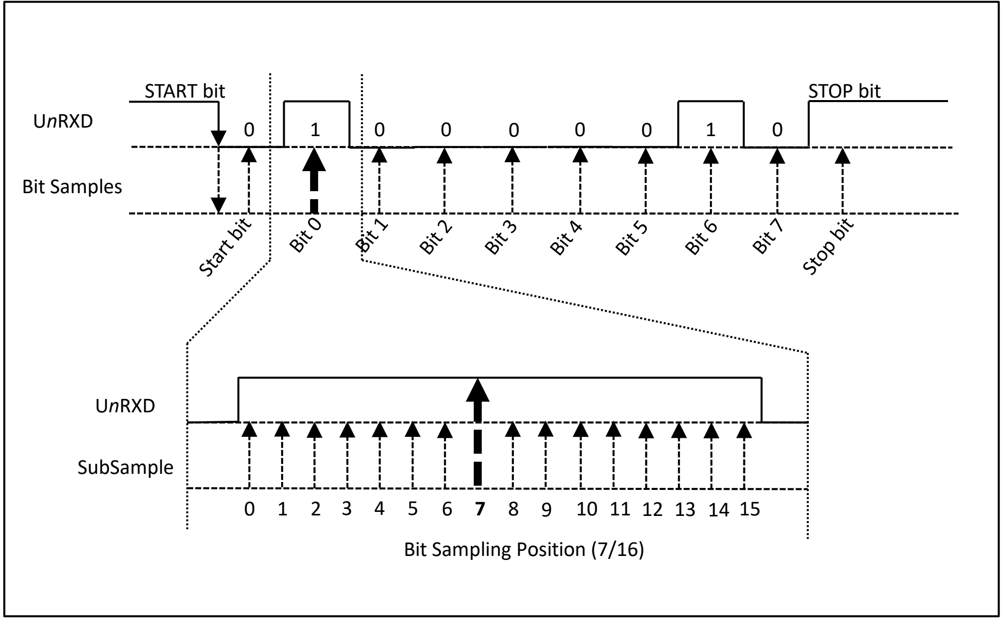
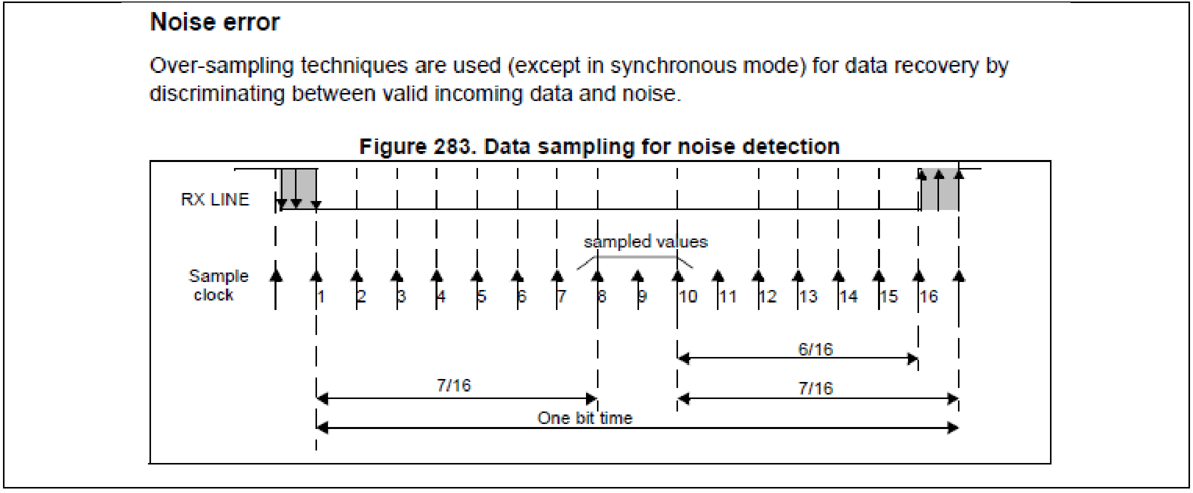


Figure 11.2. Sampling Method in 8-bit data receiving

11.6.2 Multi Sampling

If the SMS or DTS is '1', most occurrence value of signals on 7th, 8th, 9th clocks will be the bit value.



Noise error

Over-sampling techniques are used (except in synchronous mode) for data recovery by discriminating between valid incoming data and noise.

Figure 283. Data sampling for noise detection

Figure 11.3. The multi-sampling timimng of start bit and Rx data bit

11.6.3 Transmit data format

The transmitter has a data transmission function. The start bit, the data bit, the optional parity bit, and the stop bit are shifted continuously, and the least significant bit is moved first.
The number of data bits is selected in the DLEN [1: 0] field of the UARTn_LCR register.
The parity bit is set according to the PARITY and PEN bits filed in the UARTn_LCR register. For example, if the parity type is even, then the parity bit is determined by the bit sum of all the data bits. For odd parity, the parity bit is the opposite sum for all data bits. The number of stop bits is selected within the fileized STOPBIT in the UARTn_LCR register.
An example of the transmit data format is shown below.

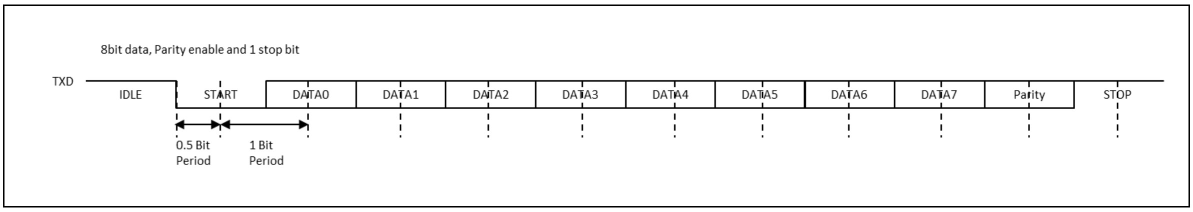


Figure 11.4. The example of transmit data format

11.6.4 Transmit Interrupt

The transfer operation produces some kind of interrupt flag. When the transmitter holding the register is empty, the THRE interrupt flag is set. When the transmitter shifter register is empty, the TXE interrupt flag is set. The user can choose which interrupt timing is best for the application.

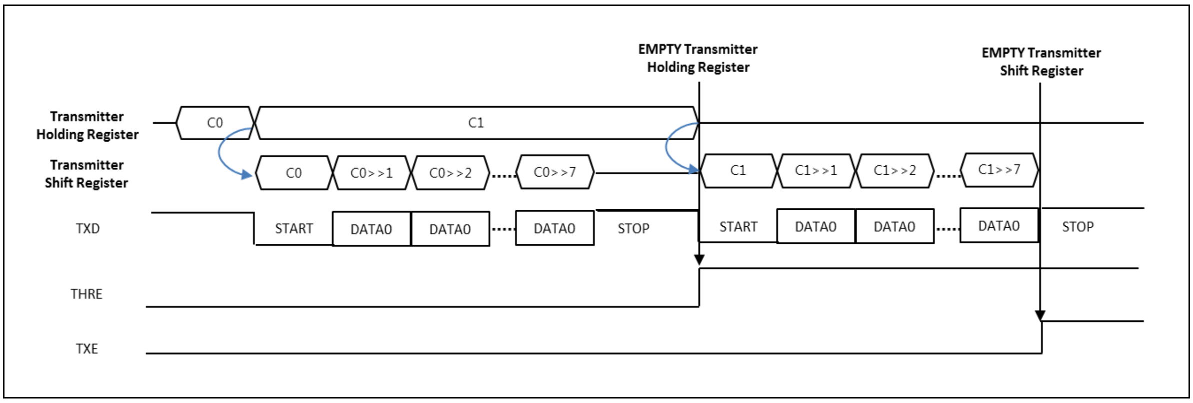

EMPTY Transmitter Holding Register
EMPTY Transmitter Shift Register
Transmitter Holding Register
C0
C1
Transmitter Shift Register
C0
C0>>1
C0>>2
C0>>7
C1
C1>>1
C1>>2
C1>>7
TXD
START
DATA0
DATA0
DATA0
STOP
START
DATA0
DATA0
DATA0
STOP
THRE
TXE

Figure 11.5. The diagram of transmit interrupt timing

11.6.5 Start Bit Detection

In the UART receiver block, there are two types of determination strategy of START bit.. One is by single-clock sampling and the other is by multi-clock sampling.

The single-clock sampling feature is the general sampling method, this feature acquire start bit value in the center of the bit period.

The multi-clock sampling feature has noise-rejection function for accurate communication against line noise. The timing diagrams of the single sampling and multi sampling features are shown as followings.

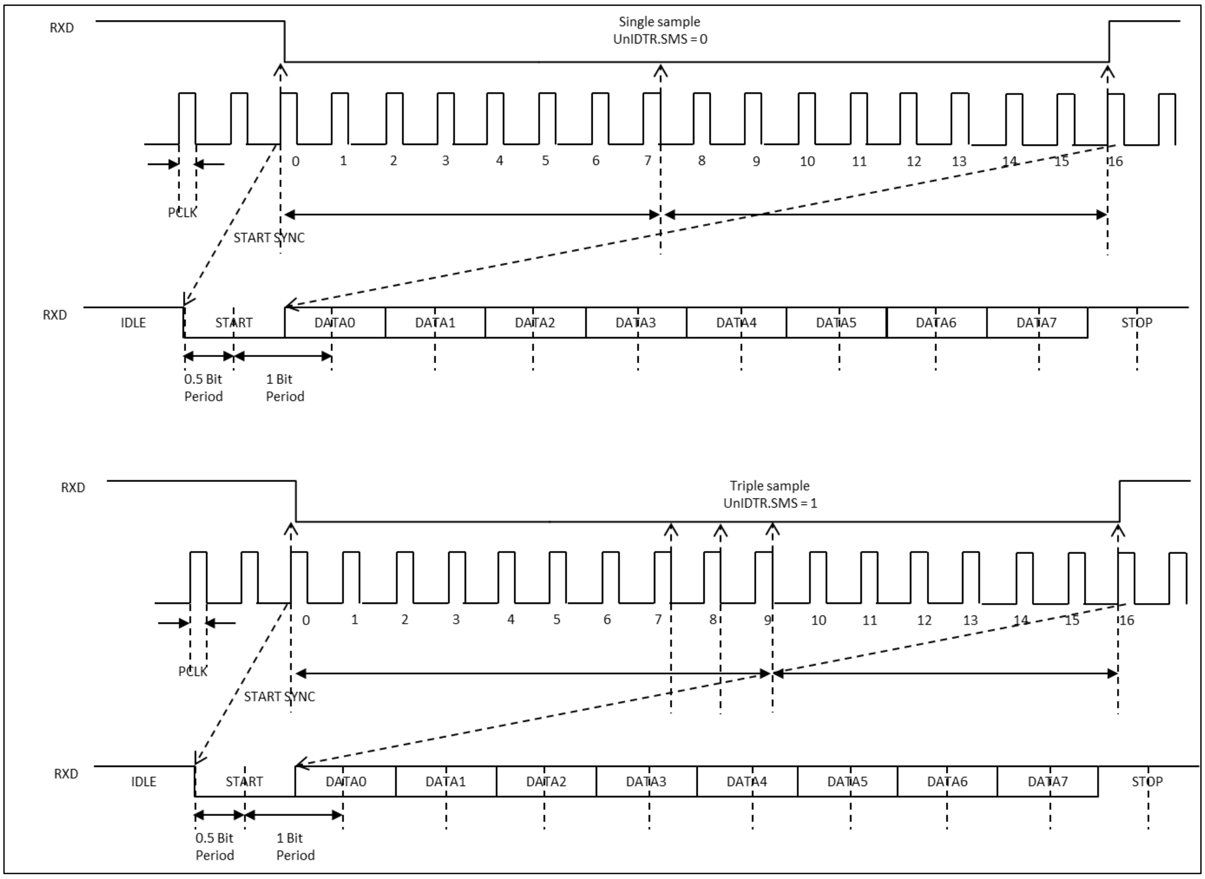


Figure 11.6. The timing diagram of the start bit detection in single sampling mode and multi sampling mode.

11.6.6 Data Sampling Strategy

In the receiver block, there are synchronous logics inside to prevent abnormal noise on the RxD input signal line. The start bit and data bit sampling type can be configured to any mode of single sampling or multi sampling. The start bit detection strategy can be selected by the SMS bit on UARTn_DTR register, the data bit detection strategy can be selected by the DMS bit on UARTn_DTR register.

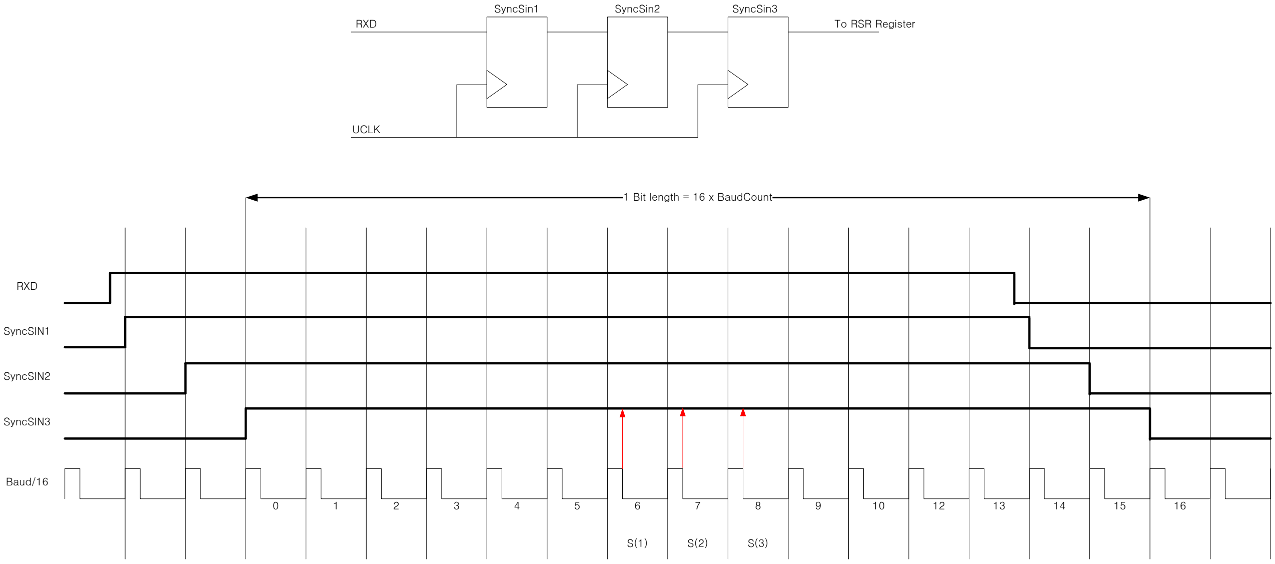


Figure 11.7. The timing diagram of data smapling

The figure 11.7 shows the timing diagram of a data sampling.
If the DMS bit was selected, the multi-sampling feature is activated and a bit value is to be determined by most occurrence value of the time of S(1), S(2), S(3) from the 8th clock in a bit period.
If the DMS bit was not selected, the single-sampling feature is activated and a bit value is tobe determined only by the value at S(2) timing. The DMS bit does not affect start bit strategy, the SMS has same feature with DMS, it only affect start bit detection.

11.6.7 Inter-frame Delay Transmittion

According to configuration of the inter-frame delay, the transmitter will insert idle timing between a successive character transmission. The period of idle between inter-frame transmission will be configured by the WAITVAL field on DTR register. If the WAITVAL field of UARTn_DTR was '0', there is no time delay on the transmission. On the contrary if this field was selected as '1', the line status of inter-frame transmission of every characters is set to be 'H' state during the configured timing.

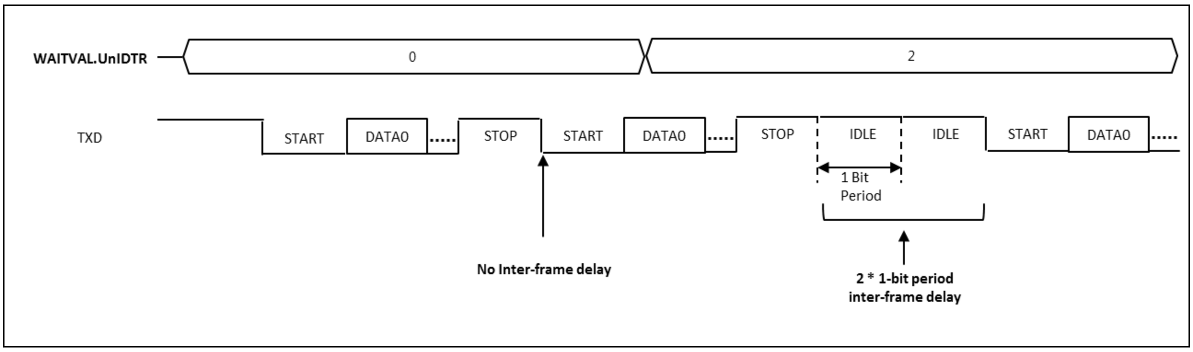


Figure 11.8. The timing diagram of inter-frame delay transmission

11.6.8 The Baud-rate Setting

Each channel of the UART has a programmable baud rate generator that can receive the PCLK clock and divide it by a divide value from 2 to 65535. (If dividing value is '1', it is the highest baud rate.). The output clock frequency of the baud rate generate is 16 x Baud rate. In order for UART communication to work properly, the UARTn_DLL and UARTn_DLM registers value must be written to appropriate values before UART operation.

The baud rate calculation formula is as follows:

$$\mathrm{BDR\ (UARTn\_DLM: UARTn\_DLL)} = \frac{PCLK/2}{16 * baud\ rate}$$

Table 11.7. Divisor value for each baud rate (PCLK=37.5MHz)

PCLK=37.5 MHz (HCLK/2)		
Baud Rate	**Decimal Divisor Value**	**Error Rate (%)**
1200	976	0.06%
2400	488	0.06%
4800	244	0.06%
9600	122	0.06%
19200	61	0.06%
38400	30	1.70%
57600	20	1.70%
115200	10	1.70%

Table 11.8. Divisor value for each baud rate (PCLK=8MHz)

PCLK=8 MHz (HCLK/2)		
Baud Rate	**Decimal Divisor Value**	**Error Rate (%)**
1200	208	0.16%
2400	104	0.16%
4800	52	0.16%
9600	26	0.16%
19200	13	0.16%
38400	X	X
57600	X	X
115200	X	X

Table 11.7 shows the baud rate generator baud rate and transmit/receive error rate at each baud rate for PCLK = 37.5 MHz, and the error rate is minimized at baud rates below 19200 bps. The frequency of the input clock affects the accuracy of the baud rate. It is not recommended to use '0' as the division value.

11.6.9 Setting Example

<Example 1> Initialization of UART0 channel (baudrate = 19200bps, PCLK = 8MHz)

PCMR<P8[17:16] = "01"	: Selects the function of PC8 pin as RXD0
PCMR<P9[19:18] = "01"	: Selects the function of PC9 pin as TXD0
PCCR<P8[17:16] = "10"	: Selects the direction of PC8(RXD0) pin as logic input
PCCR<P9[19:18] = "00"	: Selects the direction of PC9(TXD0) pin as push-pull output
UARTn_DTR< RXINV> = "0"	: Disables the inverted output of RXD0
UARTn_DTR< TXINV> = "0"	: Disables the inverted output of TXD0
UARTn_LCR<DLEN[0:1]> = "11"	: Set the data length to 8 bits.
UARTn_LCR<STOPBIT> = "0"	: Sets 1 stop bit
UARTn_LCR<PEN> = "0"	: Disables parity bit
UARTn_LCR<PARITY> = "0"	: Sets odd parity mode
UARTn_LCR<DLAB> = "0"	: Sets divisor latch access bit to normal transmission mode
UARTn_DLL <DLL[7:0]> = "00000000"	: Sets the value of divisor latch low byte to 0
UARTn_DLM< DLM[7:0]> = "00001101"	: Sets the value of divisor latch high byte to 13
UARTn_BFR<BFR[7:0]>= "0"	: Sets the value of divisor decimal point calculation register to 0
UARTn_FCR< FIFOEN> = "0"	: Disables transmit/receive FIFO type
UARTn_FCR< FCR1> = "0"	: Initializes the receive FIFO
UARTn_FCR< FCR2> = "0"	: Initializes the transmit FIFO
UARTn_FCR< FIFODEPTH > = "0"	: Sets the 1 byte receive FIFO trigger level
UARTn_SCR<SCR[7:0] ="00000000"	: Initialies the value of scratch byte data buffer
UARTn_RBR<RO[7:0]> READ	: Read the received byte data
UARTn_THR<THR[7:0]> READ	: Read the transmit byte data

CHAPTER 12. SPI (Serial Peripheral Interface)

Table 12.1. Operation Summary

Item	Description	Remark
Clock usage	PCLK	Transper rate
Reset Source	Set by PMU_PER	
Reset Generation	None	
Interrupt Generation	Tx Buffer Empty Rx Data Ready (SPI0(32), SPI1(33))	SPIn_SR
Interrupt Clear Method	Tx Buffer Empty: Write to SPn_TDR Rx Data Ready: Read from SPn_RDR	SPIn_TDR, SPIn_RDR

12.1 Overview

The A33G52x has two built-in Serial Peripheral Interface (SPI) channels. Serial Peripheral Interface (SPI) is serial communication synchronized by clock. It can support communication with multiple slaves in one master, and communication is done by selecting slave using SS line.

The A33G52x has two built-in Serial Peripheral Interface (SPI) channels. Serial Peripheral Interface (SPI) is serial communication synchronized by clock. It can support communication with multiple slaves in one master, and communication is done by selecting slave using SS line.

Serial Peripheral Interface (SPI) is a 3-wire or 4-wire synchronous transmission using four signal line like SS (Slave Select), SCK (SPI Clock), MOSI (Master-Out, Slave-Input), MISO (Master-Input, Slave-Output) and enables full-duplex communication. You can change the speed of the sending and receiving input clock. In addition, the transmit and receive buffers are configured in duplicate, allowing data to be read or written during transmission and reception.

Main Features of SPI (Serial Peripheral Interface)

- Master and Slave mode operation
- Full-duplex, 4-wired, 3-wired synchronous transmission
    - SS : Slave Select
    - SCLK : Serial Clock
    - MOSI : Maste- Output, Slave-Input
    - MISO : Master-Input, Slave-Output
- The control of SPI Clock Speed (SCK) and polarity
- Separated transmit and receive data register with 8,9,16,17-bit length
    - Selectable 8 bit, 9 bit, 16 bit, 17 bit SPI data size
- the transmission status check function
- Interrupt setting function by SS signal
- Loopback mode for internal verification
- Support boot program function using SPI0 channel when entering boot mode

12.2 Block Diagram

Below is a block diagram of the SPI (Serial Peripheral Interface).

