Datasheet

Three-phase Motor Control MCU
FU6832N1

Fortior Technology (Shenzhen) Co., Ltd

Contents

Explanation of Symbols

- The symbol “[ ]” following a register indicates a bit in the register. For example, ABCD[XY] indicates the XY bit in ABCD register.
- The symbol "x" in a register name indicates similar registers. For example, TIMx_CR0 indicates TIM3_CR0 and TIM4_CR0.
- [m:n] indicates a range of bits. For example, [3:0] means the bits from bit3 to bit0.
- Pm.n indicates the nth port of the Portm. P0.0 indicates the 0th port of Port0.
- Register read and write symbols:
  - R: Read only
  - W: Write only
  - R/W: Read/write
  - W0: Only 0 can be written
  - W1: Only 1 can be written
- The symbol “-” indicates an uncertainty value or invalid value.
- The RMW instruction cannot be used for registers with different read and written representations.
- Q (number) format is to store floating-point numbers using fixed-point numbers. MSB is the sign bit, followed by integer bits and fraction bits, where lower Q bits are assigned to the fractional part and the remaining bits are assigned to the integer part. For example, for Q12, bit15 is the sign bit, bit14 ~ bit12 represent the integer part and bit11 ~ bit0 represent the fraction part. The Q12 format has a decimal range -8 ~ 7.9998 (corresponding to 0x8000 ~ 0x7FFF).

Abbreviations

ADC: Analog to Digital Convertor

BEMF: Back Electromotive Force

BLDC: Brushless DC Motor

CRC: Cyclic Redundancy Check

DAC: Digital to Analog Convertor

DMA: Direct Memory Access

FG: Frequency Generator

FICE: Fortior Interactive Connectivity Establishment

FOC: Field Oriented Control

FOSC: Fast Oscillator

GPIO: General Purpose Input Output

I^2C: Inter Integrated Circuit

IC: Integrated Circuit

IRAM: Internal RAM

IDE: Integrated Development Environment

LDO: Low Dropout Regulator

LPF: Low Pass Filter

LVD: Low Voltage Detection

MDU: Multiplication Division Unit

ME: Motor Engine

MSB: Most Significant Bit

MOSFET: Metal Oxide Semiconductor Field Effect Transistor

NC: Not Connected

PGA: Programmable Gain Amplifier

PI/PID: Proportional Integral/Proportional Integral Derivative

PLL: Phase Locked Loop

PWM: Pulse Width Modulation

QEP: Quadrature Encoder Pulse

RAM: Random Access Memory

RMW: Read Modified Write

ROM: Read Only Memory

RSD: Rotating State Detection

RTC: Real Time Clock

SCL: Serial Clock Line

SDA: Serial Data Line

SFR: Special Function Register

SMO: Sliding Mode Observer

SOSC: Slow Oscillator

SPI: Serial Peripheral Interface

SVPWM: Space Vector PWM

TSD: Temperature Sensor Detect

UART: Universal Asynchronous Receiver/Transmitter

WDT: Watch Dog Timer

XRAM: External RAM

XSFR: External SFR

1 System Introduction

1.1 Features

- Power supply:
  - High-voltage single-power supply mode: When VCC_MODE = 0, external power supply 5V~18V is connected to VCC pin, and internal LDO supplies VDD5 voltage.
  - Low-voltage single-power supply mode: When VCC_MODE = 1, external power supply 3V~5.5V is connected to VDD5 pin, and VDD5 pin is shorted to VCC pin.
  - Dual-power supply mode: When VCC_MODE = 1, external power supply 1 (5V~18V) is connected to VCC pin, and external power supply 2 (5V) is connected to VDD5 pin.
- Dual core: 8051 core and ME core
- An instruction cycle mostly takes 1 or 2 system clock cycle(s)
- 16kB Flash ROM with CRC, self-program and code protection
- 256 bytes IRAM and 768 bytes XRAM
- ME: Core integrating PID module, FOC module, MDU auxiliary computing module and LPF module
- 16 interrupt sources with 4 configurable priority levels
- 22*GPIOs
- Timers:
  - Timer1: Timer supporting square-wave drive timing control, automatic commutation, cycle-by-cycle current limiting and Hall/BEMF-based position sensing
  - Timer2: Timer supporting PWM output, measurement of duty cycle and period of input PWM wave, measurement of the time of set PWM wave numbers, QEP decoding, tailwind/headwind detection, and rotation direction and speed detection of step motor
  - Timer3/Timer4: Timers supporting PWM output, and measurement of duty cycle and period of input PWM wave Timer4 supports FG generation and Timer3 supports up to 48MHz input
  - Systick Timer
  - RTC
- Communication interfaces:
  - 1*SPI
  - 1*I^2C
  - 2*UARTs, supporting single-wire mode
- Dual-channel DMA: supporting data transmission via I^2C/SPI/UART
- Analog Peripherals:
  - 12-bit ADC, operating with 1μs conversion time and VDD5 selectable as reference voltage
  - Number of ADC channels: 11