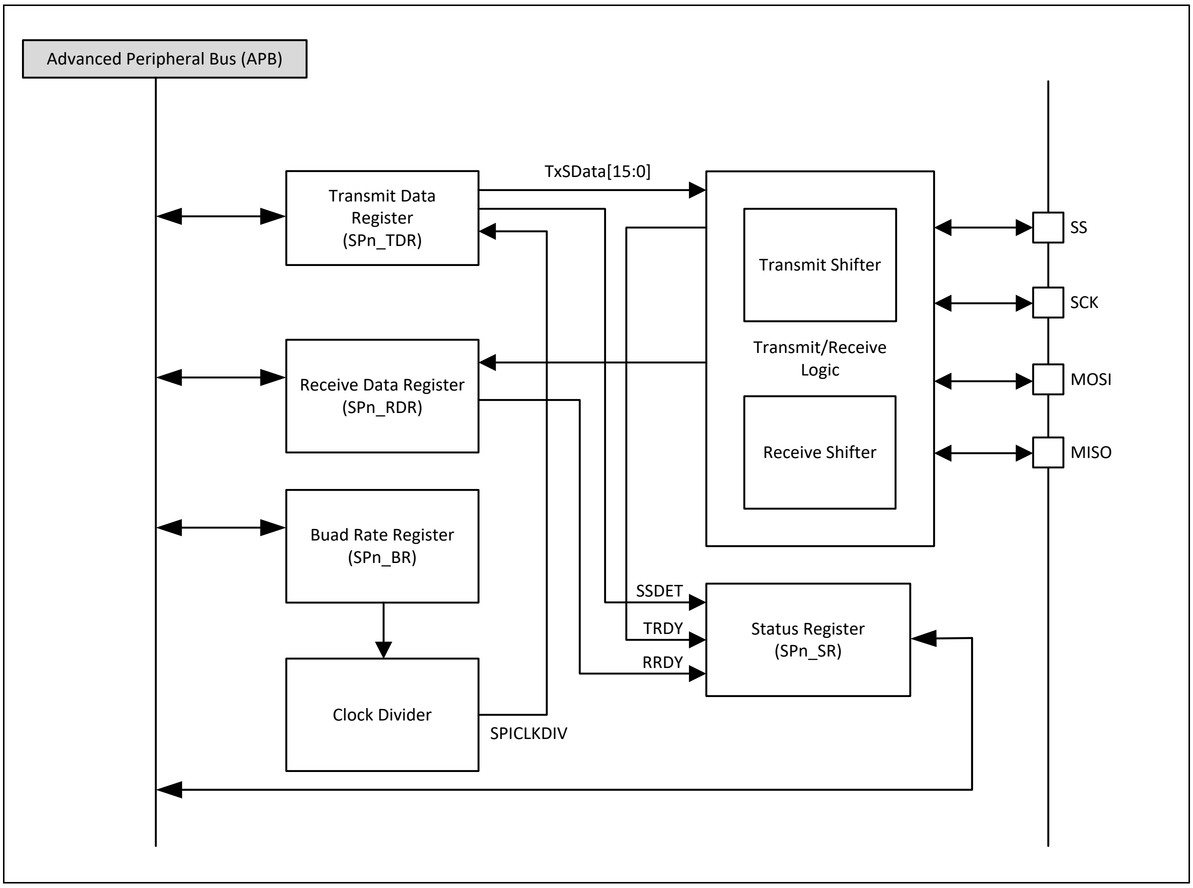


Figure 12.1. The block diagram of SPI

12.3 Pin Configuration

Table 12.2. The External Pins of SPIs

Pin Name	Type	Description	Support by package		
			A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM	A33G526MM A33G526ML A33G524MM A33G524ML	A33G527VQ A33G527VL A33G526VQ A33G526VQ
SS0	I/O	Slave Select signal of SPI0	O	O	O
SCK0	I/O	Serial Clock signal of SPI0	O	O	O
MOSI0	I/O	MOSI (Master-Out Slave-In) Data signal of SPI0	O	O	O
MISO0	I/O	MISO (Master-In Slave-Out) Data signal of SPI0	O	O	O
SS1	I/O	Slave Select signal of SPI1	O	O	O
SCK1	I/O	Serial Clock signal of SPI1	O	O	O
MOSI1	I/O	MOSI (Master-Out Slave-In) Data signal of SPI1	O	O	O
MISO1	I/O	MISO (Master-In Slave-Out) Data signal of SPI1	O	O	O

12.4 Register Map

The base address of the SPI is 0x4000_0800 and the address table for each channel is shown as follows.

Table 12.3. The address of each SPI channels

Port	Address
SPI0	0x4000_0800
SPI1	0x4000_0820

Table 12.4. The register list of SPI (Serial Peripheral Interface)

Register	Offset	Access Type	Description	Initial Value	Ref
SPIn_TDR	0x00	WO	SPI n Transmit Data Buffer Register	-	12.5.1
SPIn_RDR	0x00	RO	SPI n Receive Data Buffer Register	0x00000000	12.5.2
SPIn_CR	0x04	RW	SPI n Control Register	0x00000820	12.5.3
SPIn_SR	0x08	RW	SPI n Statue Register	0x00000002	12.5.4
SPIn_BR	0x0C	RW	SPI n Baudrate Register	0x000000FF	12.5.5
SPIn_EN	0x10	RW	SPI n Enable Register	0x00000000	12.5.6
SPIn_LR	0x14	RW	SPI n Timing Register	0x00001866	12.5.7

12.5 Register Description

12.5.1 SPIn_TDR SPI n Transmit Data Buffer Register

This register is a 17-bit data buffer that stores the transmit data to be output via the SPI channel.

SPI0_TDR=0x4000_0800, SPI1_TDR=0x4000_0820

31	30	29	28	27	26	25	24	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																
															-																
															WO																

16 0	**TDR**	Transmit data value

12.5.2 SPIn_RDR SPI n Receive Data Buffer Register

This register is a 17-bit data buffer that stores the receive data via the SPI channel.

SPI0_RDR=0x4000_0800, SPI1_RDR=0x4000_0820

31	30	29	28	27	26	25	24	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																
															-																
															WO																

16 0	**RDR**	Receive data value

12.5.3 SPIn_CR SPI n Control Register

This register provides various functions necessary for SPI communication such as sending/receiving buffer clearing and interrupt, SS signal control, communication setting function.

SPI0_CR=0x4000_0804, SPI1_CR=0x4000_0824

31	30	29	28	27	26	25	24	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	Reserved			SSCIE	TXIE	RXIE	SSMODE	SSOUT	LBE	SSMASK	SSMO	SSPOL	TEST	MS	MSBF	CPHA	CPOL	BITSEL	
-											0	0	-			0	0	0	0	1	0	0	0	0	0	1	0	0	0	00	
-											WO	WO	-			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

20	TXBC	TX BUFFER CLEAR	
		0	No operation
		1	TX BUFFER empty
19	RXBC	RX BUFFER CLEAR	
		0	No operation
		1	RX BUFFER empty
15	SSCIE	Select the edge interrupt of SSn signal	
		0	Disable SSn interrupt
		1	Enable the interrupt on the both edge (L→H, H→L) of SSn signal
14	TXIE	Select Tx data interrupt	
		0	Disable Tx data interrupt
		1	Enable Tx data interrupt
13	RXIE	Select Rx data interrupt	
		0	Disable Rx data interrupt
		1	Disable Rx data interrupt
12	SSMODE	SS automatic or manual control of SSn output.	
		0	Automatic control of SSn output
		1	Manual control of SSn output using SSOUT bit
11	SSOUT	Select SSn output level	
		0	SSn output 'L'
		1	SSn output 'H'
10	LBE	Select the Loop-back mode in Master mode	
		0	Normal communication mode
		1	Loop-back mode in Master mode
9	SSMASK	Select Ingnorance of the SSn input	
		0	Enable Rx only when the SSn signal is active (Masked SPI SCKn)
		1	Ignore SSn input (Always Rx enabled)
8	SSMO	Select SS signal output	
		0	Disable the output of SSn signal (SSMODE and SSOUT can not be output)
		1	Enable the output of SSn signal
7	SSPOL	Select the polarity of SSn signal.	
		0	Low active
		1	High active
6	TEST	Do not use (This bit is used only for internal testing purposes.)	
5	MS	Select SPI Master/Slave mode	
		0	Slave mode
		1	Master mode
4	MSBF	Select the SPI serial data type (MSB/LSB)	
		0	LSB first
		1	MSB first
3	CPHA	Select the clock phase	
		0	Output data at start phase

		1	Output data at next phase
2	CPOL	Select the polarity of SPI clock for data sample	
		0	Start clock at ‘L’
		1	Start clock at ‘H’
1 0	BITSEL	Select the size of transmit/receive data bit	
		00	8-bit
		01	9-bit
		10	16-bit
		11	17-bit

12.5.4 SPIn_SR SPI n Status Register

This register provides information on the SPI interrupt flag and SPI channel communication status.

SPI0_SR=0x4000_0808, SPI1_SR=0x4000_0828

31	30	29	28	27	26	25	24	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																									SSDET	SSON	OVRF	UDRF	SBUSY	TRDY	RRDY
-																									0	0	0	0	0	1	0
-																									RW	RW	RW	RW	RO	RO	RO

6	**SSDET**	SS negate detection flag after SSn enable	
		0	SS negate not detected
		1	SS negate detected (to clear this flag, write ‘0’ this bit)
5	**SSON**	The current status of SS input flag	
		0	Disabled SS signal
		1	Endabled SS signal
4	**OVRF**	Rx data overrun error flag	
		0	NOT occurred Rx overrun error
		1	Occurred Rx overrun error (To clear this flag, write ‘0’ or read out Rx buffer)
3	**UDRF**	Tx data underrun error flag	
		0	NOT occurred Tx underrun error
		1	Occurred Tx underrun error (To clear this flag, write’0’ or write to data buffer)
2	**SBUSY**	Busy flag	
		0	Idle status
		1	Busy status
1	**TRDY**	Tx buffer empty flag	
		0	Tx buffer is busy
		1	Tx buffer ready – completed data transmission (To clear this flag, write to data buffer)
0	**RRDY**	buffer ready flag	
		0	No received data
		1	Data received (To clear this flag, read the data buffer)

12.5.5 SPIn_BR SPI n Baudrate Register

This register sets the SPI transmit / receive transfer rate. You can set the SPI data transfer rate (SCK) based on the PCLK input clock.

SPI0_BR=0x4000_080C, SPI1_BR=0x4000_082C

31	30	29	28	27	26	25	24	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							
-																								11111111							
-																								RW							

7 0	BR	SPI Data transfer rate of SCK. The BR value must be more than 1. $$Baud\ Rate = \frac{PCLK}{BR+1}\ (\text{However BR} \geq 1)$$

12.5.6 SPIn_EN SPI n Enable Register

This register is used to initialize the SPI transmit/receive status.

SPI0_EN=0x4000_0810, SPI1_EN=0x4000_0830

31	30	29	28	27	26	25	24	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	When '1' is written to this bit, the data in the SPI transmit buffer is shifted to the transmit shifter. (When sending the first data, the ENABLE bit must be initialized.) If the ENABLE bit in the SPIn_EN register is set to '1', the value set to '1' in the UDRF, OVRF, and SSDET bits in the SPIn_SR register is initialized to '0' by setting the ENABLE bit in the SPIn_EN register to '0'. 1) When the 3rd bit of SPIn_SR, UDRF, is set to '1' and a transmit underrun error occurs. 2) When OVRF, the fourth bit of SPIn_SR, is set to '1' and a receive overrun error occurs 3) When 7th bit of SPIn_SR, SSDET, is set to '1' and SS negate is detected. The values of the remaining bits except the three flag bits are retained.

12.5.7 SPIn_LR SPI n Timing Register

This register sets the delay timing of the SPI signal. It operates based on the SCK clock set in the SPIn_BR register.

SPI0_LR=0x4000_0814, SPI1_LR=0x4000_0834

31 30 29 28 27 26 25 24 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**	**STL**
-	**00110**	**00011**	**00110**
-	**RW**	**RW**	**RW**

14 **10**	**SPL**	Set Stop delay time	
		n	Stop delay time = n * (1/SCK)
9 **5**	**BTL**	Set the packet interval for continuous output	
		n	Burst delay time = n * (1/SCK)
4 **0**	**STL**	Set Start delay time	
		n	Start delay time = n * (1/SCK)

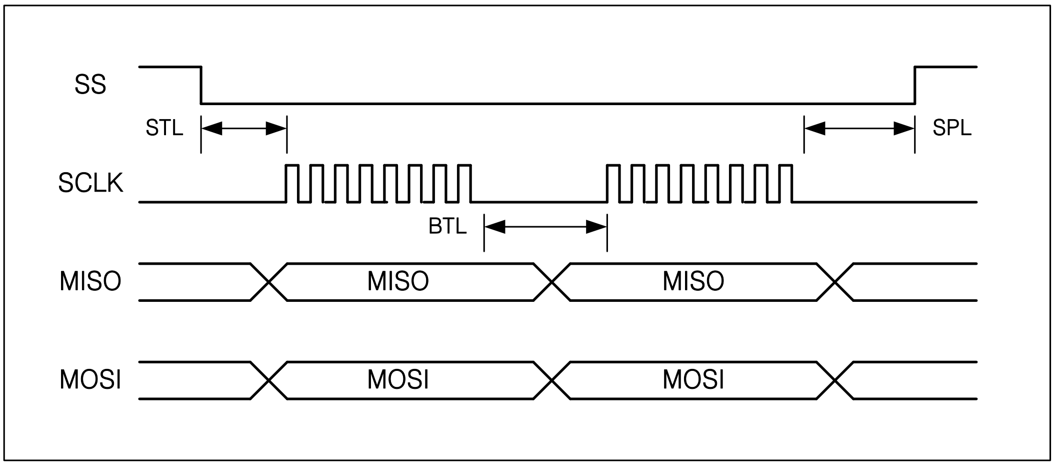


Figure 12.2. The SPI delay timing

12.6 Functional Description

12.6.1 SPI Bus Timing

The sender and receiver of the Serial Peripheral Interface (SPI) share the same clock but are independent of each other. Full duplex transmission and reception is therefore possible. The transmitting and receiving ends have a double buffer structure, so that during the reception of the subsequent data, the RDR data received before is read or the subsequent data is written to the TDR while the data is being transmitted, so that a back-to-back transfer is possible.

The operation timing according to the CPHA bit and CPOL bit of the SPI (Serial Peripheral Interface) control register SPIn_CR is as follows.

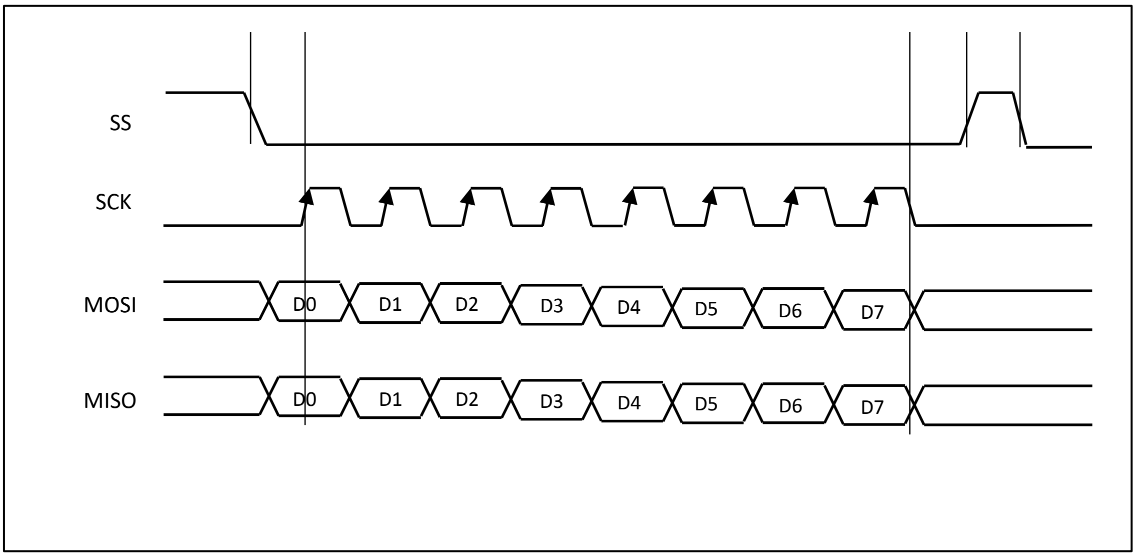


Figure 12.3. SPI Timing diagram of SPI (CPHA=0, CPOL=0, MSBF=0)

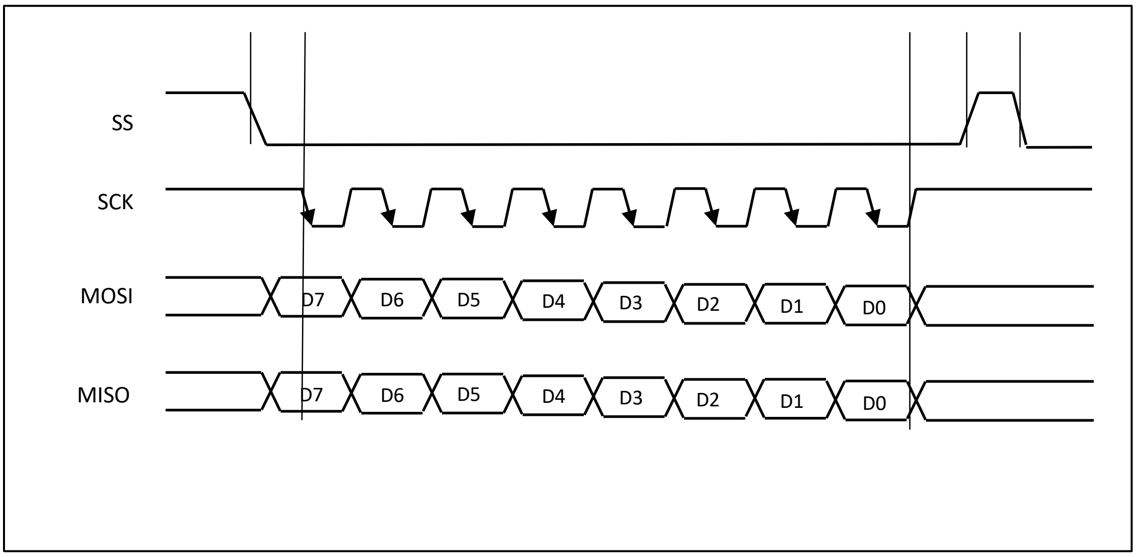


Figure 12.4. Timing diagram of SPI (CPHA=0, CPOL=1, MSBF=1)

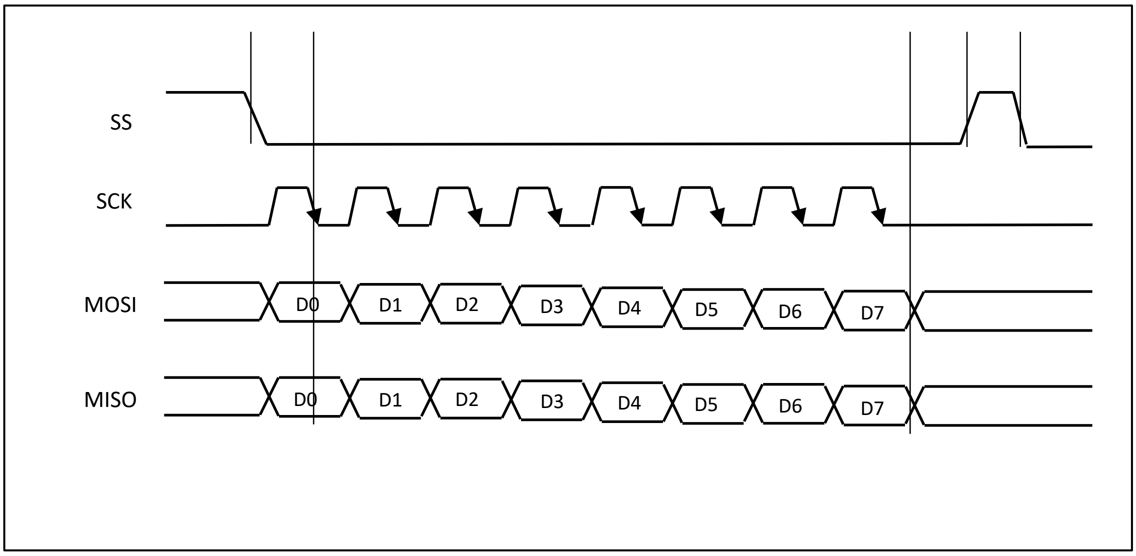


Figure 12.5. Timing diagram of SPI (CPHA=1, CPOL=0, MSBF=0)

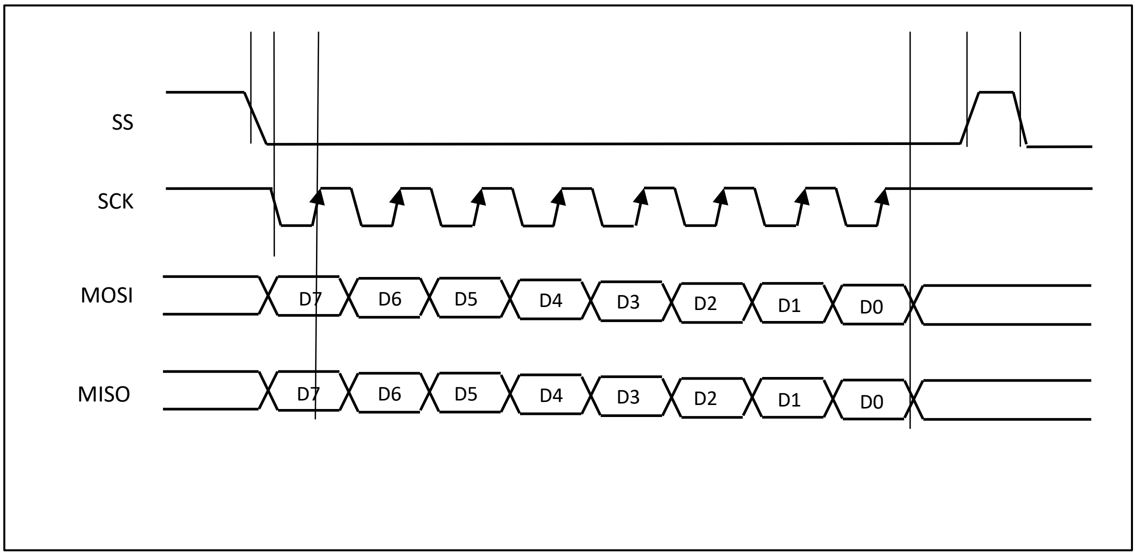


Figure 12.6. Timing diagram of SPI (CPHA=1, CPOL=1, MSBF=1)

12.6.2 Setting Example

<Example 1> SPI0 Master Initial Setting Example

PB_MR<21:20> = “01”	: Select the function of PB10 pin as SS0
PB_MR<23:22> = “01”	: Select the function of PB11 pin as SCK0
PB_MR<25:24> = “01”	: Select the function of PB12 pin as MOSI0
PB_MR<27:26> = “01”	: Select the function of PB13 as MISO0
PB_CR<21:20> = “00”	: Set the direction of PB10 (SS0) pin as push-pull output
PB_CR<23:22> = “00”	: Set the direction of PB11 (SCK0) pin as push-pull output
PB_CR<25:24> = “00”	: Set the direction of PB12 (MOSI0) pin as push-pull output
PB_CR<27:26> = “10”	: Set the direction of PB13 (MISO0) pin as logic input
PB_PCR<10> = “0”	: Disable pull-up/pull-down resistor of PB10 (SS0) pin
PB_PCR<11> = “0”	: Disable pull-up/pull-down resistor of PB11 (SCK0) pin
PB_PCR<12> = “0”	: Disable pull-up/pull-down resistor of PB12 (MOSI0) pin
PB_PCR<13> = “0”	: Disable pull-up/pull-down resistor of PB13 (MISO0) pin
SPI0_CR<BITSEL[1:0]> = “00”	: Select the size of transmit/receive data as 8-bit
SPI0_CR<MSBF> = “1”	: Select the SPI0 serial data type as MSB-first
SPI0_CR<MS> =”1”	: Select SPI0 channel as master mode
SPI0_CR<SSPOL> = “1”	: Select the start clock polarity on the SS0 pin
SPI0_CR<CPHA> = “1”	: Set the clock phase to output data at the start phase
SPI0_CR<CPOL> = “1”	: Set the start clock of SPI0 serial for data sample to high
SPI0_BR<BR[7:0]> = “0100 1111”	: Select SPI0 transfer rate as 1MHz
SPI0_EN<ENABLE> = “1”	: Reset the transmit buffer when transmitting the first data.

<Example 2> SPI1 Slave Initial Setting Example

PD_MR<17:16> = “01”	: Select the function of PD8 pin as SS1
PD_MR<19:18> = “01”	: Select the function of PD9 pin as SCK1
PD_MR<21:20> = “01”	: Select the function of PD10 pin as MOSI1
PD_MR<23:26> = “01”	: Select the function of PD11 pin as MISO1
PD_CR<17:16> > = “10”	: Select the direction of PD8 (SS1) pin as logic input
PD_CR<19:18> = “10”	: Select the direction of PD9 (SCK1) pin as logic input
PD_CR<21:20> = “10”	: Select the direction of PD10 (MOSI1) pin as logic input
PD_CR<23:26> > = “00”	: Select the direction of PD11 (MISO1) pin as logic input
PD_PCR<8> = “0”	: Disable pull-up/pull-down resistor of PD8 (SS1) pin
PD_PCR<9> = “0”	: Disable pull-up/pull-down resistor of PD9 (SCK1) pin
PD_PCR<10> = “0”	: Disable pull-up/pull-down resistor of PD10 (MOSI1) pin
PD_PCR<11> = “0”	: Disable pull-up/pull-down resistor of PD11 (MISO1) pin
SPI1_CR<BITSEL[1:0]> = “00”	: Select the size of transmit/receive data as 8-bit
SPI1_CR<MSBF> = “1”	: Select the SPI0 serial data type as MSB-first
SPI1_CR<MS> =”0”	: Select SPI0 channel as master mode
SPI1_CR<SSPOL> = “1”	: Select the start clock polarity of SS1 pin as 'H'.
SPI1_CR<CPHA> = “1”	: Set the clock phase to output data at the start phase
SPI1_CR<CPOL> = “1”	: Set the start clock of SPI1 serial for data sample to high
SPI1_BR<BR[7:0]> = “0100 1111”	: Select SPI1 transfer rate as 1MHz
SPI1_EN<ENABLE> = “1”	: Reset the transmit buffer when transmitting the first data.

<Example 3> Data Output of SPI0 Master Example

SPI0_CR<SSOUT> = “0”	: Select the start clock polarity of SS1 pin as 'L'.
SPI0_SR<TRDY> READ	: Read SPI0 transmit/receive buffer - Check if transfer is ready
SPI0_EN<ENABLE> =”0”	: Disable SPI0 to insert transfer data into the SPI0 buffer
SPI0_TDR<16:0> = “VALUE”	: Input transfer value to SPI0 transmit data register.
SPI0_En<ENABLE> = “1”	: Enable SPI0 master channel and start to output data
SPI0_SR<TRDY> READ	: Read SPI0 transmit/receive buffer - Check if receiving is ready
SPIn_CR<SSOUT> = “1”	: Select the start clock polarity of SS0 pin as 'H'.
SPI0_EN<ENABLE> =”0”	:. Disable SPI0 master channel

<Example 4> Data Output of SPI1 Slave Example

SPI1_En<ENABLE> = “1”	: Enable SPI1 Slave channel
SPI1_SR<RRDY> READ	: Check whether the SPI1 channel is receiving data
SPI1_RDR<RDR> READ	: Read the value of the SPI1 channel's receive data register

CHAPTER 13. 12-bit ADC

Table 13.1. Operation Summary

Item	Description	Remark
Clock usage	PCLK Timer 0~7	Conversion speed Start-of-Conversion
Reset Source	Set by PMU_PER Set by ADCEN/ADSTBY	ADC_MR
Reset Generation	None	
Interrupt Generation	End-of-Conversion (ADC(43))	ADC_MR
Interrupt Clear Method	Write ‘1’ to ADIF	ADC_CR

13.1 Overview

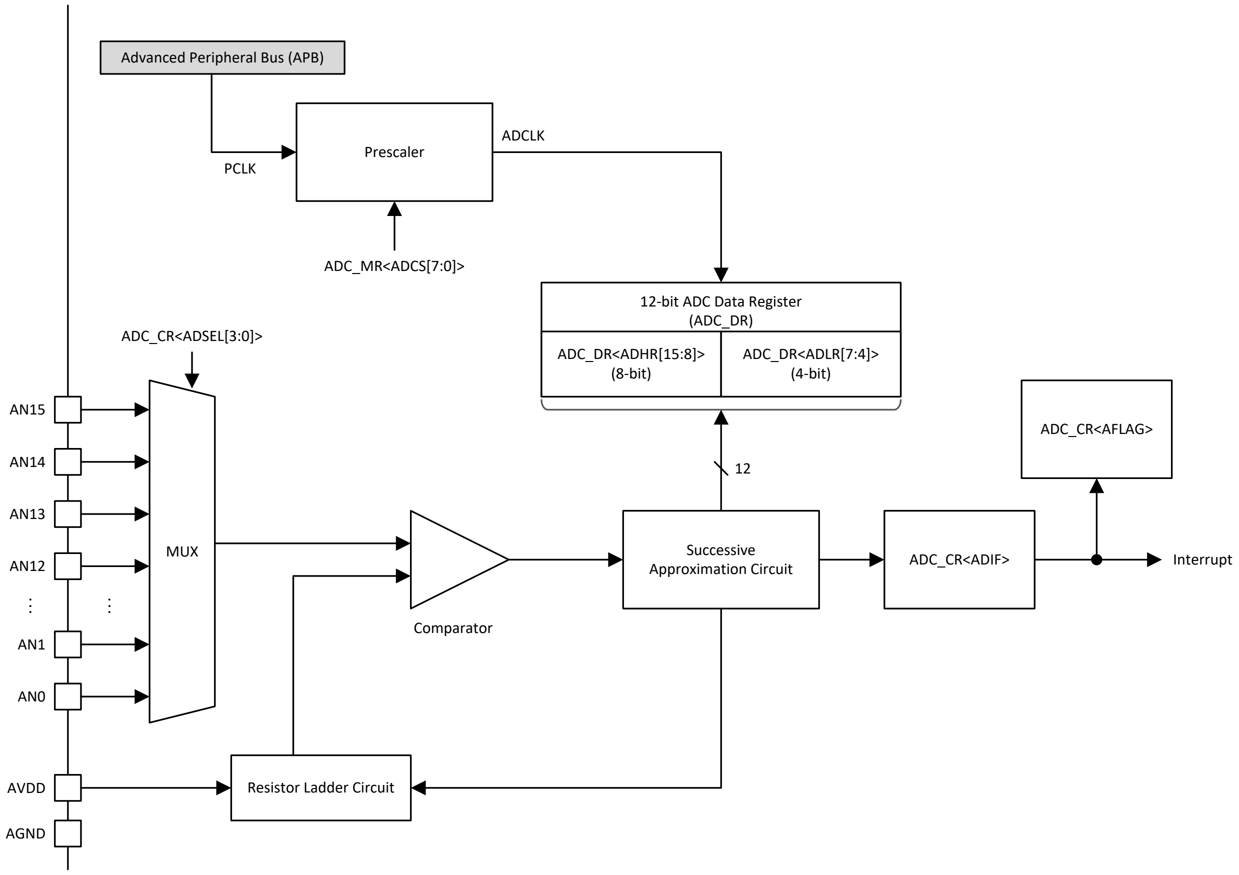
The A33G52x has a built-in 12-bit SAR type analog-to-digital converter (ADC) 1-unit and samples the analog signal as a digital signal at 70ksps speed. The ADC (Analog to Digital Converter) built in the A33G52x is designed to receive up to 16 channels in time-divisional manner.

When converting an analog signal to a digital signal, convert the analog value input to the ADC (Analog to Digital Converter) channel to a 12-bit resolution digital value using the signal applied to the AVDD pin (analog voltage supply pin) as the reference voltage . In other words, the input reference voltage of the ADC can be set by varying the voltage of the AVDD pin, and the analog voltage ranging from GND to AVDD can be input and output as a digital signal.

Main Features of 12-bit ADC (Analog Digital Converter)

- 12-bit resolution
- Up to 16 multiplexed input channels
  - 100-pin : 16 channels
  - 80-pin : 16 channels
  - 64-pin : 10 channels
- AVDD pin can be used to set the ADC input reference voltage
- Analog input voltage range: GND to AVDD
- Conversion Time: 15us per channel (at 4MHz ADC clock)
- A / D conversion end interrupt flag support
- External ADC start trigger signal input selection
  - A/D conversion trigger source through timer match function

13.2 Block Diagram



Note 1) The number of ADC input ports depends on the package type. Read the pin configuration of the ADC (CH13.3)

Note 2) The AGND port is not supported in the 64-pin package type and is connected to GND inside the chip. (Figure 16.14)

Figure 13.1. ADC block diagram

13.3 Pin Configuration

Following table shows the list of A/D converter input pins. AVDD and AVSS are the power pins to A/D converter, and AVDD is the reference voltage of A/D converter.

Table 13.2. The pin list of A/D converter

Pin Name	I/O	Description	Support by package		
			A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM	A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM	A33G527RL A33G526RL A33G526RM A33G524RL A33G524RM
AVDD*	Power	Analog Power Input	O	O	O
AGND**	Power	Analog GND	Connected to GND	O	O
AN0	Input	Analog Input Channel 0	O	O	O
AN1	Input	Analog Input Channel 1	O	O	O
AN2	Input	Analog Input Channel 2	O	O	O
AN3	Input	Analog Input Channel 3	O	O	O
AN4	Input	Analog Input Channel 4	O	O	O
AN5	Input	Analog Input Channel 5	O	O	O
AN6	Input	Analog Input Channel 6	O	O	O
AN7	Input	Analog Input Channel 7	O	O	O
AN8	Input	Analog Input Channel 8	-	O	O
AN9	Input	Analog Input Channel 9	-	O	O
AN10	Input	Analog Input Channel 10	-	O	O
AN11	Input	Analog Input Channel 11	-	O	O
AN12	Input	Analog Input Channel 12	-	O	O
AN13	Input	Analog Input Channel 13	-	-	O
AN14	Input	Analog Input Channel 14	O	O	O
AN15	Input	Analog Input Channel 15	O	O	O

Note) * Analog VDD volatage is equal to or lower than VDD voltage.
**AVDD, AGND power configuration: See 16.3.8.

13.4 Register Map

The following registers are provided for controlling the A/D converter. The base address of these registers is 0x4000_0E00.

Table 13.3. The register list of A/D Converter

Register	Offset	Access Type	Description	Initial Value	Ref
ADC_CR	0x00	RW	ADC Control Register	0x00000100	13.5.1
ADC_MR	0x04	RW	ADC Mode Register	0x000040FF	13.5.2
ADC_DR	0x08	RO	ADC Data Register	0x00000000	13.5.3

13.5 Register Description

13.5.1 ADC_CR ADC Control Register

This register has the function to control the operation of the ADC and to check the interrupt and status flags.

ADC_CR=0x4000_0E00

31	30	29	28	27	26	25	24	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																							ADEOC	ADST	AFLAG	ADIF	Reserved	ADSEL			
-																							0	0	0	0	--	0000			
-																							RO	WO	RO	RC	-	RW			

8	ADEOC	A/D conversion status (Read/Write)	
		0	No conversion or converting state
		1	End-of-Conversion
7	ADST	Start/Stop the A/D conversion	
		0	Stop A/D Conversion
		1	Start A/D Conversion
6	AFLAG	The status flag of A/D conversion (Read-only)	
		0	Conversion Busy
		1	Conversion Idle
5	ADIF	A/D conversion	
		0	No A/D interrupt
		1	Occurred A/D interrupt (Write '1' to clear this bit)
3 0	ADSEL	Select ADC input channel	
		ANn input select (n = 0 ~ 15)	

13.5.2 ADC_MR ADC Mode Register

This register provides settings for ADC operating clock and operating mode, and interrupt.

ADC_MR=0x4000_0E04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rserved																ADCEN	ADSTBY	Reserved	ADIE	EXTRG	TSEL			ADCS							
-																0	1	-	0	0	000			11111111							
-																RW	RW	-	RW	RW	RW			RW							

15	ADCEN	Enable/Disable A/D converter	
		0	Disable A/D converter
		1	Enable A/D converter
14	ADSTBY	A/D converter stand-by mode	
		0	ADC normal operation
		1	ADC stand-by mode
13	Reserved	Always '0'	
12	ADIE	Enable/Disable end-of-A/D conversion interrupt	
		0	Disable ADC interrupt
		1	Enable ADC interrupt
11	EXTRG	Select external trigger condition for start conversion	
		0	Trigger by ADST bit
		1	Trigger by external trigger source
10 8	TSEL	A/D conversion trigger source	
		000	Timer 0 match
		001	Timer 1 match
		010	Timer 2 match
		011	Timer 3 match
		100	Timer 4 match
		101	Timer 5 match
		110	Timer 6 match
		111	Timer 7 match
7 0	ADCS	Select A/D conversion clock $fADC = \frac{PCLK}{ADCS + 1}$ The ADC clock should be under 4MHz (in case of VDD=5V)	

13.5.3 ADC_DR A/D Data Register

This register is a 12-bit read-only register that stores the converted data values from the ADC.

ADC_DR=0x4000_0E08

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="8">ADHR</td><td colspan="4">ADLR</td><td colspan="4">Reserved</td></tr>
<tr><td colspan="16">-</td><td colspan="8">00000000</td><td colspan="4">0000</td><td colspan="4">-</td></tr>
<tr><td colspan="16">-</td><td colspan="8">RO</td><td colspan="4">RO</td><td colspan="4">-</td></tr>
</table>

15 8	ADHR	Upper byte of A/D conversion data
7 4	ADST	Lower byte of A/D conversion data

13.6 Functional Description

13.6.1 A/D Conversion Timing

The 12-bit ADC built into the A33G52x operates at the following timings.

After start-of-A/D conversion, 10 clocks for stabilization, MSB → LSB conversion, and 1 bit conversion takes 4 clocks. Therefore, the total conversion time is 10 clock + (12-bit data * 4 clock) = total 58 clocks.

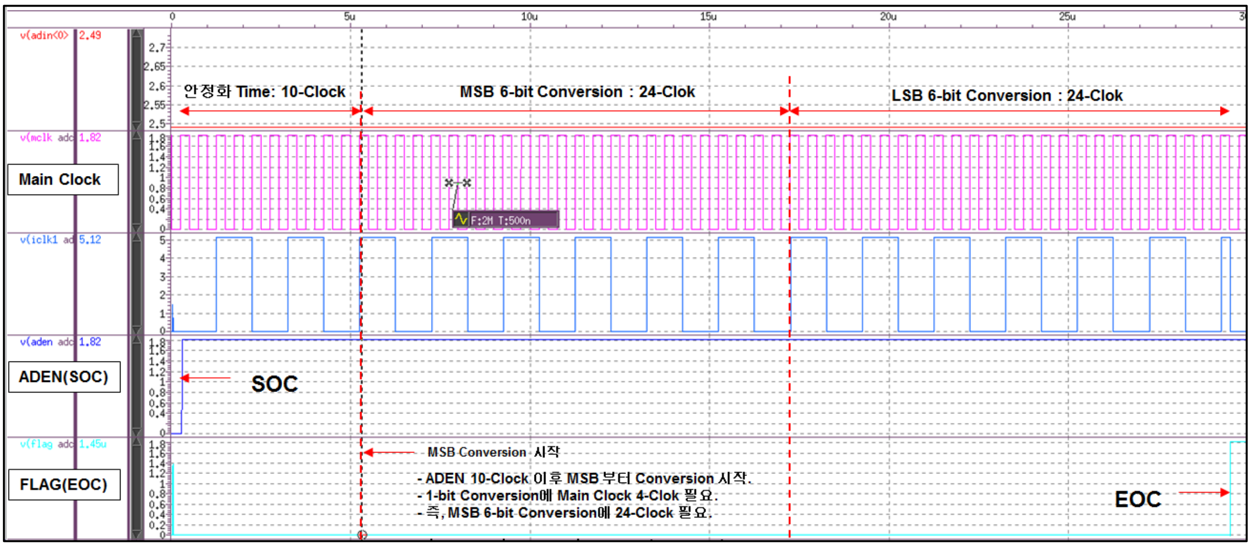

0
5u
10u
15u
20u
25u
v(adin<0> 2.49
v(mclk adc 1.82
v(iclk1 ad 5.12
v(aden adc 1.82
v(flag adc 1.45u
안정화 Time: 10-Clock
MSB 6-bit Conversion : 24-Clok
LSB 6-bit Conversion : 24-Clok
Main Clock
F:2M T:500n
ADEN(SOC)
SOC
FLAG(EOC)
MSB Conversion 시작
- ADEN 10-Clock 이후 MSB 부터 Conversion 시작.
- 1-bit Conversion에 Main Clock 4-Clok 필요.
- 즉, MSB 6-bit Conversion에 24-Clock 필요.
EOC

Figure 13.2. A/D conversion timing

13.6.2 ADC Operating Conditions

The ADC of the A33G52x MCU should be used by setting the A/D conversion time according to the reference input voltage AVDD. The following table specifies the ADC operating conditions.

Table 13.4 Operating condition of A/D conversion

AVDD Voltage (V)	ADC Clock (MHz)	A/D Conversion Time (μs)
2.4	0.25	240
2.7	1.00	60
3.0	2.00	30
4.0	3.00	20
5.0	4.00	15

13.6.3 The Resolution of 12-bit ADC

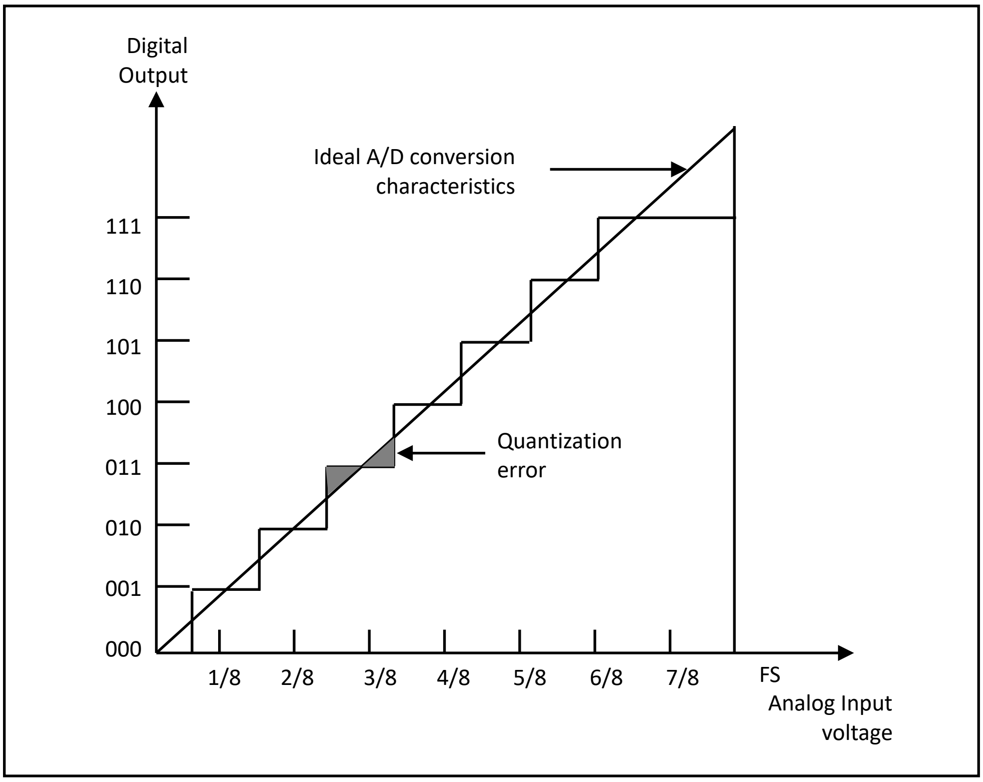


Figure 13.3. The definition of A/D conversion resolution (1)

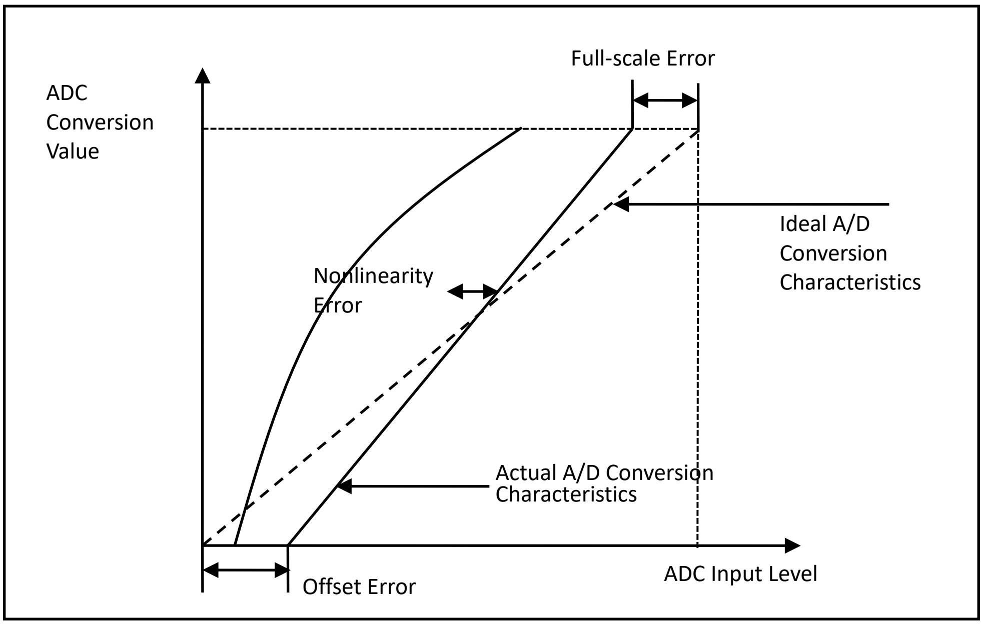


Figure 13.4. The definition of A/D conversion resolution (2)

13.6.4 Setting Example

<Example 1> Setting ADC Interrupt operation (PCLK = 74MHz)

PA_MR<PA0[1:0]> = “11”	: Selects the function of PA0 pin as AN0 (ADC Input)
PA_CR<P0[1:0]> = “11”	: Selects the direction of PA0 pin as analog input
PA_PCR<P0> = “0”	: Disables pull-up/pull-down resistor of PA0(AN0) pin
PMU_PER<ADC> = “1”	: Sets ADC peripheral enable
PMU_PCCR<ADC> = “1”	: Enables ADC input clock
ADC_MR<ADCEN> = “1”	: Enables ADC
ADC_CR<ADIF> = “1”	: Clears the ADC interrupt flag
ADC_MR<ADSTBY> = “0”	: Sets ADC normal operation
ADC_MR<EXTRG> = “0”	: Selects the ADC trigger with ADST bit
ADC_MR<ADCS[7:0]> = “00010010”	: Input value 18 for A/D conversion clock setting. f_{ADC} = PCLK/(ADCS+1) = 74MHz/(18+1) = 3.75MHz
ADC_MR<ADIE> = “1”	: Enables ADC interrupt
ADC_CR<ADSEL> = “0”	: Selects ADC conversion input channel as AN0.
ADC_CR<ADST> = “1”	: Starts A/D conversion

13.7 Notice

The user should be observe the following notice to ensure that the correct operation of A/D converter.

1. Analog input voltage range : During A/D conversion, voltage level of analog input pin have to be in Vss ≤ ANn ≤ AVDD

2. Analog, Digital voltage:
   Even if ADC is not being used, VDD and GND must not be unconnected.

3. AVREF conversion range : Reference voltage, AVREF is fixed as AVDD.

4. Note for board design : Separates digital circuit from analog circuit in board layout.
   Especially, it needs to avoid the layout that digital signal line crosses analog signal line or goes closely around analog signal line; otherwise, it can cause an incorrect operation of analog circuit and it can affect to the accuracy of A/D conversion.

5. Note about noise : In order to protect circuit from surge voltage or abnormal voltage in AVREF or analog input pins, make a protection circuit between AVDD and AGND like figure 14.5. Bypass condensers connected to AVDD and filter condenser connected to analog input pin should be connected to AGND.

6. Influence to absolute accuracy : Attaching external condenser can cause a coupling or a ground connection, and then if there is a noise on ground connection line, it can decrease an absolute accuracy. This condenser should be connected to stable ground like the AGND.
   If filter circuit is used, take into consideration on inference with digital signal at the same board, and check whether the circuit operates like an antenna.