- Internal VHALF, with 1/2 VREF as the internal reference
  - 3 standalone operational amplifiers, where the gain of AMP0 is configurable
  - 3-channel analog comparator
  - DAC: Single-channel 9-bit, single-channel 6-bit
- Built-in MOSFET driver: 3P3N pre-driver output
- FOC module supports single/dual/triple-shunt current sampling
- System clock
  - Built-in 24MHz ± 2% high-speed oscillator
  - Built-in 32.8kHz low-speed oscillator
- WDT
- LVD
- TSD
- Two-wire FICE protocol based in-circuit emulation
- AEC-Q100 Certification (Grade 1)

1.2 Applications

The chip can be used for motor control in automobile electronics, including water pumps, wipers, seat ventilators, active air intake grilles, air-conditioning outlets, automotive valves, etc.

1.3 Overview

The high-performance motor drive chip incorporates ME core and 8051 core. ME core integrates FOC, MDU, LPF, PID and SVPWM modules that allow for automatic calculation of FOC or square-wave control by the hardware for sensored/sensorless BLDC/PMSM motors. 8051 core is used for parameter configuration and routine processing. Most of 8051 core instruction cycle takes 1T or 2T clock cycle(s). The dual cores work in parallel to achieve high-performance motor control. The chip integrates high-speed operational amplifiers, comparators, high-speed ADC, CRC, SPI, I^2C, UART, Timers, built-in high-voltage LDO, which are suitable for FOC or square-wave based BLDC/PMSM motors.