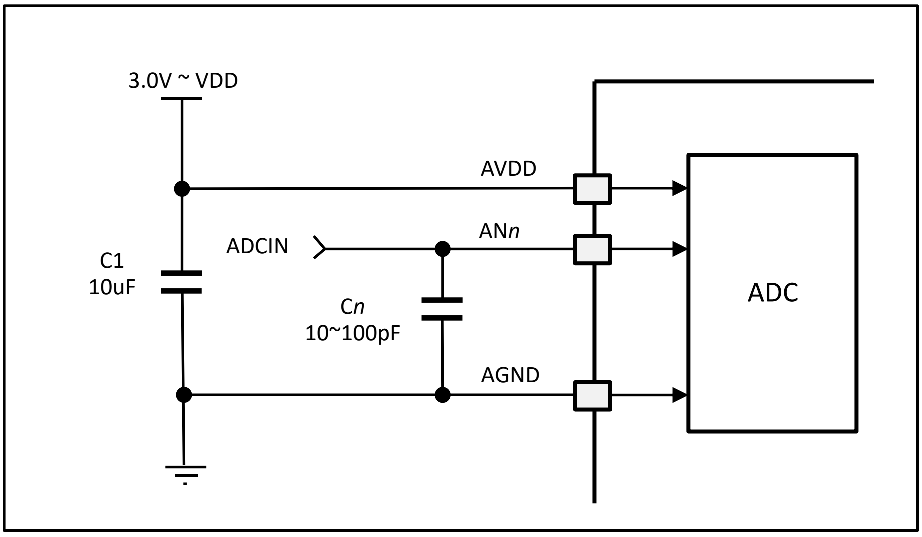


Figure 13.5. A example of ADC circuit

CHAPTER 14. FMC (Flash Memory Controller)

Table 14.1. Operation Summary

Item	Description	Remark
Clock usage	HCLK	Access speed
Reset Source	None	
Reset Generation	None	
Interrupt Generation	None	
Interrupt Clear Method	None	

14.1 Overview

The A33G52x MCU contains a non-volatile memory, code flash and data flash, which are electrically able to erase, program, each with a 1KB sector size. A33G52x series offers a high-capacity code flash, you can use the flash area according to the purpose of user application.

Code and data flash memory of A33G52x also supports CRC16 (Cyclic Redundancy Check 16) function, which enables error detection on data written to flash memory.

When the Extended mode is activated by setting the EX bit value in the FMC_TEST register, functions such as Self-Program, Self-Erase, 512-byte Page Erase, and Boot Block Protection are also available.

In addition, it supports limitations on read/program/erase for specific areas of memory, providing security features such as flash memory protection, external access protection, and security.

Main Features of FMC (Flash Memory Controller)

- Code Flash Capacity : Built-in high capacity code flash memory
  - A33G527 (384KB, 384 Sectors)
  - A33G526 (256KB, 256 Sectors)
  - A33G524 (128KB 128 Sectors)
- Data Flash Capacity : 32KB (32 Sectors)
- Up to 25 MHz Flash Access Timing
- Program size unit
  - Code Flash : 1-word (4 Bytes)
  - Data Flash : 1-byte
- 512 Bytes or 1KB sector ERASE
- Flash Self-PROGRAM of code flash memory
  - Supports to update data in some code flash memory region during execution of user program in code flash area.
- CRC16 generation and verification for error detection
- The restrict a specific area of flash memory for protection and security.
- Endurance
  - Code Flash 10,000 cycles
  - Data Flash 100,000 cycles
- Lifetime : 10 yesrs

14.2 Block Diagram

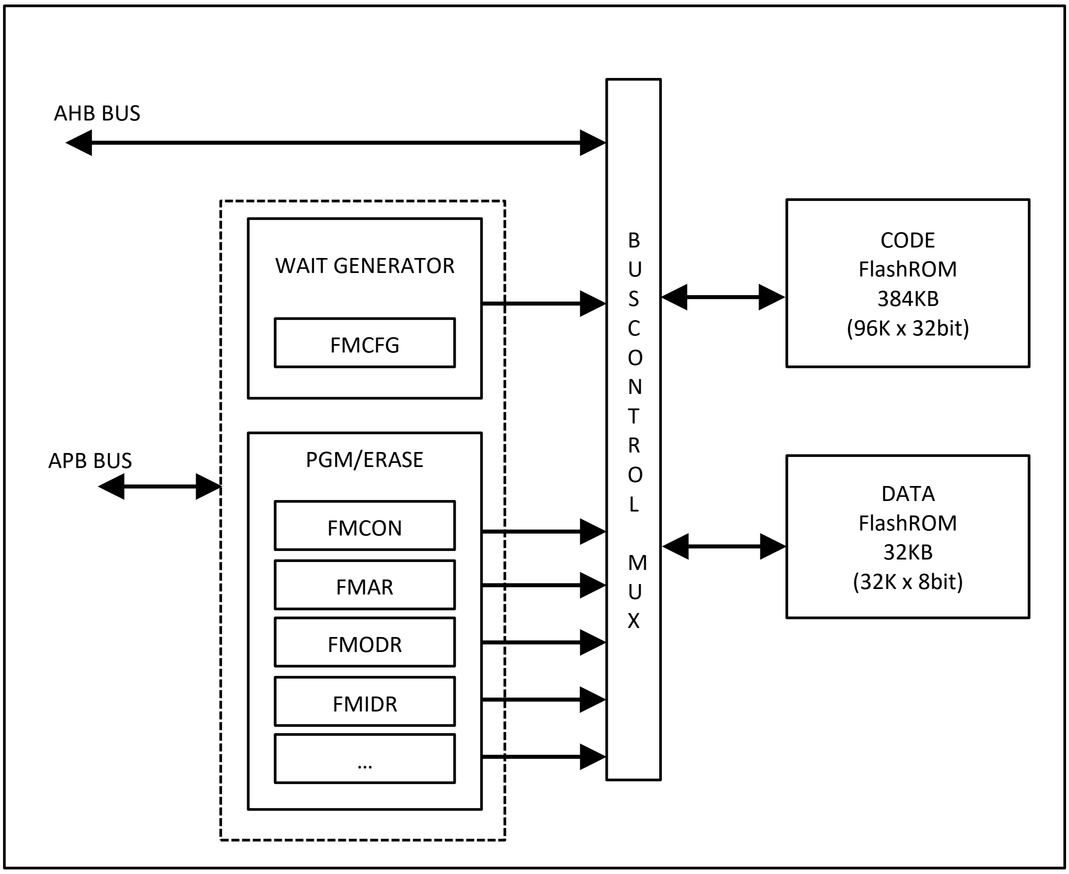
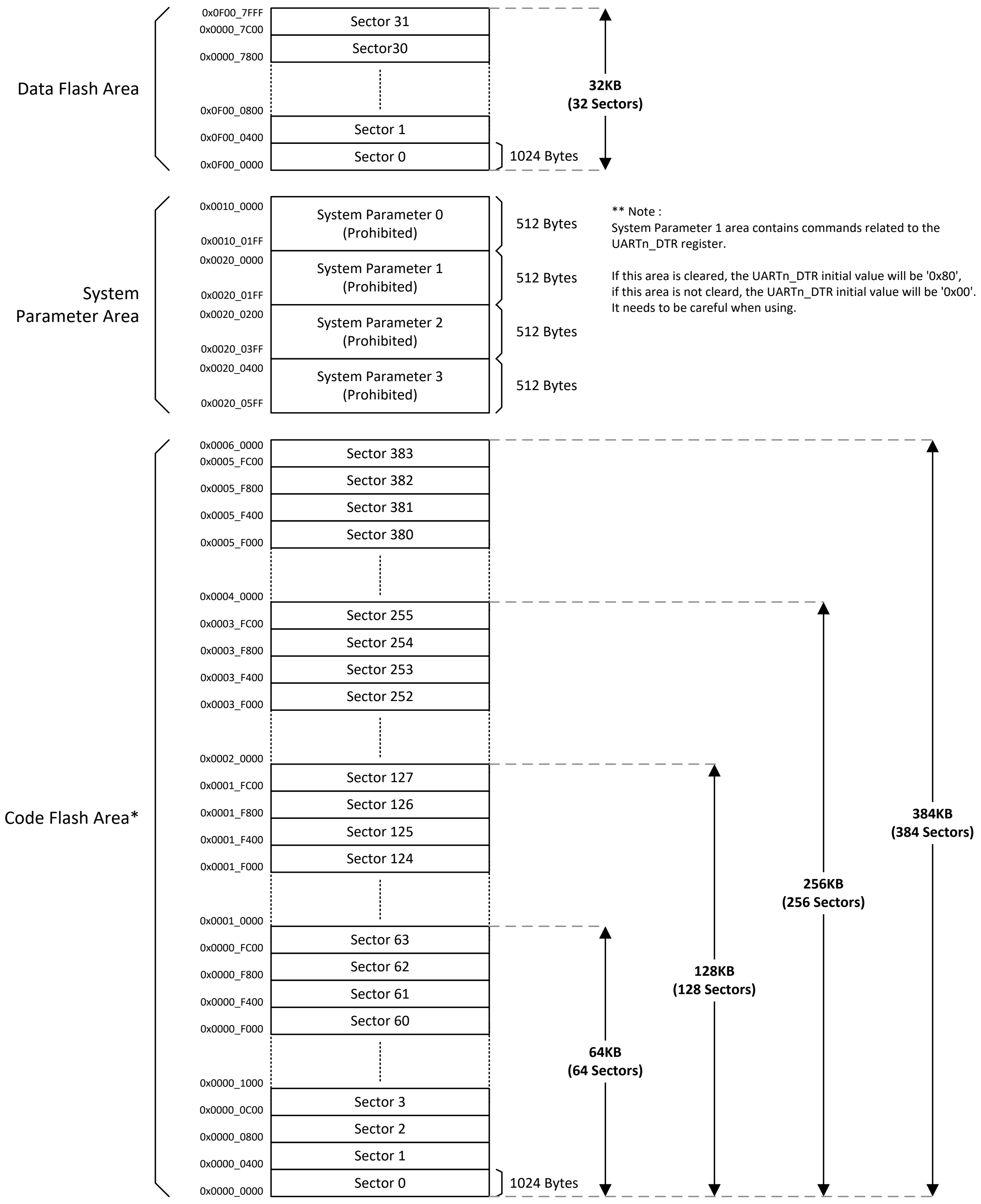


Figure 14.1. The block diagram of FMC (Flash memory controller)

14.2.1 Flash Memory Map

The flash memory map of the A33G52x is shown below.



** Note :
System Parameter 1 area contains commands related to the UARTn_DTR register.

If this area is cleared, the UARTn_DTR initial value will be '0x80',
if this area is not cleard, the UARTn_DTR initial value will be '0x00'.
It needs to be careful when using.

*** NOTE:** To write or erase data in the flash memory area, the address to be accessed must be explicated. If an address out of the flash memory area is accessed, system operation is not guaranteed.

Figure 14.2. A33G52x Flash Memory Map

14.2.2 Configuring Code Flash Memory Sectors

Code flash memory on the A33G52x is a sector configuration of 1KB units. Addresses for each sector are shown in the table below.

Table 14.2. Sector list of code flash memory

Sector No.	Sector Size	Address	Sector No.	Sector Size	Address	Sector No.	Sector Size	Address	Sector No.	Sector Size	Address
0	1KB	0x0000_0000	96	1KB	0x0001_8000	192	1KB	0x0003_0000	288	1KB	0x0004_8000
1	1KB	0x0000_0400	97	1KB	0x0001_8400	193	1KB	0x0003_0400	289	1KB	0x0004_8400
2	1KB	0x0000_0800	98	1KB	0x0001_8800	194	1KB	0x0003_0800	290	1KB	0x0004_8800
3	1KB	0x0000_0C00	99	1KB	0x0001_8C00	195	1KB	0x0003_0C00	291	1KB	0x0004_8C00
4	1KB	0x0000_1000	100	1KB	0x0001_9000	196	1KB	0x0003_1000	292	1KB	0x0004_9000
5	1KB	0x0000_1400	101	1KB	0x0001_9400	197	1KB	0x0003_1400	293	1KB	0x0004_9400
6	1KB	0x0000_1800	102	1KB	0x0001_9800	198	1KB	0x0003_1800	294	1KB	0x0004_9800
7	1KB	0x0000_1C00	103	1KB	0x0001_9C00	199	1KB	0x0003_1C00	295	1KB	0x0004_9C00
8	1KB	0x0000_2000	104	1KB	0x0001_A000	200	1KB	0x0003_2000	296	1KB	0x0004_A000
9	1KB	0x0000_2400	105	1KB	0x0001_A400	201	1KB	0x0003_2400	297	1KB	0x0004_A400
10	1KB	0x0000_2800	106	1KB	0x0001_A800	202	1KB	0x0003_2800	298	1KB	0x0004_A800
11	1KB	0x0000_2C00	107	1KB	0x0001_AC00	203	1KB	0x0003_2C00	299	1KB	0x0004_AC00
12	1KB	0x0000_3000	108	1KB	0x0001_B000	204	1KB	0x0003_3000	300	1KB	0x0004_B000
13	1KB	0x0000_3400	109	1KB	0x0001_B400	205	1KB	0x0003_3400	301	1KB	0x0004_B400
14	1KB	0x0000_3800	110	1KB	0x0001_B800	206	1KB	0x0003_3800	302	1KB	0x0004_B800
15	1KB	0x0000_3C00	111	1KB	0x0001_BC00	207	1KB	0x0003_3C00	303	1KB	0x0004_BC00
16	1KB	0x0000_4000	112	1KB	0x0001_C000	208	1KB	0x0003_4000	304	1KB	0x0004_C000
17	1KB	0x0000_4400	113	1KB	0x0001_C400	209	1KB	0x0003_4400	305	1KB	0x0004_C400
18	1KB	0x0000_4800	114	1KB	0x0001_C800	210	1KB	0x0003_4800	306	1KB	0x0004_C800
19	1KB	0x0000_4C00	115	1KB	0x0001_CC00	211	1KB	0x0003_4C00	307	1KB	0x0004_CC00
20	1KB	0x0000_5000	116	1KB	0x0001_D000	212	1KB	0x0003_5000	308	1KB	0x0004_D000
21	1KB	0x0000_5400	117	1KB	0x0001_D400	213	1KB	0x0003_5400	309	1KB	0x0004_D400
22	1KB	0x0000_5800	118	1KB	0x0001_D800	214	1KB	0x0003_5800	310	1KB	0x0004_D800
23	1KB	0x0000_5C00	119	1KB	0x0001_DC00	215	1KB	0x0003_5C00	311	1KB	0x0004_DC00
24	1KB	0x0000_6000	120	1KB	0x0001_E000	216	1KB	0x0003_6000	312	1KB	0x0004_E000
25	1KB	0x0000_6400	121	1KB	0x0001_E400	217	1KB	0x0003_6400	313	1KB	0x0004_E400
26	1KB	0x0000_6800	122	1KB	0x0001_E800	218	1KB	0x0003_6800	314	1KB	0x0004_E800
27	1KB	0x0000_6C00	123	1KB	0x0001_EC00	219	1KB	0x0003_6C00	315	1KB	0x0004_EC00
28	1KB	0x0000_7000	124	1KB	0x0001_F000	220	1KB	0x0003_7000	316	1KB	0x0004_F000
29	1KB	0x0000_7400	125	1KB	0x0001_F400	221	1KB	0x0003_7400	317	1KB	0x0004_F400
30	1KB	0x0000_7800	126	1KB	0x0001_F800	222	1KB	0x0003_7800	318	1KB	0x0004_F800
31	1KB	0x0000_7C00	127	1KB	0x0001_FC00	223	1KB	0x0003_7C00	319	1KB	0x0004_FC00
32	1KB	0x0000_8000	128	1KB	0x0002_0000	224	1KB	0x0003_8000	320	1KB	0x0005_0000
33	1KB	0x0000_8400	129	1KB	0x0002_0400	225	1KB	0x0003_8400	321	1KB	0x0005_0400
34	1KB	0x0000_8800	130	1KB	0x0002_0800	226	1KB	0x0003_8800	322	1KB	0x0005_0800
35	1KB	0x0000_8C00	131	1KB	0x0002_0C00	227	1KB	0x0003_8C00	323	1KB	0x0005_0C00
36	1KB	0x0000_9000	132	1KB	0x0002_1000	228	1KB	0x0003_9000	324	1KB	0x0005_1000
37	1KB	0x0000_9400	133	1KB	0x0002_1400	229	1KB	0x0003_9400	325	1KB	0x0005_1400
38	1KB	0x0000_9800	134	1KB	0x0002_1800	230	1KB	0x0003_9800	326	1KB	0x0005_1800
39	1KB	0x0000_9C00	135	1KB	0x0002_1C00	231	1KB	0x0003_9C00	327	1KB	0x0005_1C00
40	1KB	0x0000_A000	136	1KB	0x0002_2000	232	1KB	0x0003_A000	328	1KB	0x0005_2000
41	1KB	0x0000_A400	137	1KB	0x0002_2400	233	1KB	0x0003_A400	329	1KB	0x0005_2400
42	1KB	0x0000_A800	138	1KB	0x0002_2800	234	1KB	0x0003_A800	330	1KB	0x0005_2800
43	1KB	0x0000_AC00	139	1KB	0x0002_2C00	235	1KB	0x0003_AC00	331	1KB	0x0005_2C00
44	1KB	0x0000_B000	140	1KB	0x0002_3000	236	1KB	0x0003_B000	332	1KB	0x0005_3000
45	1KB	0x0000_B400	141	1KB	0x0002_3400	237	1KB	0x0003_B400	333	1KB	0x0005_3400
46	1KB	0x0000_B800	142	1KB	0x0002_3800	238	1KB	0x0003_B800	334	1KB	0x0005_3800
47	1KB	0x0000_BC00	143	1KB	0x0002_3C00	239	1KB	0x0003_BC00	335	1KB	0x0005_3C00
48	1KB	0x0000_C000	144	1KB	0x0002_4000	240	1KB	0x0003_C000	336	1KB	0x0005_4000
49	1KB	0x0000_C400	145	1KB	0x0002_4400	241	1KB	0x0003_C400	337	1KB	0x0005_4400
50	1KB	0x0000_C800	146	1KB	0x0002_4800	242	1KB	0x0003_C800	338	1KB	0x0005_4800
51	1KB	0x0000_CC00	147	1KB	0x0002_4C00	243	1KB	0x0003_CC00	339	1KB	0x0005_4C00
52	1KB	0x0000_D000	148	1KB	0x0002_5000	244	1KB	0x0003_D000	340	1KB	0x0005_5000
53	1KB	0x0000_D400	149	1KB	0x0002_5400	245	1KB	0x0003_D400	341	1KB	0x0005_5400
54	1KB	0x0000_D800	150	1KB	0x0002_5800	246	1KB	0x0003_D800	342	1KB	0x0005_5800
55	1KB	0x0000_DC00	151	1KB	0x0002_5C00	247	1KB	0x0003_DC00	343	1KB	0x0005_5C00
56	1KB	0x0000_E000	152	1KB	0x0002_6000	248	1KB	0x0003_E000	344	1KB	0x0005_6000
57	1KB	0x0000_E400	153	1KB	0x0002_6400	249	1KB	0x0003_E400	345	1KB	0x0005_6400
58	1KB	0x0000_E800	154	1KB	0x0002_6800	250	1KB	0x0003_E800	346	1KB	0x0005_6800
59	1KB	0x0000_EC00	155	1KB	0x0002_6C00	251	1KB	0x0003_EC00	347	1KB	0x0005_6C00
60	1KB	0x0000_F000	156	1KB	0x0002_7000	252	1KB	0x0003_F000	348	1KB	0x0005_7000
61	1KB	0x0000_F400	157	1KB	0x0002_7400	253	1KB	0x0003_F400	349	1KB	0x0005_7400
62	1KB	0x0000_F800	158	1KB	0x0002_7800	254	1KB	0x0003_F800	350	1KB	0x0005_7800

63	1KB	0x0000_FC00	159	1KB	0x0002_7C00	255	1KB	0x0003_FC00	351	1KB	0x0005_7C00
64	1KB	0x0001_0000	160	1KB	0x0002_8000	256	1KB	0x0004_0000	352	1KB	0x0005_8000
65	1KB	0x0001_0400	161	1KB	0x0002_8400	257	1KB	0x0004_0400	353	1KB	0x0005_8400
66	1KB	0x0001_0800	162	1KB	0x0002_8800	258	1KB	0x0004_0800	354	1KB	0x0005_8800
67	1KB	0x0001_0C00	163	1KB	0x0002_8C00	259	1KB	0x0004_0C00	355	1KB	0x0005_8C00
68	1KB	0x0001_1000	164	1KB	0x0002_9000	260	1KB	0x0004_1000	356	1KB	0x0005_9000
69	1KB	0x0001_1400	165	1KB	0x0002_9400	261	1KB	0x0004_1400	357	1KB	0x0005_9400
70	1KB	0x0001_1800	166	1KB	0x0002_9800	262	1KB	0x0004_1800	358	1KB	0x0005_9800
71	1KB	0x0001_1C00	167	1KB	0x0002_9C00	263	1KB	0x0004_1C00	359	1KB	0x0005_9C00
72	1KB	0x0001_2000	168	1KB	0x0002_A000	264	1KB	0x0004_2000	360	1KB	0x0005_A000
73	1KB	0x0001_2400	169	1KB	0x0002_A400	265	1KB	0x0004_2400	361	1KB	0x0005_A400
74	1KB	0x0001_2800	170	1KB	0x0002_A800	266	1KB	0x0004_2800	362	1KB	0x0005_A800
75	1KB	0x0001_2C00	171	1KB	0x0002_AC00	267	1KB	0x0004_2C00	363	1KB	0x0005_AC00
76	1KB	0x0001_3000	172	1KB	0x0002_B000	268	1KB	0x0004_3000	364	1KB	0x0005_B000
77	1KB	0x0001_3400	173	1KB	0x0002_B400	269	1KB	0x0004_3400	365	1KB	0x0005_B400
78	1KB	0x0001_3800	174	1KB	0x0002_B800	270	1KB	0x0004_3800	366	1KB	0x0005_B800
79	1KB	0x0001_3C00	175	1KB	0x0002_BC00	271	1KB	0x0004_3C00	367	1KB	0x0005_BC00
80	1KB	0x0001_4000	176	1KB	0x0002_C000	272	1KB	0x0004_4000	368	1KB	0x0005_C000
81	1KB	0x0001_4400	177	1KB	0x0002_C400	273	1KB	0x0004_4400	369	1KB	0x0005_C400
82	1KB	0x0001_4800	178	1KB	0x0002_C800	274	1KB	0x0004_4800	370	1KB	0x0005_C800
83	1KB	0x0001_4C00	179	1KB	0x0002_CC00	275	1KB	0x0004_4C00	371	1KB	0x0005_CC00
84	1KB	0x0001_5000	180	1KB	0x0002_D000	276	1KB	0x0004_5000	372	1KB	0x0005_D000
85	1KB	0x0001_5400	181	1KB	0x0002_D400	277	1KB	0x0004_5400	373	1KB	0x0005_D400
86	1KB	0x0001_5800	182	1KB	0x0002_D800	278	1KB	0x0004_5800	374	1KB	0x0005_D800
87	1KB	0x0001_5C00	183	1KB	0x0002_DC00	279	1KB	0x0004_5C00	375	1KB	0x0005_DC00
88	1KB	0x0001_6000	184	1KB	0x0002_E000	280	1KB	0x0004_6000	376	1KB	0x0005_E000
89	1KB	0x0001_6400	185	1KB	0x0002_E400	281	1KB	0x0004_6400	377	1KB	0x0005_E400
90	1KB	0x0001_6800	186	1KB	0x0002_E800	282	1KB	0x0004_6800	378	1KB	0x0005_E800
91	1KB	0x0001_6C00	187	1KB	0x0002_EC00	283	1KB	0x0004_6C00	379	1KB	0x0005_EC00
92	1KB	0x0001_7000	188	1KB	0x0002_F000	284	1KB	0x0004_7000	380	1KB	0x0005_F000
93	1KB	0x0001_7400	189	1KB	0x0002_F400	285	1KB	0x0004_7400	381	1KB	0x0005_F400
94	1KB	0x0001_7800	190	1KB	0x0002_F800	286	1KB	0x0004_7800	382	1KB	0x0005_F800
95	1KB	0x0001_7C00	191	1KB	0x0002_FC00	287	1KB	0x0004_7C00	383	1KB	0x0005_FC00

14.2.3 Configuring Data Flash Memory Sectors

Data flash memory on the A33G52x is a sector configuration of 1KB units. Addresses for each sector are shown in the table below.

Table 14.3. Sector list of data flash memory

Sector No.	Sector Size	Address	Sector No.	Sector Size	Address
0	1KB	0x0F00_0000	16	1KB	0x0F00_4000
1	1KB	0x0F00_0400	17	1KB	0x0F00_4400
2	1KB	0x0F00_0800	18	1KB	0x0F00_4800
3	1KB	0x0F00_0C00	19	1KB	0x0F00_4C00
4	1KB	0x0F00_1000	20	1KB	0x0F00_5000
5	1KB	0x0F00_1400	21	1KB	0x0F00_5400
6	1KB	0x0F00_1800	22	1KB	0x0F00_5800
7	1KB	0x0F00_1C00	23	1KB	0x0F00_5C00
8	1KB	0x0F00_2000	24	1KB	0x0F00_6000
9	1KB	0x0F00_2400	25	1KB	0x0F00_6400
10	1KB	0x0F00_2800	26	1KB	0x0F00_6800
11	1KB	0x0F00_2C00	27	1KB	0x0F00_6C00
12	1KB	0x0F00_3000	28	1KB	0x0F00_7000
13	1KB	0x0F00_3400	29	1KB	0x0F00_7400
14	1KB	0x0F00_3800	30	1KB	0x0F00_7800
15	1KB	0x0F00_3C00	31	1KB	0x0F00_7C00

14.3 Register Map

The base address of internal FMC (Flash Memory Controller) is 0x4000_0100 and the register list is show as follows.

Table 14.4. The register list of internal FMC (Flash Memory Controller)

Register	Offset	Access Type	Description	Initial Value	Ref
FMC_CFG	0x00	RW	Flash Memory Configuration Regsiter	0x00000303	14.4.1
FMC_CON	0x04	RW	Flash Memory Control Register	0x00000000	14.4.2
FMC_ODR	0x08	RW	Flash Memory Output Data Register	0x00000000	14.4.3
FMC_IDR	0x0C	RO	Flash Memory Input Data Register	0x00000000	14.4.4
FMC_AR	0x10	RW	Flash Memory Address Register	0x00000000	14.4.5
FMC_TEST	0x14	RW WO	Flash Memory Extension Mode Control Register	0x00000000	14.4.6
FMC_CRC	0x18	RW RO	Flash CRC Register	0x0000FFFF	14.4.7
FMC_PROTECT	0x1C	RW	Flash Memory Protection Register	0x00000000	14.4.8
FMC_RPROT	0x20	RW	Flash Read Protection Register	0x000000FF	14.4.9
FMC_BOOT	0x2C	RW	Flash Boot Mode Register	0x00000000	14.4.10

14.4 Register Description

14.4.1 FMC_CFG Flash Memory Configuration Reigster

The FMC_CFG register is a timing setting register that adjusts the speed of the internal flash memory bus according to the internal bus speed of system.

FMC_CFG=0x4000_0100

31–13	12–8	7	6–5	4–0
Reserved	DWAIT	CZWAIT	Reserved	CWAIT
-	00011	0	-	00011
-	RW	RW	-	RW

12 **8**	**DWAIT**	Wait value of Data Flash memory (Access timing is DWAIT + 1.5 clocks)
7	**CZWAIT**	Selection of zero-wait Code Flash (Do not use this function - Only for test) 0 Disable zero-wait 1 Enable zero-wait (for test only)
4 **0**	**CWAIT**	Wait value of Code Flash memory (Access timing is CWAIT + 1.5 clocks)

If CZWAIT is 1 (zero wait for code flash), the operating cycle of code flash's bus clock is 0.5 cycles and a maximum access timing of 16 MHz regardless a CWAIT value will be ignored. When CZWAIT is set to 0 (zero wait prohibition), the operating cycle that 1.5 cycles (Action 0.5 cycles + basic 1 cycle) is applied to the CWAIT setting value is applied for the flash access, and the maximum access timing is 25 MHz.

Then, A33G52x the maximum access timing of code or data flash memory can be obtained following fomula:

$$Code\ or\ Data\ Flash\ Access\ Timing = \frac{HCLK}{(0.5)} \leq 16MHz\ (\boldsymbol{CZWAIT = 1})$$

Otherwise,

$$Code\ or\ Data\ Flash\ Access\ Timing = \frac{HCLK}{(1.5 + CWAIT\ or\ DWAIT)} \leq 25MHz\ (\boldsymbol{CZWAIT = 0})$$

The table below shows flash access timing for each setting.

Table 14.5. Access Time Setting of Flash Memory depends on Bus Clock

Bus Clock (HCLK)	CZWAIT	CWAIT or DWAIT	Operating Cycle	Flash Access Timing	Max. Access Timing
8MHz	0	0	1.5	5.3MHz	25MHz
8MHz	1	X (Don't care)	0.5	16MHz	16MHz
16MHz	0	0	1.5	10.6MHz	25MHz
16MHz	1	X (Don't care)	-	Do not use	Do not use
20MHz	0	0	1.5	13.3MHz	25MHz
20MHz	0	1	2.5	8MHz	25MHz
20MHz	1	X (Don't care)	-	Do not use	Do not use
25MHz	0	0	1.5	16.6MHz	25MHz
25MHz	0	1	2.5	10MHz	25MHz
25MHz	1	X (Don't care)	-	Do not use	Do not use
32MHz	0	0	1.5	21.3MHz	25MHz
32MHz	0	1	2.5	12.8MHz	25MHz

32MHz	0	2	3.5	9.14MHz	25MHz
74MHz	0	2	3.5	21.1MHz	25MHz
74MHz	0	3	4.5	16.4MHz	25MHz
75MHz	0	2	3.5	21.4MHz	25MHz
75MHz	0	3	4.5	16.6MHz	25MHz

14.4.2 FMC_CON Flash Memory Control Register

It is a register that controls overall write / erase of flash memory and related operation control.

FMC_CON=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
RST	STOP	Reserved	SELF	SELF_CODE				Reserved		BBLK	FSRD	Reserved	PAGE	TRSL	TSMD	DSEL	CSEL	AE	OE	CS	Reserved	DTBIT	CTBIT	NVSTR	Reserved				MASE	SERA	PROG
0	0	-	0	0000				-		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	-	RO	RO	RW	-				RW	RW	RW

31	**RST**	Control flash reset signal	
		0	Disable flash reset signal
		1	Enable flash reset signal
30	**STOP**	Control flash status (When the MCU enters the PWDN state, the flash automatically sets the PWDN state even if this bit is set to '0')	
		0	Set flash normal state (Disable Flash Stop)
		1	Set flash powerdown state (Enable Flash Stop)
28	**SELF**	Code flash self program mode can be written with SELF_CODE [27:24] = 0110 (Self mode is SERA and PROG possible operating)	
		0	Disable flash self program
		1	Enable flash self program
27 **24**	**SELF_CODE**	Code value for self program of code flash (When performing self-programming function in code flash memory, 0110b value should be written in this field.)	
21	**BBLK**	BOOT block protection (Boot block protection during chip erase)	
		0	Disable
		1	Enable
20	**FSRD**	Fuse Area (For internal flash test)	
		0	Disable
		1	Enable
18	**PAGE**	Code/Data flash erase per page (512 Bytes) (It is NOT supported in its own program mode..)	
		0	Disable
		1	Enable
17	**TRSL**	Test registers access (For internal flash test)	
		0	Disable
		1	Enable
16	**TSMD**	TEST mode access (For internal flash test)	
		0	Disable
		1	Enable
15	**DSEL**	Data flash selection	
		0	NO select
		1	Select
		Code flash selecton	

14	**CSEL**	0	NO select
		1	select
13	**AE**	Address latch enable	
		0	Disable the output the address
		1	Enable the output the address (Read access at the rising edge)
12	**OE**	Output Enable	
		0	No action
		1	Code flash program mode
11	**CS**	Chip select	
		0	No action
		1	Select flashROM
9	**DTBIT**	The program/erase status flag of data flash	
		0	Data flash is idle (Completed)
		1	Data flash is busy
8	**CTBIT**	The program/erase status flag of code flash	
		0	Code flash is idle (Completed)
		1	Code flash is busy
7	**NVSTR**	Program/erase operation of flash memory	
		0	Stop
		1	Start
2	**MASE**	Mass (Full) erase of code or data flash memory	
		0	Disable
		1	Enable
1	**SERA**	Select Code or Data Flash Sector (1KB) erase	
		0	Disable
		1	Enable
0	**PROG**	Select Code or Data Flash program	
		0	Disable
		1	Enable

14.4.3 FMC_ODR Flash Memory Output Data Register

This register is used to write data to flash memory. In program mode, the data written to this register is stored in the flash memory.

FMC_ODR=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
FMODR																															
00000000 00000000 00000000 00000000																															
RW																															

31 **0**	**FMODR**	The output data to the flash memory

14.4.4 FMC_IDR Flash Memory Data Input Data Register

This register is used to read data stored in the flash memory in flash memory program mode.

FMC_IDR=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
FMIDR																															
XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX																															
RO																															

31 **0**	**FMIDR**	Input data from the flash memory

14.4.5 FMC_ARFlash Memory Address Register

This register sets the output address of the flash memory and is used in the flash memory program mode.
When setting the Code Flash memory address value, you can use a width of up to 18 bits.
When setting the Data Flash memory address value, use a width of up to 15 bits.
Since the memory address can be set in 1-word (4 Bytes) in Self Program in Code Flash memory, the bit value of A0 and A1 is always '0'. In contrast, the memory address can be set in 1-byte units in the program in the Data Flash memory, so that the values of A0 and A1 can be set to '1' or '0'.

FMC_AR=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
Reserved													ADDR																	A1	A0
-													00000000 00000000																	0	0
-													RW																	RW	RW

18 2	ADDR	Address output to the flash memory
1	A1	16-bit mode address A1 (for Data Flash)
0	A0	Byte mode address A0 (for Data Flash)

NOTE: To write or erase data in the flash memory area, the address to be accessed must be explicated. If an address out of the flash memory area is accessed, system operation is not guaranteed.

14.4.6 FMC_TEST Flash Memory Extended Mode Control Register

This register is the Extended mode control register of the flash memory. When Extended mode is enabled, the FMC_CON register bits [31:16] are enabled. When Extended mode is enabled, you can use Self-Program, Self-Erase, 512-byte Page Erase, and Boot Block Protection.
The NVSTR signal controlled by the AUTODIS bit is an internal signal, not the same as NVSTR in bit 7 of the FMC_CON register. The NVSTR bit in FMC_CON is the user interface bit, which must be processed by the user after the flash program.

FMC_TEST=0x4000_0114

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="8">WRITE_KEY</td><td colspan="3">Reserved</td><td>AUTODIS</td><td colspan="3">Resesrved</td><td>EX</td></tr>
<tr><td colspan="16">-</td><td colspan="8">00000000</td><td colspan="3">-</td><td>0</td><td colspan="3">-</td><td>0</td></tr>
<tr><td colspan="16">-</td><td colspan="8">WO</td><td colspan="3">-</td><td>RW</td><td colspan="3">-</td><td>RW</td></tr>
</table>

15 **8**	**WRITE_KEY**	To set the AUTODIS or EX bit, this field must be written with 0x4A. (WRITE KEY: 0x4A)	
4	**AUTODIS**	NVSTR Auto disable function	
		0	Enable auto-clear
		1	Disable auto-clear
0	**EX**	Extended Mode Selection	
		0	disable
		1	enable

14.4.7 FMC_CRC Flash Memory CRC Register

FMC_CRC is a register used to calculate the 16-bit CRC of the value stored in flash memory. Setting the CE bit to 0 initializes the CRC16 field to 0xFFFF. Then, after activating the CRC function with the CE bit set to 1, the address of the flash is designated as a pointer, and when the value stored in the flash is read, it is automatically calculated using the CRC-CCITT polynomial and the result is displayed in the CRC16 field. If the debugger is connected, close the memory window to calculate the correct CRC value.

FMC_CRC=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											CE	Reserved			SEL	CRC16															
-											0	-			0	11111111 11111111															
-											RW	-			RW	RO															

20	CE	Enable/Disable CRC16 function	
		0	Disable (CRC16 result Initialized to 0xFFFF))
		1	Enable
16	SEL	Selection of CRC calculation input data	
		0	Code Flash
		1	Data Flash
15 0	CRC16	The result of CRC16 (Cyclic Redundency Check 16) CRC-CCITT Polynomial : G1(x) = $x^{16} + x^{12} + x^5 + 1$	

***Note)** When CRC16 is read more than 16 times, the value read at that time is calculated.

14.4.8 FMC_PROTECT Flash Memory Protection Register

This register protects the internal flash memories from unintended write and erase operation
If written ‘0’ to a bit, the relavant sectors are protected to be programmed or erased.
If written ‘1’ to a bit, the relavant sectors are can be programmed or erased.
(In case of writing ‘0’, the NVSTR signal to the relavant address or sector is masked.)

FMC_PROTECT=0x4000_011C

31	30	29	28	27	26	25	24	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																DIP	CIP	BP5	BP4	DP3	DP2	DP1	DP0	BP3	BP2	BP1	BP0	SP3	SP2	SP1	SP0
-																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	**DIP**	Always ‘0’
14	**CIP**	Always ‘0’
13	**BP5**	Protection bit from 0x0005_0000 to 0x0005_FFFF
12	**BP4**	Protection bit from 0x0004_0000 to 0x0004_FFFF
11	**DP3**	Protection bit of the Data Flash 8KB #3
10	**DP2**	Protection bit of the Data Flash 8KB #2
9	**DP1**	Protection bit of the Data Flash 8KB #1
8	**DP0**	Protection bit of the Data Flash 8KB #0
7	**BP3**	Protection bit from 0x0003_0000 to 0x0003_FFFF
6	**BP2**	Protection bit from 0x0002_0000 to 0x0002_FFFF
5	**BP1**	Protection bit from 0x0001_0000 to 0x0001_FFFF
4	**BP0**	Protection bit from 0x0000_4000 to 0x0000_FFFF
3	**SP3**	Protection bit from 0x0000_3000 to 0x0000_3FFF
2	**SP2**	Protection bit from 0x0000_2000 to 0x0000_2FFF
1	**SP1**	Protection bit from 0x0000_1000 to 0x0000_1FFF
0	**SP0**	Protection bit from 0x0000_0000 to 0x0000_0FFF

14.4.9 FMC_RPROT Flash Memory Read Protection Register

This register is used to set the protection to the read, access to the flash memory.

FMC_RPROT=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	RPROT	Reserved	CERS	Reserved																				FMRPROT							
-	0	-	0	-																				11111111							
-	RO	-	RO	-																				RW							

30	**RPROT**	Display the read protection status	
		0	Disabled (Allow read access)
		1	Enabled (Prevent read access)
28	**CERS**	Display of the status that has been flash erase (This bit is set to 1 when the 'Chip Erase' command is executed during the erase command)	
		0	chip-erase NOT done
		1	chip-erase done
7 **0**	**FMRPROT**	Setting read access protection (If read protection is enabled and a debugger is connected, The flash value is scrambled in the memory window and read as 0xAA55AA55.)	
		No protection	Permitted read access
		Protection from Debug	Permitted read access, protected from debugger access (Set with 0x75)
		Protection Level 1	Protected from read access, permitted debugger access (Set with 0x39)

14.4.10 FMC_BOOT Flash Memory Boot Register

This register identifies the boot mode and indicates the operating status of BootRom. The BOOT field can only be accessed in boot mode that boot code located in BootRom is executed. In normal user code execution mode, BOOT field value is always displayed 0.

FMC_PROTECT=0x4000_012C

31	30	29	28	27	26	25	24	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
-																															RW

0	**BOOT**	The operation status of bootROM. This field is '0' in normal user mode.

CHAPTER 15. Internal SRAM

Table 15.1. Operation Summary

Item	Description	Remark
Clock usage	HCLK	Access speed
Reset Source	None	
Reset Generation	None	
Interrupt Generation	None	
Interrupt Clear Method	None	

15.1 Overview

The A33G52x has a built-in 24-kbyte SRAM capable of data and code area.

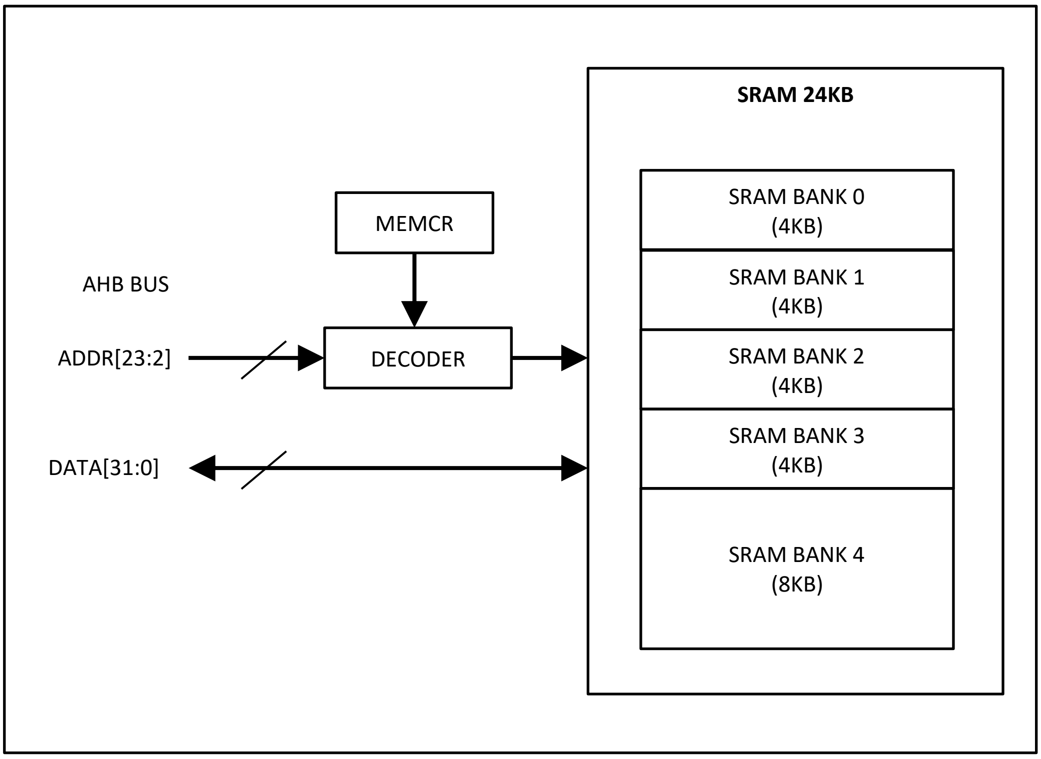


Figure 15.1. SRAM Block Diagram

15.1.1 Bank Configuration of SRAM

The SRAM consists of several blocks as below.

0x2000_0000 0x2000_0FFF	SRAM Block 0	remapped at 0x0000_0000 (Area 0)
0x2000_1000 0x2000_1FFF	SRAM Block 1	remapped at 0x0000_1000 (Area 1)
0x2000_2000 0x2000_2FFF	SRAM Block 2	remapped at 0x0000_2000 (Area 2)
0x2000_3000 0x2000_3FFF	SRAM Block 3	remapped at 0x0000_3000 (Area 3)
0x2000_4000 0x2000_5FFF	SRAM Block 4	Fixed Block

Figure 15.2. Configuration of the Banks of SRAM

15.2 Register Map

To use the SRAM, there is a register that performs remap control. The base address of the SRAM block is 0x4000_0E00, and the register map is as shown in the following table.

Table 15.2. SRAM Control Register

Register	Offset	Access Type	Description	Initial Value	Ref
SRAM_CR	0x00	RW	SRAM Memory Control Register	0x00000000	15.3.1

15.3 Register Description

15.3.1 SRAM_CR SRAM Memory Control Reigster

The SRAM_CR register provides the mapping control of the corresponding area in units of 4 KB of the maximum 16 KB SRAM area.

SRAM_CR=0x4000_00E0

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="28">Reserved</td><td>REMAP3</td><td>REMAP2</td><td>REMAP1</td><td>REMAP0</td></tr>
<tr><td colspan="28">-</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="28">-</td><td>RW</td><td>RW</td><td>RW</td><td>RW</td></tr>
</table>

3	**REMAP3**	0x00003000 ~ 0x00003FFF Area (Area 3)	
		0	Set Flash ROM at relavant area
		1	Remap SRAM at relavant area
2	**REMAP2**	Remap 0x00002000 ~ 0x00002FFF Area (Area 2)	
		0	Set Flash ROM at relavant area
		1	Remap SRAM at relavant area
1	**REMAP1**	Remap 0x00001000 ~ 0x00001FFF Area (Area 1)	
		0	Set Flash ROM at relavant area
		1	Remap SRAM at relavant area
0	**REMAP0**	Remap 0x00000000 ~ 0x00000FFF Area (Area 0)	
		0	Set Flash ROM at relavant area
		1	Remap SRAM at relavant area

CHAPTER 16. Electrical Characteristics

16.1 DC Characteristics

16.1.1 Absolute Maximum Ratings

The maximum rating of A33G527 is shown as follows

Table 16.1. Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Digital Power Supply	VDD	-0.3 ~ +6.5	V
Analog Power Supply	AVDD	-0.3 ~ +5.5	V
Input Voltage	VI	VSS-0.3 ~ VDD+0.3	V
Output Voltage	VO	VSS-0.3 ~ VDD+0.3	V
Output Current	IOH	10	mA
	∑IOH	80	
	IOL	20	
	∑IOL	160	
Input Main Clock Range	fXIN	8	MHz
Operating Frequency	fPLLOUT	75	MHz
Total Power Dissipation	PT	600	mW
Storage Temperature	T_{STG}	-45 ~ +125	℃
Maximum Operating Junction Temperature	T_J	105	℃

Table 16.2. The Recommandation for Operating

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Digital Power Supply	VDD	4 ~ 75MHz	3.0	5.0	5.5	V
Analog Power Supply	AVDD		3.0	-	VDD	V
Operating Temperature	T_{OPR}	VDD=3.0~5.5V	-40	-	85	℃
Operating Frequency	FREQ	fXIN	-	4	8	MHz
		Internal OSC		16		
		PLL	4	-	75	
Supply Rise Rate	t_{rVDD}		-	-	0.05	V/us
Supply Fall Rate	t_{fVDD}		-	-	0.1	

16.1.2 I/O Ports DC Characteristics

The DC Characteristics of A33G52x I/O ports is shown as follows

Table 16.3. I/O Ports DC Characteristics(1) (VDD = +5V, VSS=0V, Temperature=25℃)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Input Low Voltage	V_{IL}	Port A,B,C,E,F	-0.3	-	1	V
		Port D	-0.3	-	1	V
Input High Voltage	V_{IH}	Port A,B,C,E,F	4	-	5.3	V
		Port D	2.7	-	5.3	V
Hysteresis Voltage[1]	V_H		-	1.0	-	V
Output Low Voltage (Port A,B,C,E,F)	V_{OL}	I_{OL}=3mA	0	-	1	V
Output Low Voltage (Port D)		I_{OL}=18mA	0	-	1	V
Output High Voltage (Port A,B,C,E,F)	V_{OH}	I_{OH}=-1.2mA	4	-	5	V
Output High Voltage (Port D)		I_{OH}=-8mA	4	-	5	V
Output Low Current	I_{OL}	V_{OL}=1V (Port A,B,C,E,F)	+3	-	-	mA
		V_{OL}=1V (Port D)	+18	-	-	mA
Output High Current	I_{OH}	V_{OH}=4V (Port A,B,C,E,F)	-	-	-1.2	mA
		V_{OH}=4V (Port D)	-	-	-8	mA
Input Low Leakage Current	I_{IL}		-4	-	-	uA
Input High Leakage Current	I_{IH}		-	-	+4	uA
Pull-up Resistor	R_{PU}		10	-	60	kΩ
Pull-down Resistor	R_{PD}		40		70	kΩ

[1] **Hysteresis voltage between schmitt trigger switching levels. Based on characterization, NOT tested in production**

Table 16.4 I/O Ports DC Characteristics(2) (VDD = +3.3V, VSS=0V, Temperature=25℃)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Input Low Voltage	V_{IL}	Port A,B,C,E,F	-0.3	-	0.6	V
		Port D	-0.3	-	0.6	V
Input High Voltage	V_{IH}	Port A,B,C,E,F	2.7	-	3.6	V
		Port D	2.2	-	3.6	V
Hysteresis Voltage[1]	V_H	Port A,B,C,E,F	-	1.0	-	V
Output Low Voltage (Port A,B,C,E,F)	V_{OL}	I_{OL}=2mA	0	-	0.66	V
Output Low Voltage (Port D)		I_{OL}=10mA	0	-	0.66	V
Output High Voltage (Port A,B,C,E,F)	V_{OH}	I_{OH}=0.8mA	2.64	-	3.3	V
Output High Voltage (Port D)		I_{OH}=6mA	2.64	-	3.3	V
Output Low Current	I_{OL}	V_{OL}=0.6V (Port A,B,C,E,F)	+2	-	-	mA
		V_{OL}=0.6V (Port D)	+10	-	-	mA
Output High Current	I_{OH}	V_{OH}=2.7V (Port A,B,C,E,F)	-	-	-0.8	mA
		V_{OH}=2.7V (Port D)	-	-	-6	mA
Input Low Leakage Current	I_{IL}		-4	-	-	uA
Input High Leakage Current	I_{IH}		-	-	4	uA
Pull-up Resistor	R_{PU}		10	-	100	kΩ
Pull-down Resistor	R_{PD}		40		70	kΩ

[1] Hysteresis voltage between schmitt trigger switching levels. Based on characterization, NOT tested in production.