1.4 Functional Block Diagram

1.4.1 FU6832N1

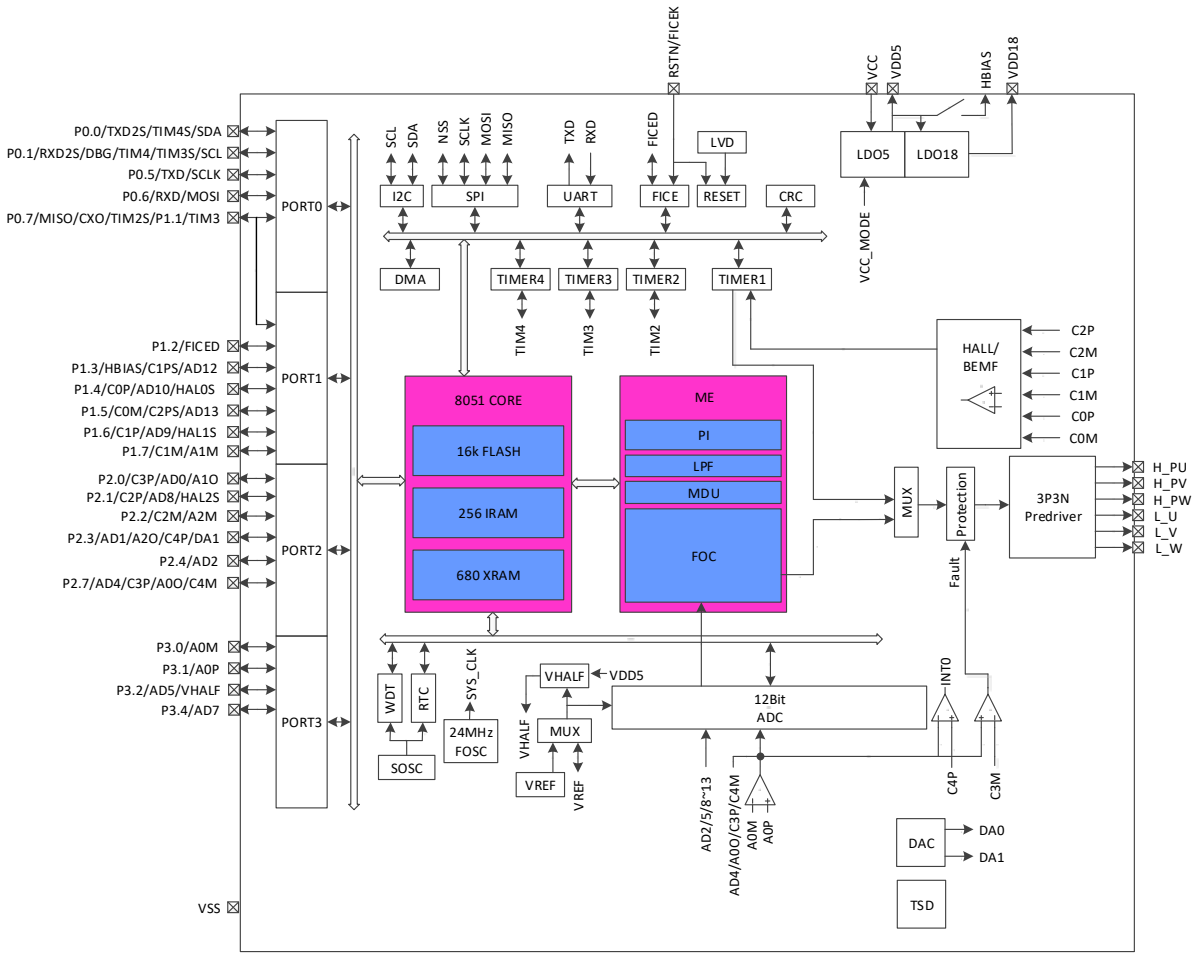


Figure 1-1 FU6832N1 Functional Block Diagram

1.5 Memory Organization

The internal storage space is divided into Program Memory and Data Memory, which are independently addressed.

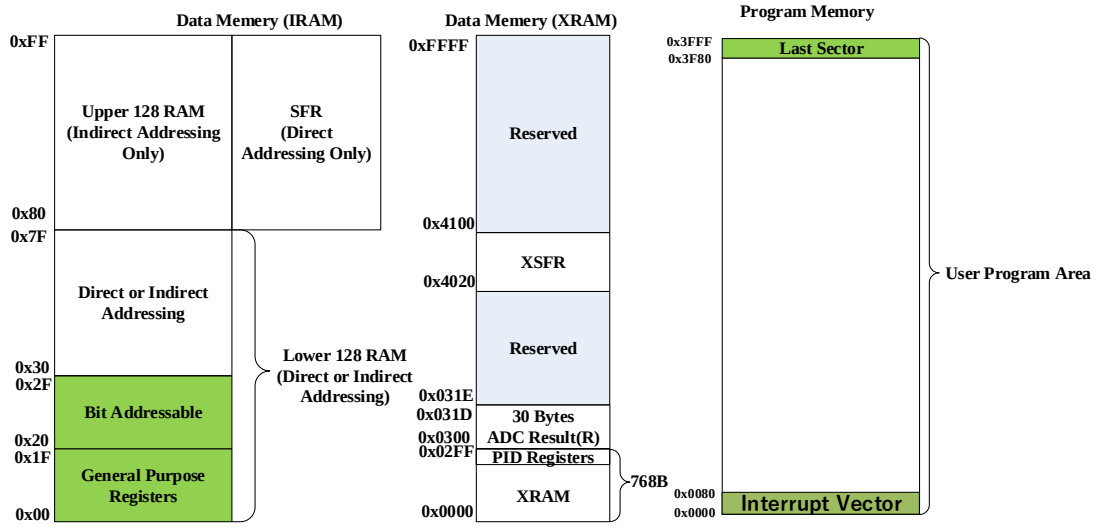


Figure 1-2 Memory Organization

1.5.1 Program Memory

The chip implements this program memory as Flash memory with a block from addresses 0x0000 to 0x3FFF to store the control program.