16.1.3 POR (Power On Reset) Operating Characteristics

The characteristics of the Power-on-Reset is shown as follows

Table 16.5 The POR Operating Characteristics (Temperature range : -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Current	IDD	Typ. <6uA If always on	-	60	-	nA
POR Set Level	VIH	VDD rising (slow)	1.05	1.20	1.35	V
POR Reset Level	VIL	VDD falling (slow)	1.00	1.10	1.20	V

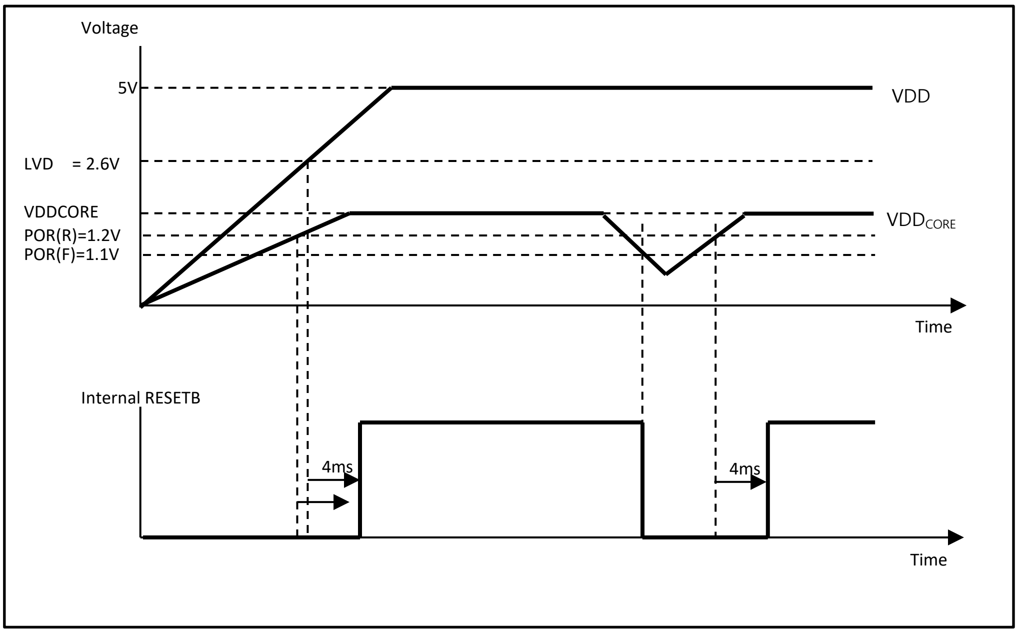


Figure 16.1. The Reset Timing

16.1.4 LVD (Low Voltage Detector) Operating Characteristics

The Characteristics of the LVD is shown as follows

Table 16.6 The LVD (Low Voltage Detector) Operating Characteristics (Temperature range : -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Current	IDD	-	-	-	50	uA
Stop Current	I_{STOP}	-	-	-	-	uA
LVD LEVEL	V_{DET}	2.60V 2.80V 3.00V 3.30V 3.75V 4.00V 4.25V 4.50V	-10	-	+10	%
Hysteresis	V_H	2us delay		100		mV

Once the VDD level has fallen below the LVD level, it does not guarantee the operation before the stabilization time about 200us after rising over the LVD voltage.
The operating timing of LVD is shown as below.

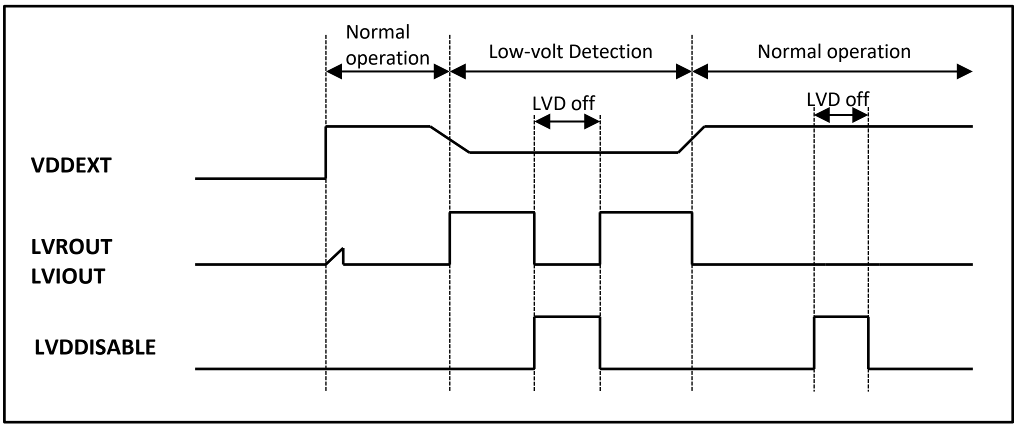


Figure 16.2. The Reset Timing

16.1.5 DC Characteristics of Internal Oscillator (IOSC16, RINGOSC)

The DC characteristics of internal 16MHz RC-Oscillator and RING-Oscillator are shown as follows.

Table 16.7 DC Characteristics of Internal Oscillator (Temperature=0 ~ +40℃)

Item	Min	Typ	Max	Unit	Remark
IDD	-	160	250	uA	
Output frequency	15.76	16	16.24	MHz	±1.5%
Output frequency duty	40	-	60	%	

Table 16.8 DC Characteristics of Internal Oscillator (Temperature=-20 ~ +70℃)

Item	Min	Typ	Max	Unit	Remark
IDD	-	160	250	uA	
Output frequency	15.52	16	16.48	MHz	±3%
Output frequency duty	40	-	60	%	

Table 16.9 DC Characteristics of Internal Oscillator (Temperature=-40 ~ +85℃)

Item	Min	Typ	Max	Unit	Remark
IDD	-	160	250	uA	
Output frequency	15.36	16	16.64	MHz	±4%
Output frequency duty	40	-	60	%	

Table 16.10 DC Characteristicss of Internal Ring Oscillator (Temperature=-40 ~ +85℃)

Item	Min	Typ	Max	Unit	Remark
IDD	-	4	8	uA	
Output frequency	0.5	1.0	1.5	MHz	±50%

16.1.6 DC Characteristics of Main and Sub Crystal Oscillator

The operating characteristics of the main oscillator and sub oscillator built into the A33G52x are shown in the table below. To enhance noise immunity, the A33G52x adopts a full-swing scheme.

Table 16.11 Characteristics of Main Crystal Oscillator (Temperature=-40 ~ +85℃)

Item	Min	Typ	Max	Unit	Remark
IDD	-	500*	1000	uA	@8MHz/5V
Operating Frequency	4	8	10	MHz	
Output Voltage (V_{XOUT})	0.8 VDD		-	V	
Output Currrent (I_{XOUT})	3**			mA	
Load Capacitance		18/22	35	pF	

[*] The swing scheme changed from Half-Swing to Full-Swing, the IDD current was adjusted.

[**] Output Current(I_{XOUT}) indicates the current driving capacity of the transistor that constitutes the oscillation part.

Table 16.12 Built-in sub oscillator operating characteristics (temperature = -40 to + 85℃)

Item	Min	Typ	Max	Unit	Remark
IDD	-	4	7	uA	
Operating Frequency	-	32.768	-	kHz	
Output Voltage	-	1.5	-	V	
Load Capacitance	5	15	25	pF	

16.1.7 DC Characteristics of Internal PLL

The operating characteristics of the PLL built in the A33G52x are shown in the table below.

Table 16.13 The Characteristics of Internal PLL (Temperature=-40 ~ +85℃)

Item	Min	Typ	Max	Unit	Remark
IDD	-	1.3	-	mA	75MHz
Output Frequency	8	75	90	MHz	
Duty	40	-	60	%	
P-P Jitter	-	-	500	ps	@Lock state
VCO Frequency (Extended mode)	50	-	200	MHz	
VCOx2[1] Frequency(Extended mode)	100		250	MHz	
Input Frequecy	4	-	8	MHz	
Lock Time	-	-	10	ms	

1) VCOx2 means that VCOMODE bit of PMU_PLLCON register is set to '1' then VCO Doubler Mode is enabled.

16.1.8 Current Consumption for Each Mode

The current consumption for each operation mode is shown as follows.

Table 16.14 Current Consumption for Each Mode (Room Temperature)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Normal Operation (Run Mode)	IDD_{NORMAL}	ROSC=RUN IOSC16=RUN MXOSC=RUN SUBOSC=RUN	-	30	-	mA
Sleep Mode (Idle Mode)	IDD_{SLEEP}	ROSC=RUN IOSC16=STOP MXOSC=STOP SUBOSC=STOP	-	500	-	uA
Deep-Sleep Mode (PowerDown Mode)	IDD_{STOP}	ROSC=STOP IOSC16=STOP MXOSC=STOP SUBOSC=STOP	-	100	-	uA

16.1.9 Specifications of Internal Flash Memory

The specifications of built-in flash memories are shown as follows.

Table 16.15 Specification of the Code Flash Memory

Parameter	Min	Typ.	Max	Unit
Operating Temperature	-40	-	+85	℃
Total code memory size*	-	-	384	KB
Unit sector size	1024	-	-	byte
BUS width	-	-	32	bit
Erase/program Endurance	10,000	-	-	cycle
Data retention	10	-	-	year

Note *) A33G527 : 384KB, A33G526 : 256KB, A33G524: 128KB

Table 16.16 Characteristics of the Code Flash Memory

Parameter	Symbol	Min	Typ.	Max	Unit
Standby current	I_{SB}	-	-	10	uA
Active read current (f=25MHz)	I_{CC1}	-	-	10	mA
Write current	I_{CC2}	-	-	7	mA
Erase current	I_{CC3}	-	-	7	mA
Read access time	tAC	40	-	-	ns
Read cycle time	tAAD	40	-	-	ns
Program time	tPROG	21	30	39	us
Sector erase time	tSER	2.8	4	5.2	ms
Macro erase time	tMER	5.6	8	10.4	ms

Table 16.17 Specification of the Data Flash Memory

Parameter	Min	Typ.	Max	Unit
Operating Temperature	-40	-	+85	℃
Total data memory size	-	-	32	KB
Unit sector size	1024	-	-	byte
BUS width	-	-	8	bit
Erase/program endurance	100,000	-	-	cycle
Data retention	10	-	-	year

Table 16.18 Characteristics of the Data Flash Memory

Parameter	Symbol	Min	Typ.	Max	Unit
Standby current	I_{SB}	-	1	10	uA
Active read current (f=20MHz)	I_{CC1}	-	-	3	mA
Write current	I_{CC2}	-	5	7	mA
Erase current	I_{CC3}	-	5	7	mA
Read access time	tAC	40	-	-	ns
Read cycle time	tAAD	40	-	-	ns
Program time	tPROG	21	30	39	us
Sector erase time	tSER	2.8	4	5.2	ms
Macro erase time	tMER	5.6	8	10.4	ms

16.1.10 A/D Converter Operating Characteristics

The electrical characteristics of A/D converter of A33G52x are as follows

Table 16.19 The Electrical Characteristics of A/D converter (Temperature=-20 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	AVDD		3.0	5	5.5	V
Resolution				12		Bit
Operating Current	IDDA	VDD=5.0V		1	2	mA
Analog Input Range	VAN		VSS		AVDD	V
Conversion Rate	fCON	5.0V<AVDD <5.5V		60*MCLK	66.6	kSPS
Operating Frequency	MCLK				4	MHz
DC Accuracy	INL	AVDD=5.0V, T_A = 25 ˚C		±8		LSB
	DNL			±2		LSB
Zero Offset Error*	ZOE			2		LSB
Full Scale Error**	FSE			2		LSB

Notes)

* Zero Offset Error is the difference between the zero input voltage (VSS) converted to a digital value and 0x000.

** full Scale Error is the difference between the top input voltage (VDD) converted to a digital value and 0xFFF.

16.2 AC Characteristics

16.2.1 AC Characteristics of the Internal Flash Memory

Table 16.20 AC Characteristics of the Internal Flash Memory (Room Temperature)

Parameter	Symbol	Min	Typ.	Max	Unit
Address setup time	tAS	2		-	ns
Address hold time	tAH	2		-	ns
Setup time of write and erase	tS	5		-	ns
Hold time of write and erase	tH	2		-	ns
Data setup time	tDS	2		-	ns
Data hold time	tDH	2		-	ns
Read access time	tAC	-		40	ns
AE pulse width	tAE	10		-	ns
AE to AE delay during read (Read cycle time)	tAAD	40		-	ns
AE low pulse width	tAEL	10			ns
Program time	tPROG	-		20	us
Sector erase time	tSER	-		2	ms
Macro erase time	tMER	-		10	ms
NVSTR to AE delay	tNVSTR	10		-	ns
CS setup time to AE rising edge	tCS	5		-	ns
TBIT rises after NVSTR	tTR	-		100	ns
TBIT falls after NVSTR for write	tTF	21	30	39	us
TBIT falls after NVSTR for sector erase	tTF	2.8	4	5.2	ms
TBIT falls after NVSTR for macro erase	tTF	5.6	8	10.4	ms

16.2.2 I2C Characteristics

The following table and figure show the timing codition of SDA and SCL bus lines for I2C bus devices.

Table 16.21 Timing Characteristics of the I2C

Parameter	Symbol	STANDARD MODE		FAST MODE		Unit
		Min	Max	Min	Max	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Hold time after (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD;STA}$	4.0	-	0.6	-	us
LOW period of the SCL clock	t_{LOW}	4.7	-	1.3	-	us
HIGH period of the SCL clock	t_{HIGH}	4.0	-	0.6	-	us
Setup time for a repeated START condition	$t_{SU;STA}$	4.7	-	0.6	-	us
Data hold time	$t_{HD;DAT}$	0	3.45	0	0.9	us
Data setup time	$t_{SU;DAT}$	100	-	100	-	ns
Clock/data fall time	t_F	0	300	0	300	ns
Clock/data rise time	t_R	0	1000	0	300	ns
Setup time for STOP condition	$t_{SU;STO}$	4.0	-	0.6	-	us
Bus free time between a STOP and START condtion	t_{BUF}	4.7	-	1.3	-	us

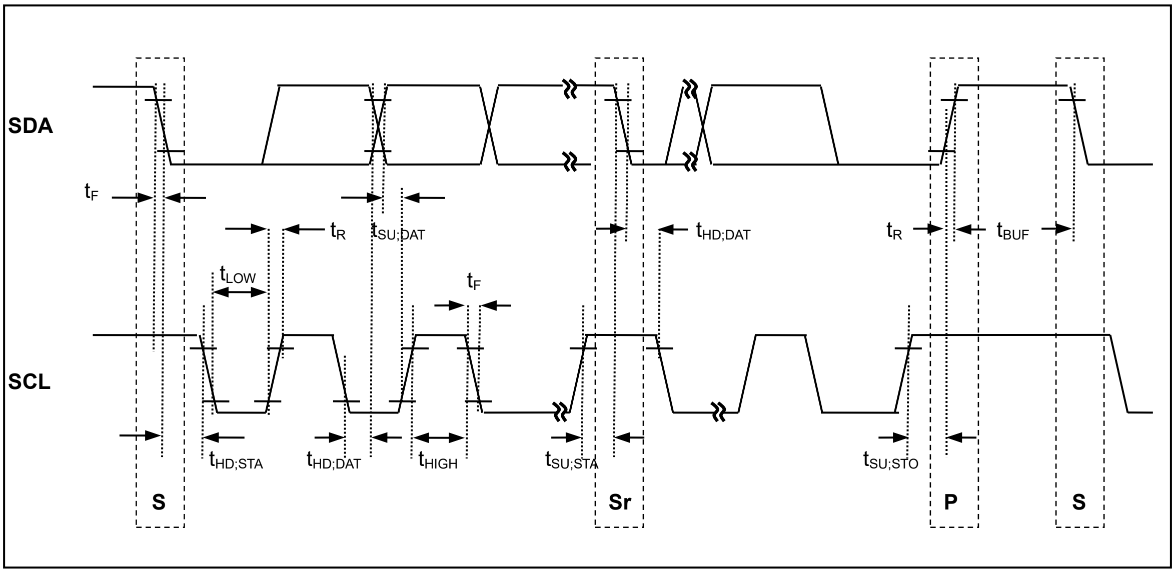


Figure 16.3. The timing of I2C interface bus

16.3 Characteristics of Analog Block

16.3.1 Power-On-Reset

A33G52x has a built-in Power-On Reset (POR) that monitors the internal voltage of VDD-Core and generates a reset signal for the MCU's cold start.

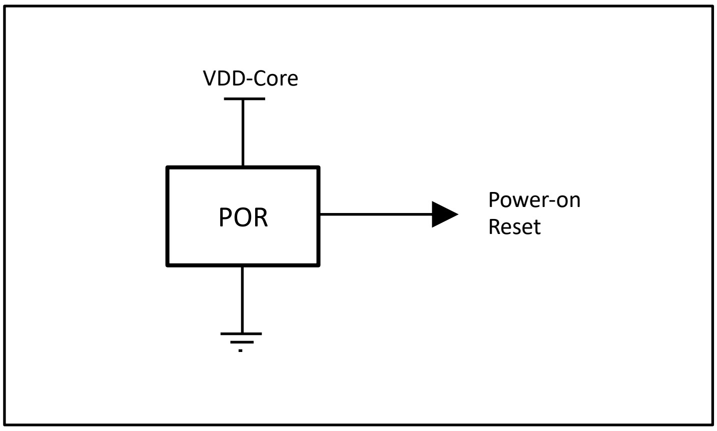


Figure 16.4. POR Block diagram

16.3.2 VDC (Voltage-Down Converter)

Built-in VDC generates the power regulated from external power for internal logics and analog block.

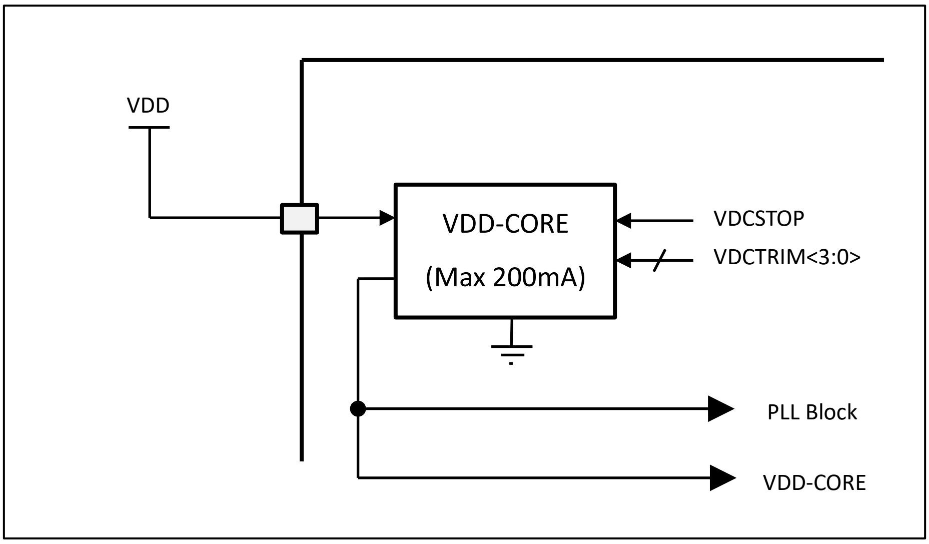


Figure 16.5. VDC Block diagram

16.3.3 LVD (Low Voltage Detector)

A33G52x has two built-in LVDs. Each LVD generate interrupt or reset depend on their settings. LVDs are monitoring external +5V power.
The configuration of LDCs is shown as follows.

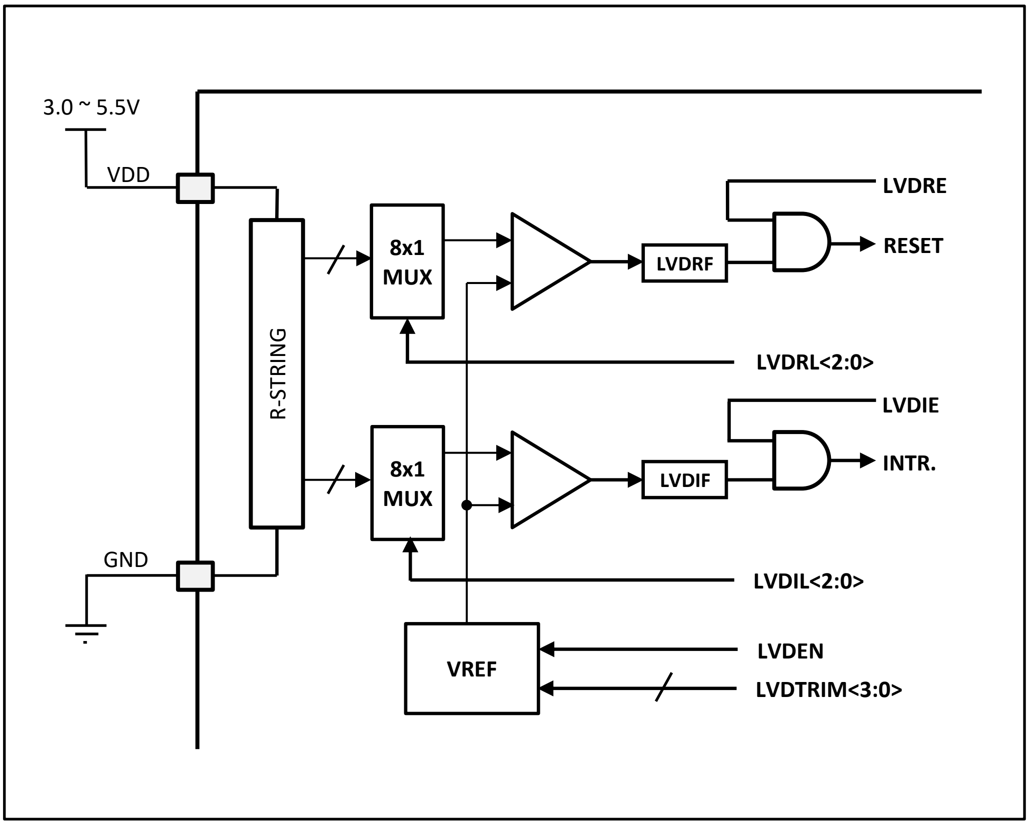


Figure 16.6. The Internal Configuration of LVDs

16.3.4 Internal Ring Oscillator (RingOSC/ROSC)

A33G52x has a simple ring oscillator to monitor system and supply to other peripherals. The Ring oscillator generates the frequency of 1MHz. It used for such as POR-Reset, WDT and clock monitoring. And for more smaller power consumption in the sleep mode, it used as the system operating clock. But ring oscillator is not desirable to be used in where need precise frequency due to the characteristics that ring oscillator is easily affected by the power and temperature.
The block diagram of the ring oscillator is shown as follows.

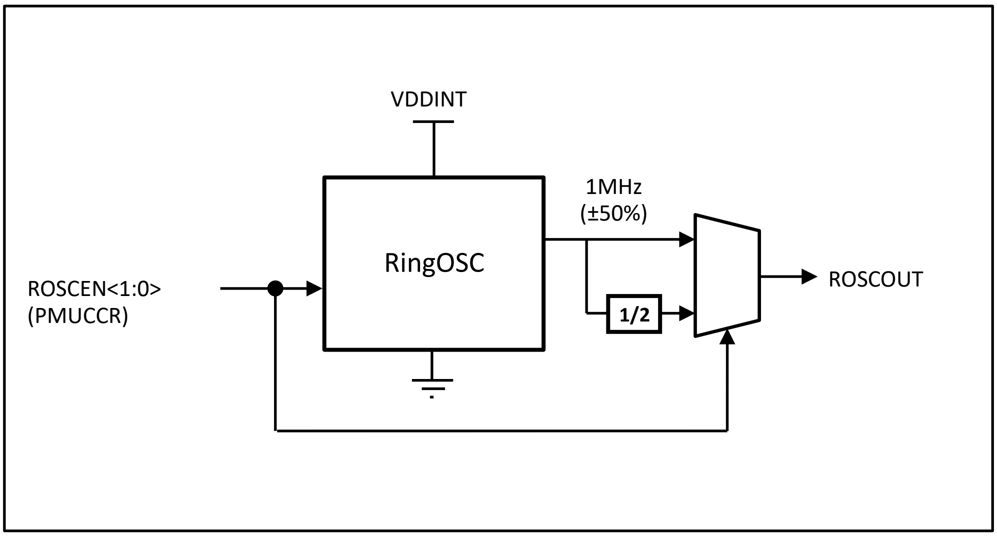


Figure 16.7. The Internal Configuration of RingOSC

16.3.5 Internal 16MHz Oscillator (IOSC16)

A33G52x has built-in 16MHz oscillator so that it can operate without any external clock.
The configuration of internal 16MHz oscillator is shown as follows

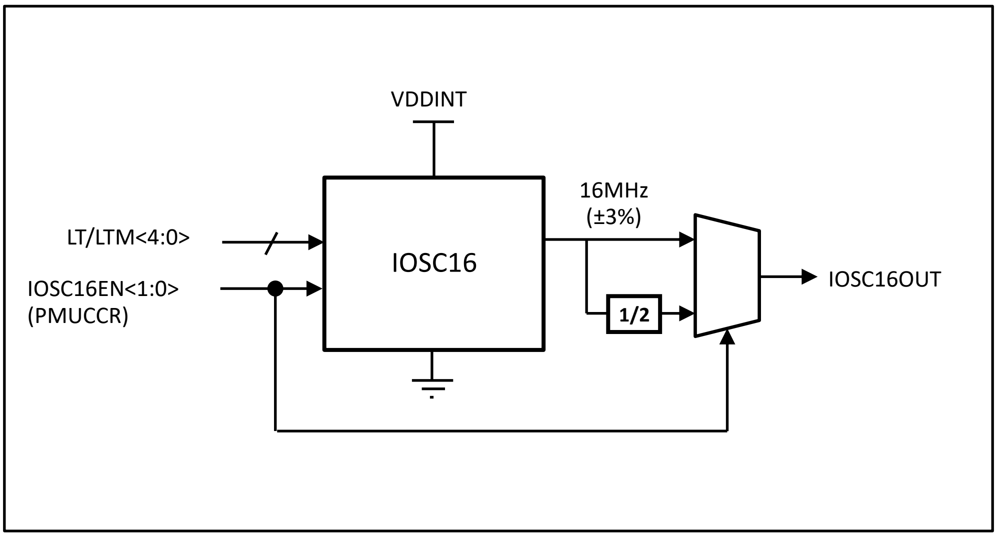


Figure 16.8. The Internal Configuration of IOSC16

16.3.6 PLL (Phase-Locked-Loop)

The built-in PLL can multiply internal or external clock to Max 75MHz. The PLL frequency synthesizer can be used to set the PLL output frequency up to 75MHz in 1MHz increments. The VDD-CORE power required for the PLL is supplied from inside the MCU.

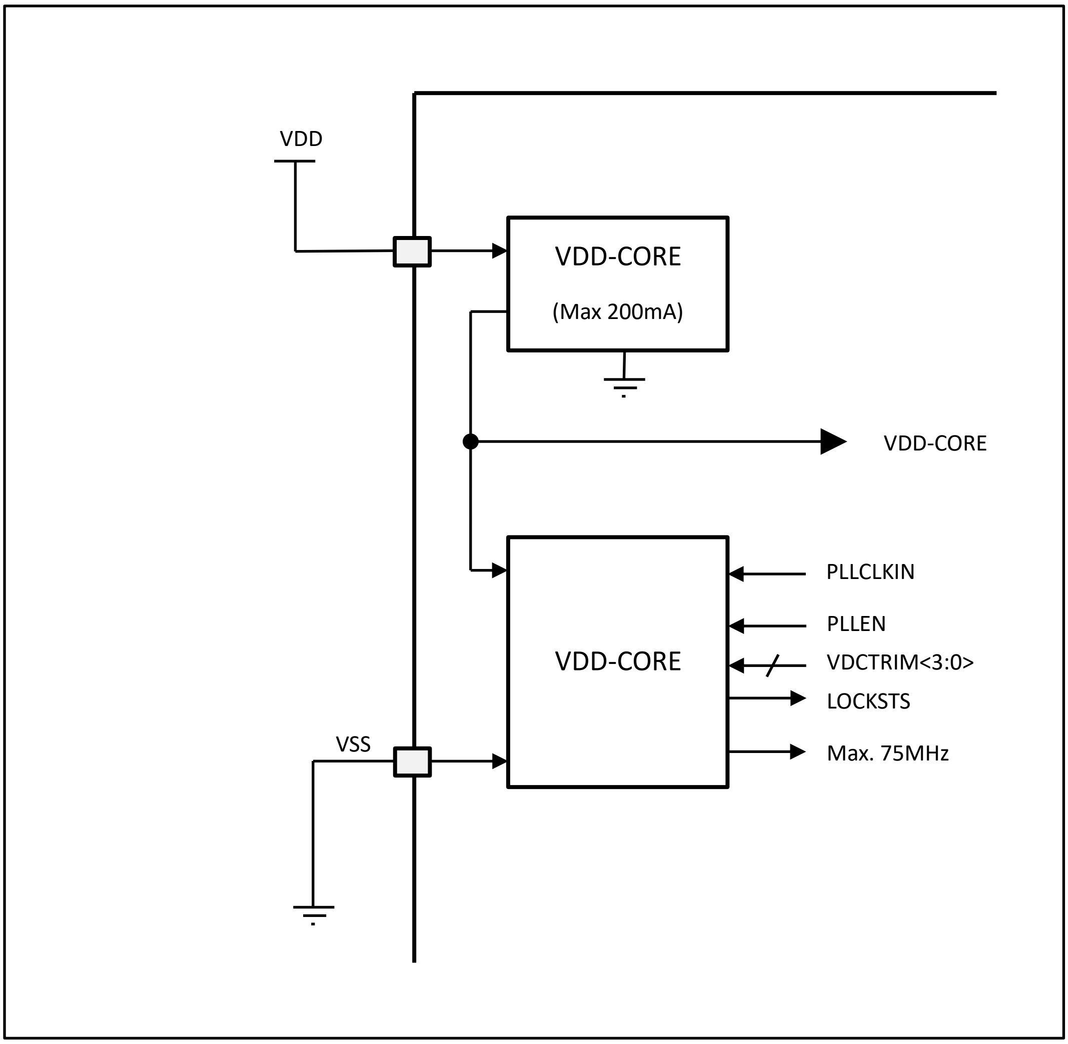


Figure 16.9. The Configuration of PLL

16.3.7 External Main and Sub Oscillator

A33G52x has a 4~10MHz main oscillator and a 32.768kHz suboscillator, which are crystal oscillators and designed for more immunity performance to electrical noises.
The configuration of the main and sub oscillator is shown as follows.

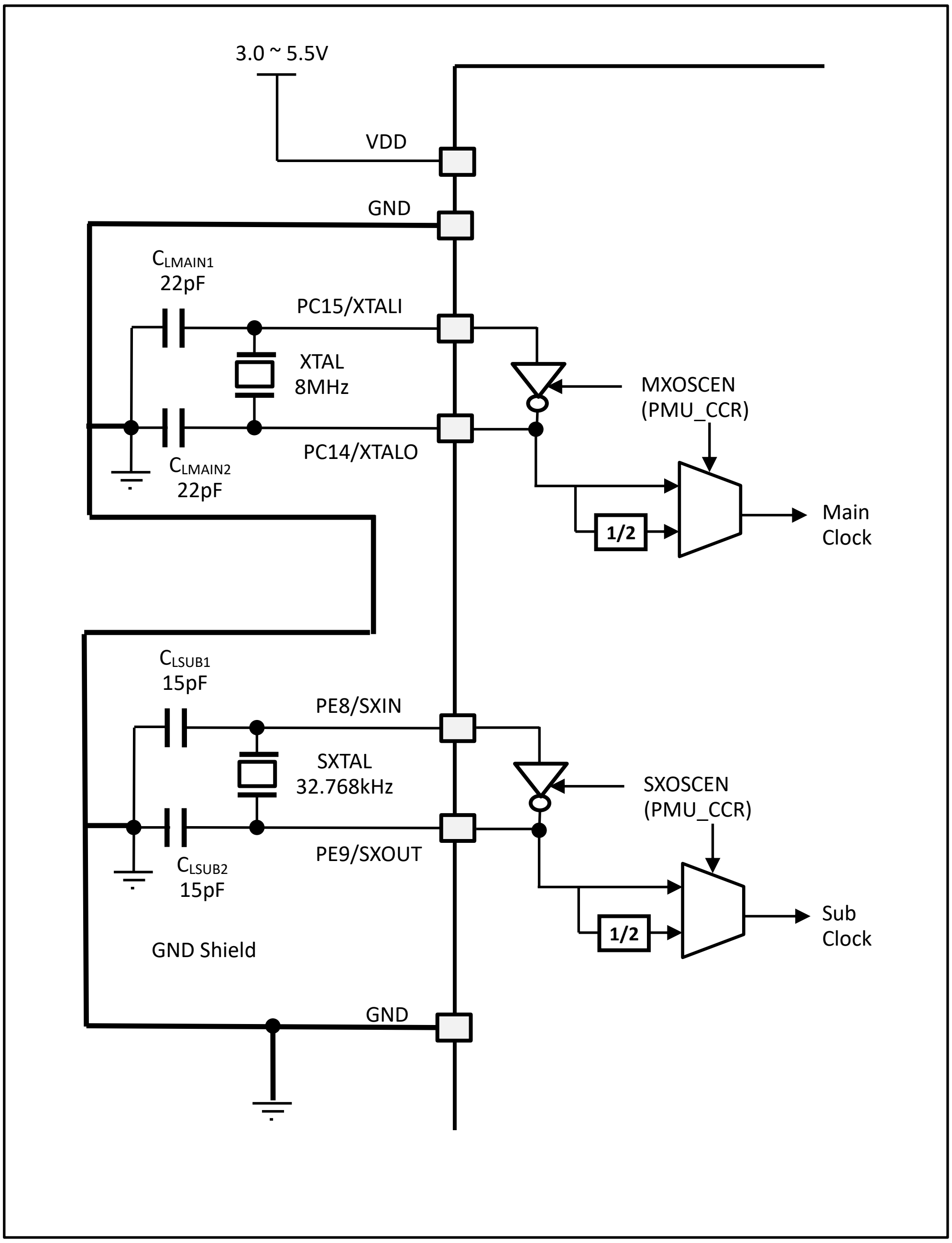


Figure 16.10. The Configuration of Main and Sub Oscillator

16.3.8 Power Configuration

The power configuration of A33G52x is shown as follows.

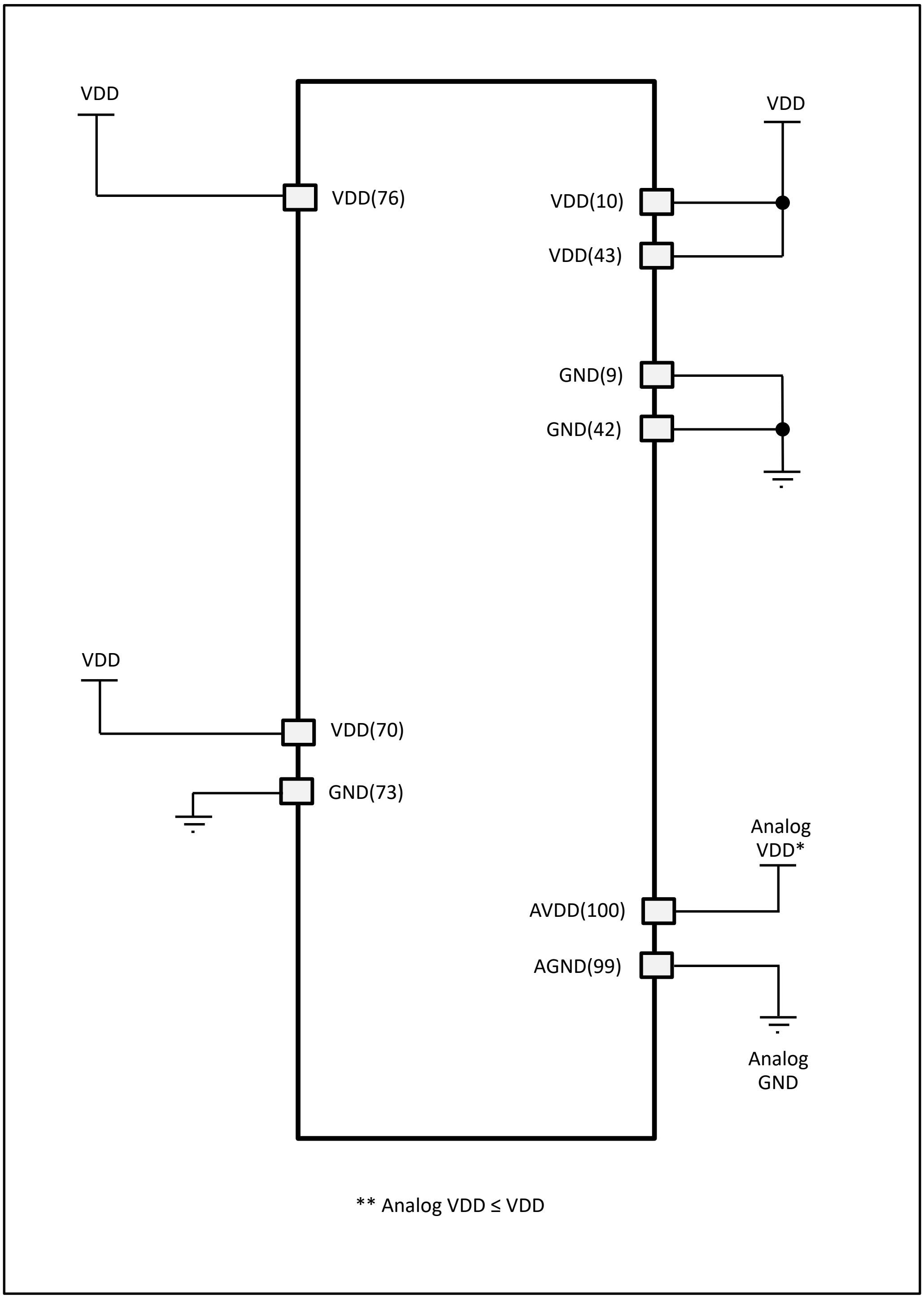


Figure 16.11. The Power Configuration of A33G52x (100MQFP)

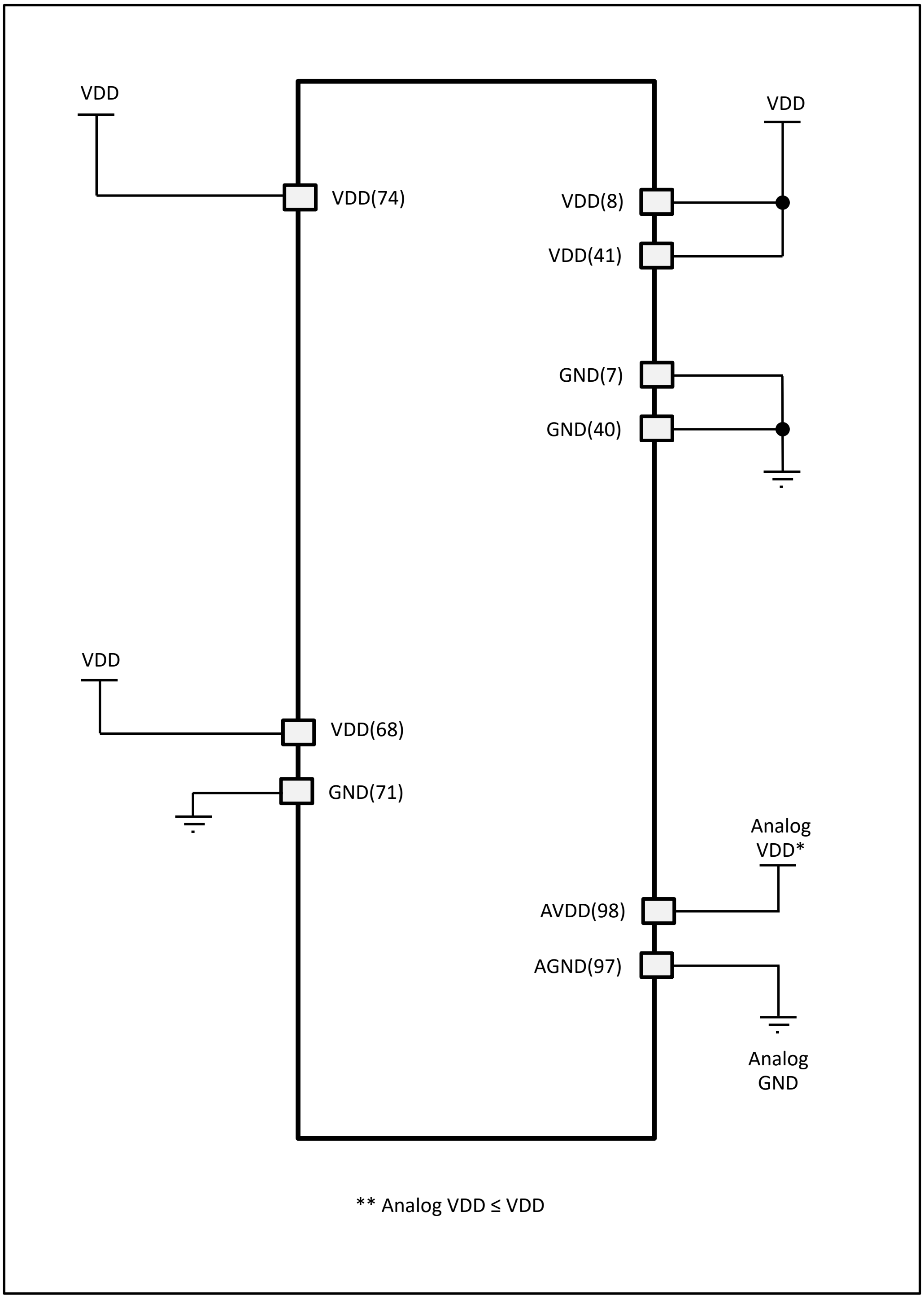
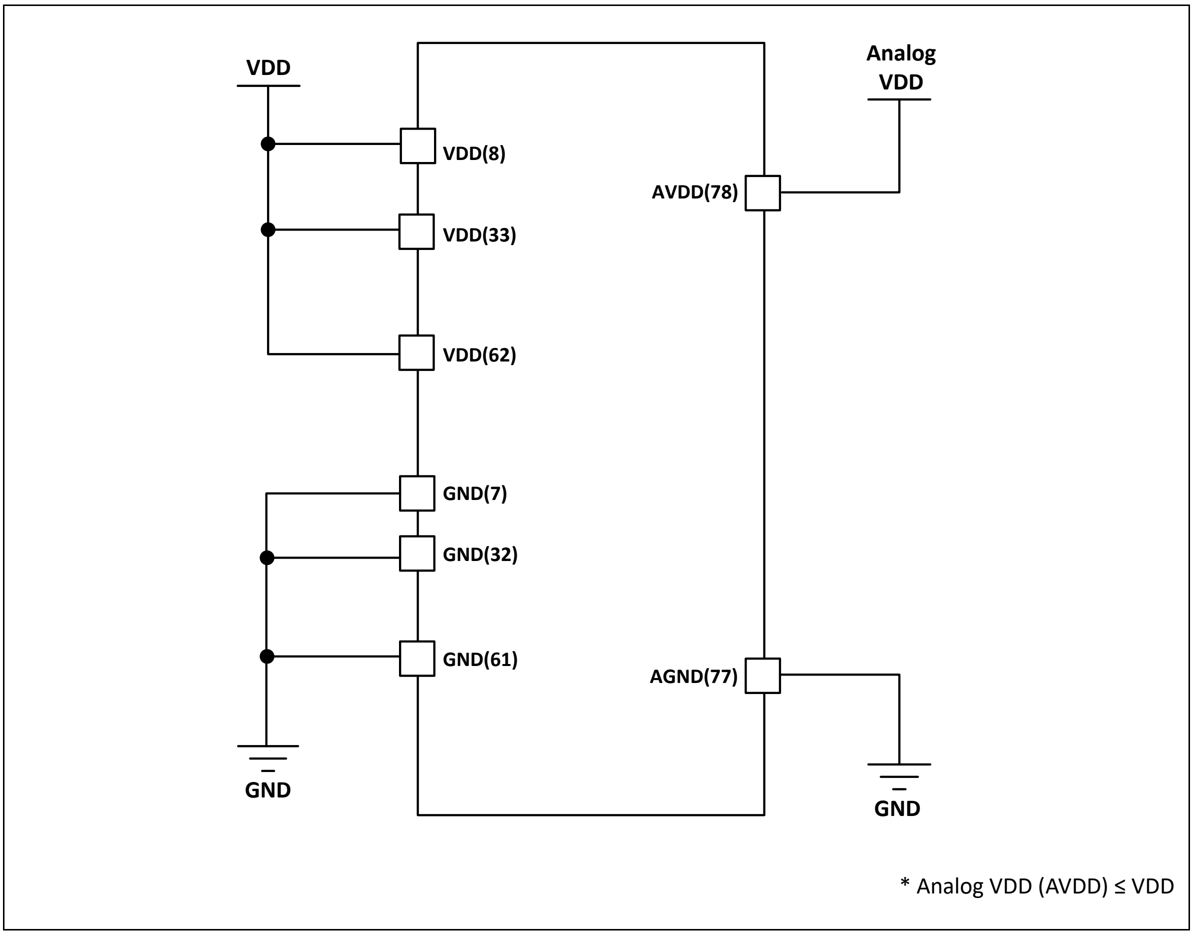
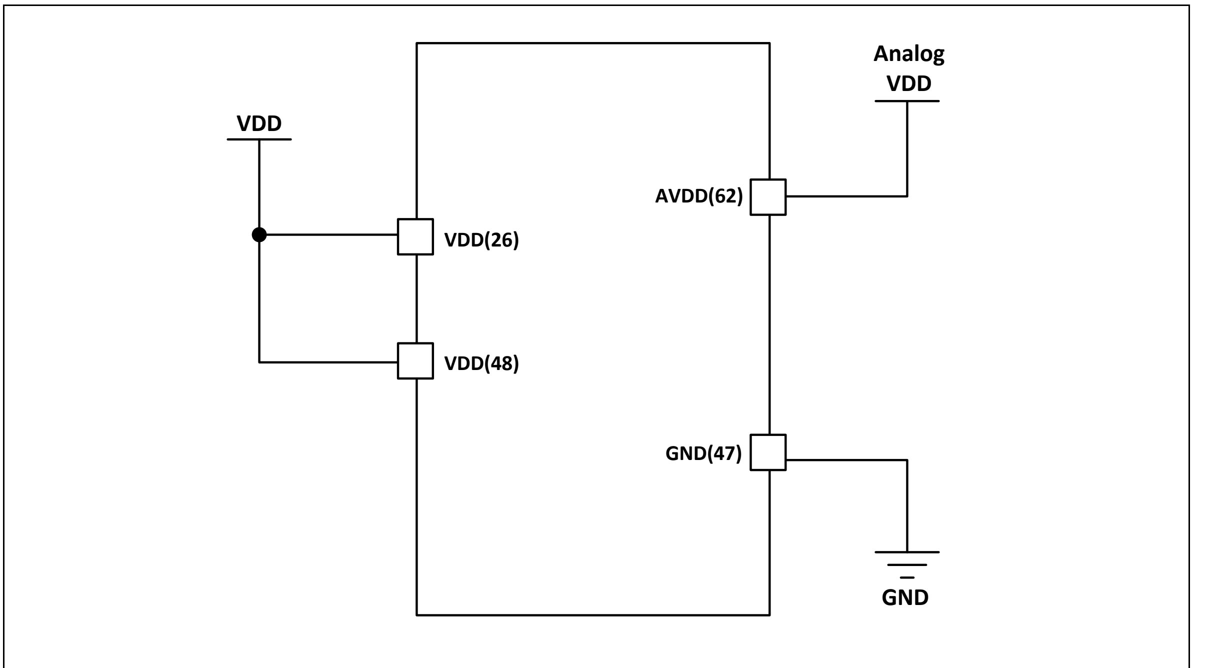