The first sector (0x0000 ~ 0x007F) is the interrupt vector address area, which is used to store the start address of each interrupt subroutine. The last sector (0x3F80 ~ 0x3FFF) contains internal control bits of the chip.

1.5.2 Data Memory

The data memory is divided into External Data Memory and Internal Data Memory, as shown in Figure 1-2.

The External Data Memory is addressed in the range from 0x0000 to 0xFFFF, which can be accessed only with MOVX instructions. It comprises XRAM (0x0000 ~ 0x02A7), extended control register space (0x02A8 ~ 0x02EF, 0x4020 ~ 0x40FF) and ADC result memory area (0x0300 ~ 0x031D). If PI/PID module is disabled, XRAM space available for user programs is 768Bytes, and if PI/PID module is enabled, XRAM space available for user programs is 680Bytes.

The Internal Data Memory is addressed from 0x00 to 0xFF. Locations 0x00 ~ 0x1F are addressable as 4 banks of general purpose registers, each bank consisting of 8 registers, adding up to 32 registers. Locations 0x20 ~ 0x7F are used for general purpose RAM memory, supporting direct and indirect addressing. Locations 0x20 ~ 0x2F are 16-bit addressable. When locations 0x80 ~ 0xFF are accessed by indirect addressing, it points to RAM. When locations 0x80 ~ 0xFF are accessed by direct addressing, it points to SFR.

1.5.3 SFR

Table 1-1 SFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xF8	DRV_OUT	PI_CR			P0_OE	P1_OE	P2_OE	P3_OE
0xF0	B							
0xE8	P4	P4_OE						
0xE0	ACC	CMP_CR4						
0xD8	IP3	EVT_FILT	CMP_CR2	LVSR	CMP_CR3			
0xD0	PSW	P1_IE	P1_IF	P2_IE	P2_IF	CMP_CR0	CMP_CR1	CMP_SR
0xC8	IP2	RST_SR	MDU_MD	MDU_D				
0xC0	IP1	MDU_CR	MDU_CL	MDU_CH	MDU_BL	MDU_BH	MDU_AL	MDU_AH
0xB8	IP0							
0xB0	P3							
0xA8	IE	TIM2_CR1	TIM2__CNTRL	TIM2__CNTRH	TIM2__DRL	TIM2__DRH	TIM2__ARRL	TIM2__ARRH
0xA0	P2	TIM2_CR0	TIM3__CNTRL	TIM3__CNTRH	TIM3__DRL	TIM3__DRH	TIM3__ARRL	TIM3__ARRH
0x98	UT_CR	UT_DR	UT_BAUDL	UT_BAUDH	TIM3_CR0	TIM3_CR1	TIM4_CR0	TIM4_CR1
0x90	P1		TIM4__CNTRL	TIM4__CNTRH	TIM4__DRL	TIM4__DRH	TIM4__ARRL	TIM4__ARRH
0x88	TCON	UT2_DR	UT2_CR					
0x80	P0	SP	DPL	DPH	FLA_KEY	FLA_CR		PCON

Notes:

- Registers containing the symbol "__" are 16-bit snapshot registers. Snapshot registers are the dynamic registers which shall be read using variables. The value will be incorrect when the register is read directly.
- 8-bit MCU shall read a 16-bit register twice to get the value, the 8 high-order bits and the 8 low-order bits respectively. The result will be incorrect when 8 low-order bits of the register change after MCU has read the 8 high-order bits. Therefore, when 8 high-order bits of the snapshot register are read by MCU, the corresponding 8 low-order bits are stored and read.
- Snapshot register must be read as a whole, the 8 high-order bits first and then the 8 low-order bits.

1.5.4 XSFR

Table 1-2 XSFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x40D8	FOC__POWH	FOC__POWL	FOC__IAMAXH	FOC__IAMAXL	FOC__IBMAXH	FOC__IBMAXL	FOC__ICMAXH	FOC__ICMAXL
	FOC_EOMEKLPF							
0x40D0	FOC__EALPH	FOC__EALPL	FOC__EBETH	FOC__EBETL	FOC__EOMEH	FOC__EOMEL	FOC__UQEXH	FOC__UQEXL
							FOC_KFGH	FOC_KFGL
0x40C8	FOC__IBH	FOC__IBL	FOC__IAH	FOC__IAL	FOC__THETAH	FOC__THETAL	FOC__ETHETAH	FOC__ETHETAL
0x40C0	FOC__IBETH	FOC__IBETL	FOC__VBETH	FOC__VBETL	FOC__VALPH	FOC__VALPL	FOC__ICH	FOC__ICL
			FOC_UDCPSH	FOC_UDCPSL	FOC_UQCPSH	FOC_UQCPSL		
0x40B8	FOC__UDH	FOC__UDL	FOC__UQH	FOC__UQL	FOC__IDH	FOC__IDL	FOC__IQH	FOC__IQL
0x40B0	FOC_DMAXH	FOC_DMAXL	FOC_DMINH	FOC_DMINL	FOC_QMAXH	FOC_QMAXL	FOC_QMINH	FOC_QMINL
0x40A8	FOC__RTHESTEPH	FOC__RTHESTEPL	FOC_RTHEACCH	FOC_RTHEACCL	FOC_RTHECNT	FOC_THECOR	FOC_THECOMPH	FOC_THECOMPL
			FOC_EOMELPFH	FOC_EOMELPFL		CMP_SAMR		
0x40A0	FOC_CR1	FOC_CR2	FOC_TSMIN	FOC_TGLI	FOC_TBLO	FOC_TRGDLY	FOC_CSOH	FOC_CSOL
0x4098	FOC__UDCFLTH	FOC__UDCFLTL						FOC_CR0
	TIM1__ITRIPH	TIM1__ITRIPL						
0x4090	FOC_IDREFH	FOC_IDREFL	FOC_IQREFH	FOC_IQREFL	FOC_DQKPH	FOC_DQKPL	FOC_DQKIH	FOC_DQKIL
	TIM1__URESH	TIM1__URESL	TIM1__UIGNH	TIM1__UIGNL	TIM1__KFH	TIM1__KFL	TIM1__KRH	TIM1__KRL
0x4088	FOC_EK3H	FOC_EK3L	FOC_EK4H	FOC_EK4L	FOC_EK1H	FOC_EK1L	FOC_EK2H	FOC_EK2L
	TIM1__RARRH	TIM1__RARRL	TIM1__RCNTRH	TIM1__RCNTRL	TIM1__UCOPH	TIM1__UCOPL	TIM1__UFLPH	TIM1__UFLPL
0x4080	FOC_FBASEH	FOC_FBASEL	FOC_EFREQACCH	FOC_EFREQACCL	FOC_EFREQMINH	FOC_EFRQMINL	FOC_EFREQHOLDH	FOC_EFREQHOLDL
	TIM1_DBR7H	TIM1_DBR7L	TIM1__BCNTRH	TIM1__BCNTRL	TIM1__BCCRH	TIM1__BCCRL	TIM1__BARRH	TIM1__BARRL
0x4078	FOC_KSLIDEH	FOC_KSLIDEL	FOC_EKLPFMINH	FOC_EKLPFMINL	FOC_EBMFKH	FOC_EBMFKL	FOC_OMEKLPFH	FOC_OMEKLPFL
	TIM1_DBR3H	TIM1_DBR3L	TIM1_DBR4H	TIM1_DBR4L	TIM1_DBR5H	TIM1_DBR5L	TIM1_DBR6H	TIM1_DBR6L
0x4070	TIM1_BCORH	TIM1_BCORL			FOC_EKPH	FOC_EKPL	FOC_EKIH	FOC_EKIL
					TIM1_DBR1H	TIM1_DBR1L	TIM1_DBR2H	TIM1_DBR2L
0x4068	TIM1_CR0	TIM1_CR1	TIM1_CR2	TIM1_CR3	TIM1_CR4	TIM1_ IER	TIM1_SR	