Figure 16.12. The Power Configuration of A33G52x (100LQFP14)



* Analog VDD (AVDD) ≤ VDD

Figure 16.13. The Power Configuration of (80LQFP14/80LQFP12)



* Analog VDD (AVDD) ≤ VDD

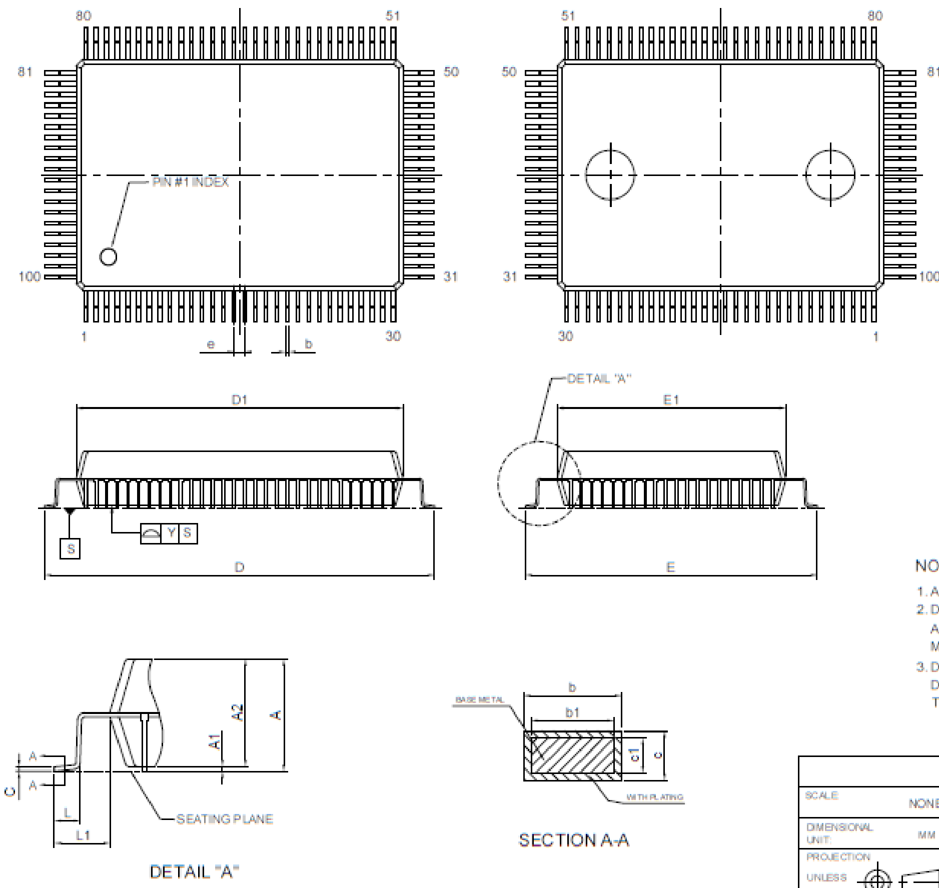
* Analog GND (AGND) is internally connected to GND of chip.

Figure 16.14. The Power Configuration of (64LQFP12/64LQFP10)

CHAPTER 17. Packages

17.1 Package Demension of 100MQFP

REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE	04/11/17	JH. KIM	YJ. CHAE



DETAIL "A"

SECTION A-A

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	3.40
A1	0.05	—	0.15
A2	2.60	—	2.90
b	0.22	—	0.38
b1	0.22	0.30	0.33
c	0.11	—	0.23
c1	0.11	—	0.17
D	23.65	23.90	24.15
D1	19.90	20.00	20.10
E	17.65	17.90	18.15
E1	13.90	14.00	14.10
e	0.65 BSC		
L	0.73	0.88	1.03
L1	1.95 REF		
R1	0.13	—	0.30
R2	0.13	—	0.30
Y	—	—	0.10
θ	0°	—	7°

NOTES:

1. ALL DIMENSIONS IN MILLIMETERS.
2. DIMENSION D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE D1 AND E1 ARE MATIMUM PLASTIC BODY SIZE DIMENSION 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.

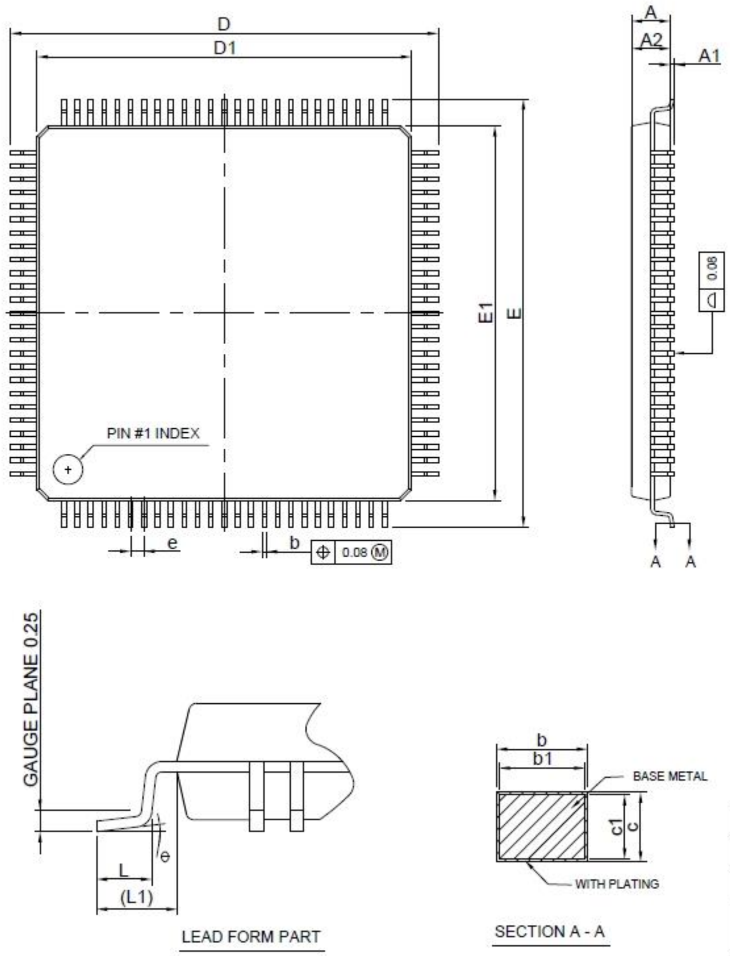
LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 04/11/17	DRAWN: JS. KANG	TITLE: MQFP 14X20MM 100LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS FRAC: .XX .XXX .XXXX ANGLE	ENGINEER: JH. KIM	
PROJECTION UNLESS SPECIFIED		CHECKED: YJ. CHAE	
		APPROVED: YJ. CHAE	
ABOV SEMICONDUCTOR		CAD NAME: OMQ-1420100	DRAWINGNUMBER: OMQ-1420100 REV: — SHEET: 1 OF 1

Figure 17.1. Package Dimension (100MQFP)

17.2 Package Demension of 100LQFP14

REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE.	06/12/'17	JH. KIM	YJ. CHAE



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	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	15.80	16.00	16.20
D1	13.80	14.00	14.20
E	15.80	16.00	16.20
E1	13.80	14.00	14.20
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-BED.
2. DIMENSION 'D1' AND 'E1' DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25MM PER SIDE. 'D1' AND 'E1' ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.
3. DIMENSION 'b' DOES NOT INLCUDE 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.

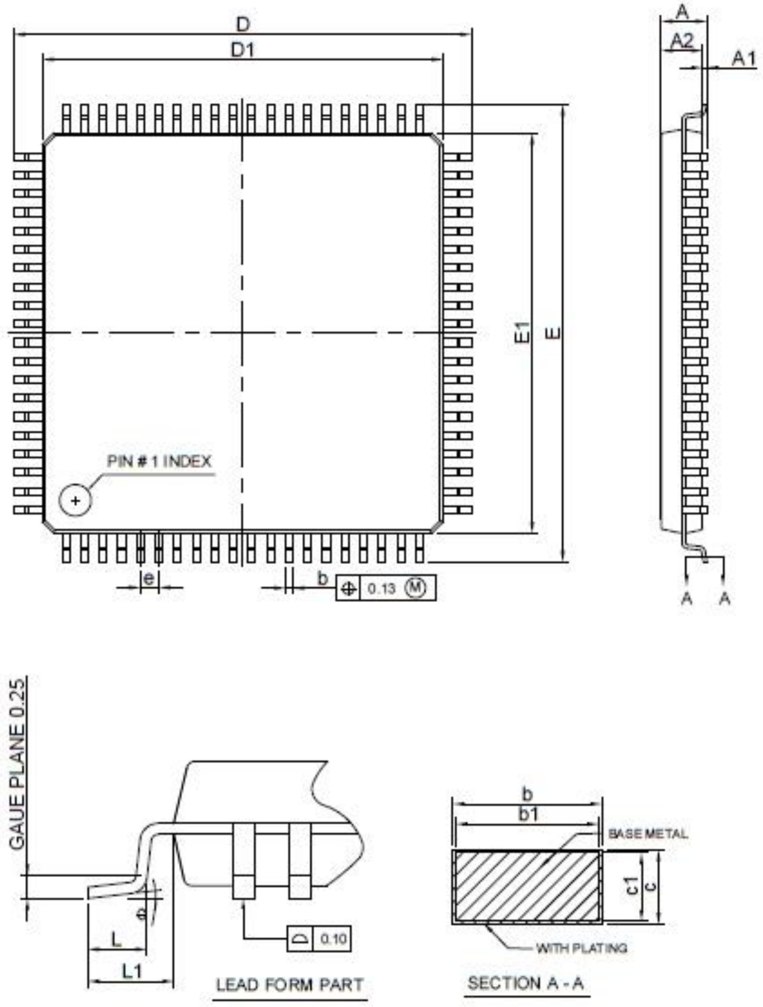
LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 06/12/17	DRAWN: JS. KANG	TITLE: LQFP 14X14MM 100LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS FRAC: .XX .XXX .XXXX ANGLE	ENGINEER: JH. KIM	
PROJECTION UNLESS SPECIFIED		CHECKED: YJ. CHAE	
		APPROVED: YJ. CHAE	
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1414100	DRAWING NUMBER: OLQ-1414100 · REV.: — · SHEET: 1 OF 1

Figure 17.2. Package Dimension (100LQFP)

17.3 Package Demension of 80LQFP14

REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE	04/18/'17	JH. KIM	YJ. CHAE



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.22	0.32	0.38
b1	0.22	0.30	0.33
c	0.09	—	0.20
c1	0.09	—	0.16
D	15.80	16.00	16.20
D1	13.80	14.00	14.20
E	15.80	16.00	16.20
E1	13.80	14.00	14.20
e		0.65 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-BEC
2. DIMENSION 'D1' AND 'E1' DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25MM PER SIDE. 'D1' AND 'E1' ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.
3. DIMENSION 'b' DOES NOT INLCUDE 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.

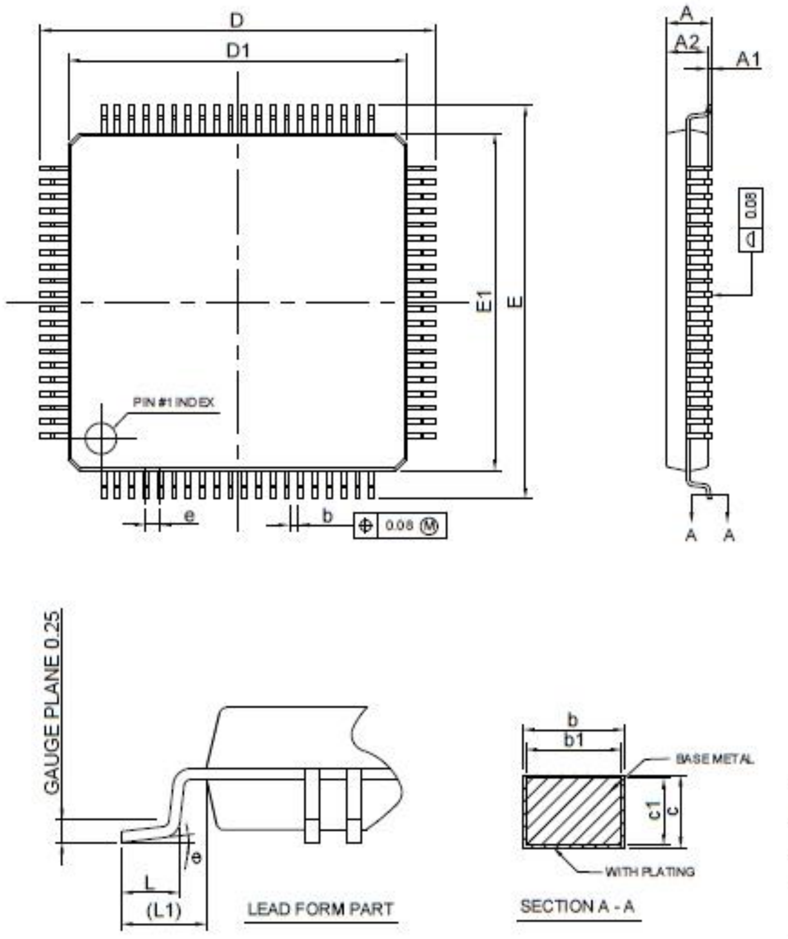
LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE 04/18/'17	DRAWN: JS. KANG	TITLE LQFP 14X14MM 80LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS	ENGINEER: JH. KIM	
PROJECTION UNLESS SPECIFIED	FRAC .XX .XXX .XXXX ANGLE	CHECKED: YJ. CHAE	
		APPROVED: YJ. CHAE	
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1414080	DRAWING NUMBER: OLQ-1414080 REV.: — SHEET: 1 OF 1

Figure 17.3. Package Dimension (80LQFP14)

17.4 Package Demension of 80LQFP12

REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE.	06/12/17	JH. KIM	YJ. CHAE



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	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	13.80	14.00	14.20
D1	11.80	12.00	12.20
E	13.80	14.00	14.20
E1	11.80	12.00	12.20
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
θ	0°	3.5°	7°

NOTES:

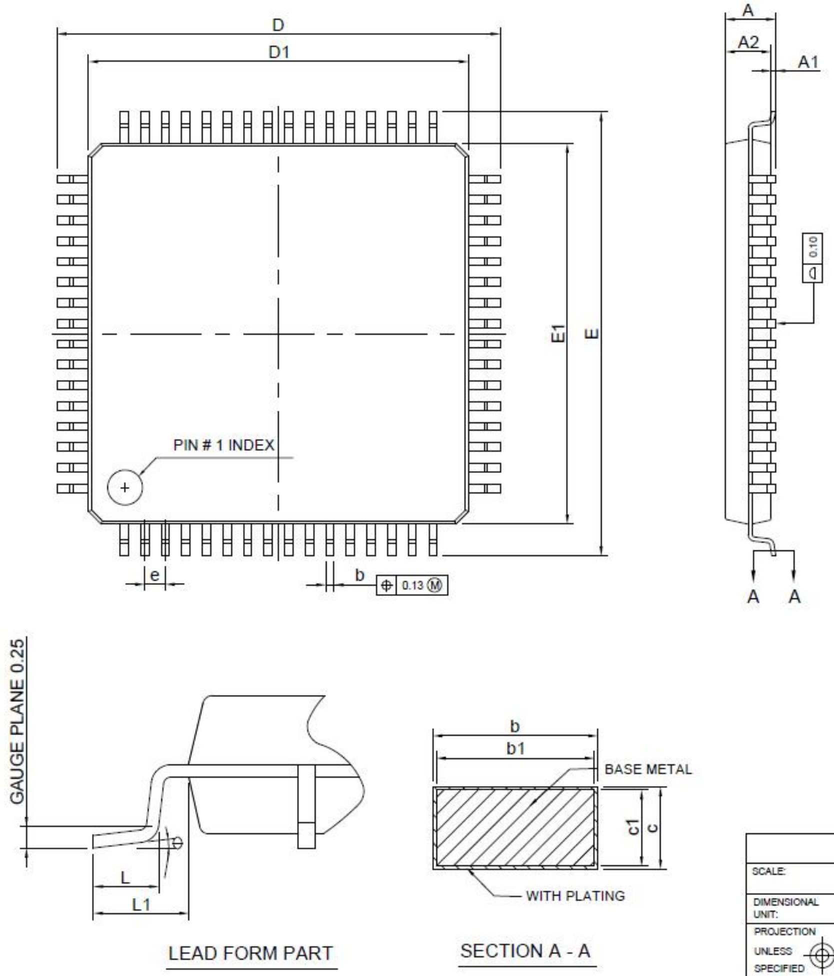
1. ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026-BDD.
2. DIMENSION 'D1' AND 'E1' DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25MM PER SIDE. 'D1' AND 'E1' ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.
3. DIMENSION 'b' DOES NOT INLCUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR 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.

LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 06/12/17	DRAWN: JS. KANG	TITLE: LQFP 12X12MM 80LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS	ENGINEER: JH. KIM	
PROJECTION	FRAC .XX .XXX .XXXX ANGLE	CHECKED: YJ. CHAE	
UNLESS SPECIFIED		APPROVED: YJ. CHAE	
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1212080	DRAWING NUMBER: OLQ-1212080 · REV.: — · SHEET: 1 OF 1

Figure 17.4. Package Dimension (80LQFP12)

17.5 Package Demension of 64LQFP12



REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE.	04/18/'17	JH. KIM	YJ. CHAE

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.22	0.32	0.38
b1	0.25	0.30	0.33
c	0.09	—	0.20
c1	0.09	—	0.16
D	13.80	14.00	14.20
D1	11.80	12.00	12.20
E	13.80	14.00	14.20
E1	11.80	12.00	12.20
e	0.65 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-BDC.
2. DIMENSIONS 'D1' AND 'E1' DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25MM PER SIDE. 'D1' AND 'E1' ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD FLASH.
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 TO THE LOWEST POINT ON THE PACKAGE BODY.

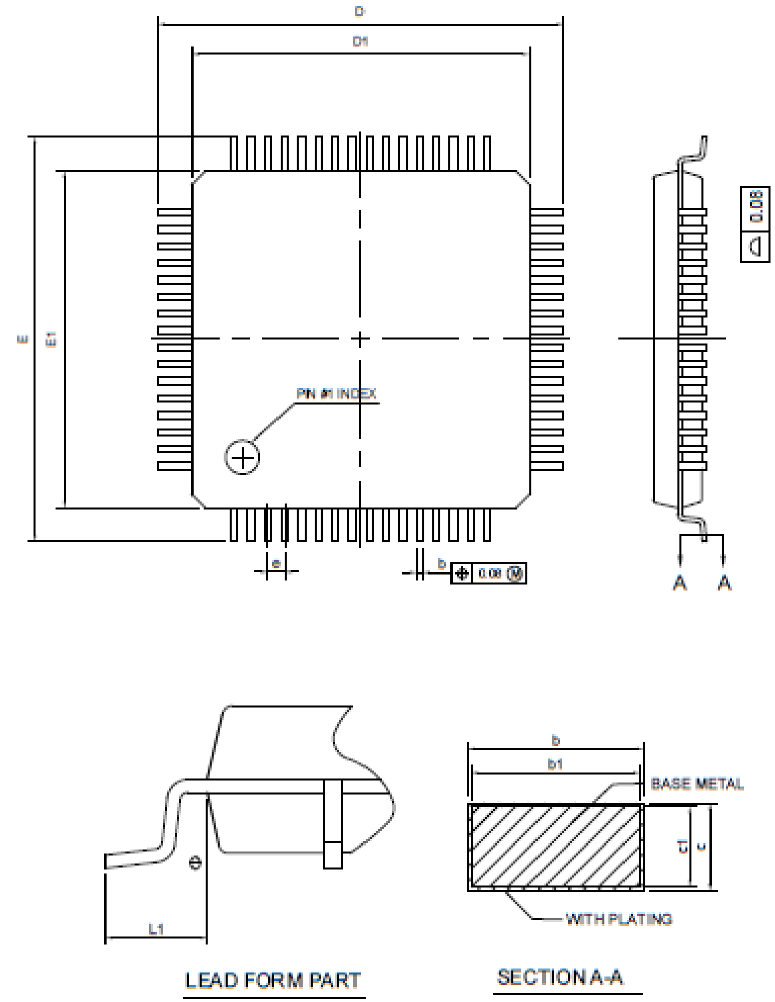
LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 04/18/'17	DRAWN: JS. KANG	TITLE: LQFP 12X12MM 64LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS FRAC: .XX .XXX .XXXX ANGLE	ENGINEER: JH. KIM	
PROJECTION UNLESS SPECIFIED		CHECKED: YJ. CHAE	
		APPROVED: YJ. CHAE	
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1212064	DRAWING NUMBER: OLQ-1212064 REV.: — SHEET: 1 OF 1

Figure 17.5. Package Dimension (64LQFP12)

17.6 Package Demension of 64LQFP10

REV.	DESCRIPTION	DATE	ENG	APPROVAL
—	GENERATE	04/18/'17	JH. KIM	YJ. CHAE



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	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	11.80	12.00	12.20
D1	9.80	10.00	10.20
E	11.80	12.00	12.20
E1	9.80	10.00	10.20
e		0.50 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-BCD.
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 FROMTHE SEATING PLANE TO THE LOWEST POINT ON THE PACKAGE BODY.

LIST OF MATERIAL AND APPLICABLE DOCUMENTS

SCALE: NONE	DATE: 04/18/'17	DRAWN: JS. KANG	TITLE: LQFP 10X10MM 64LD PACKAGE OUTLINE
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS FRAC .X .XX .XXX ANGLE	ENGINEER: JH. KIM	
PROJECTION UNLESS SPECIFIED		CHECKED: YJ. CHAE	
		APPROVED: YJ. CHAE	
ABOV SEMICONDUCTOR		CAD NAME: OLQ-1010064	DRAWING NUMBER: OLQ-1010064 · REV: — · SHEET: 1 OF 1

Figure 17.6. Package Dimension (64LQFP10)

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

ABOV Disclaimer

IMPORTANT NOTICE – PLEASE READ CAREFULLY

ABOV Semiconductor ("ABOV") reserves the right to make changes, corrections, enhancements, modifications, and improvements to ABOV products and/or to this document at any time without notice. ABOV does not give warranties as to the accuracy or completeness of the information included herein. Purchasers should obtain the latest relevant information of ABOV products before placing orders. Purchasers are entirely responsible for the choice, selection, and use of ABOV products and ABOV assumes no liability for application assistance or the design of purchasers' products. No license, express or implied, to any intellectual property rights is granted by ABOV herein. ABOV disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of ABOV products in such unauthorized applications. ABOV and the ABOV logo are trademarks of ABOV. All other product or service names are the property of their respective owners. Information in this document supersedes and replaces the information previously supplied in any former versions of this document.