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x4060	DRV_DTR	DRV_SR	DRV_CR		SYST_ARRH	SYST_ARRL	DRV__CNTRH	DRV__CNTRL
0x4058	DRV_DRH	DRV_DRL	DRV_COMRH	DRV_COMRL	DRV_CMRH	DRV_CMRL	DRV_ARRH	DRV_ARRL
0x4050	P1_AN	P2_AN	P3_AN	P0_PU	P1_PU	P2_PU	P3_PU	P4_PU
0x4048			DAC1_DR	DAC_DR	PH_SEL	PH_SEL1	AMP_CR	VREF_VHALF_CR
0x4040	DMA1_BAH	DMA1_BAL	UT2_BAUDL	UT2_BAUDH	CAL_CR0	CAL_CR1		
0x4038	ADC_SCYC	ADC_CR	DMA0_CR0	DMA1_CR0	DMA0_LEN	DMA1_LEN	DMA0_BAH	DMA1_BAL
0x4030	SPI_CR0	SPI_CR1	SPI_CLK	SPI_DR	AMP0_GAIN	DAC_CR	ADC_MASK_SYSCH	ADC_MASK_SYSCL
0x4028	I2C_CR	I2C_ID	I2C_DR	I2C_SR	RTC_TMH	RTC_TML	RTC_STA	TSD_CR
0x4020		CRC_DIN	CRC_CR	CRC_DR	CRC_BEG	CRC_CNT	WDT_CR	WDT_ARR
0x4018								
0x4010								
0x4008								
0x4000								
0x0318	AD12_DRH	AD12_DRL	AD13_DRH	AD13_DRL	AD14_DRH	AD14_DRH		
0x0310	AD8_DRH	AD8_DRL	AD9_DRH	AD9_DRL	AD10_DRH	AD10_DRL	AD11_DRH	AD11_DRL
0x0308	AD4_DRH	AD4_DRL	AD5_DRH	AD5_DRL	AD6_DRH	AD6_DRL	AD7_DRH	AD7_DRL
0x0300	AD0_DRH	AD0_DRL	AD1_DRH	AD1_DRL	AD2_DRH	AD2_DRL	AD3_DRH	AD3_DRL
0x02F8								
0x02F0								
0x02E8	PI0_EK1		PI0_EK		PI0_UKH		PI0_UKL	
0x02E0	PI0_KP		PI0_KI		PI0_UKMAX		PI0_UKMIN	
0x02D8	PI1_EK1		PI1_EK		PI1_UKH		PI1_UKL	
0x02D0	PI1_KP		PI1_KI		PI1_UKMAX		PI1_UKMIN	
0x02C8	PI2_UKH		PI2_UKL		PI2_KD		PI2_EK2	
0x02C0	PI2_UKMAX		PI2_UKMIN		PI2_EK1		PI2_EK	
0x02B8	PI3_KD		PI3_EK2		PI2_KP		PI2_KI	
0x02B0	PI3_EK1		PI3_EK		PI3_UKH		PI3_UKL	
0x02A8	PI3_KP		PI3_KI		PI3_UKMAX		PI3_UKMIN	

Notes:

- Registers containing the symbol "__" are 16-bit snapshot registers. Snapshot registers are the dynamic registers which shall be read using variables. The value will be incorrect when the register is read directly.
- 8-bit MCU shall read a 16-bit register twice to get the value, the 8 high-order bits and the 8 low-order bits respectively. The result will be incorrect when 8 low-order bits of the register change after MCU has read the 8 high-order bits. Therefore, when 8 high-order bits of the snapshot register are read by MCU, the corresponding 8 low-order bits are stored and read.
- Snapshot register must be read as a whole, the 8 high-order bits first and then the 8 low-order bits.

2 Pin Definitions

The IO types are defined as follows:

- DI = Digital Input
- DO = Digital Output
- DB = Digital Bidirectional
- AI = Analog Input
- AO = Analog Output
- P = Power Supply

2.1 FU6832N1 QFN32 Pins

Table 2-1 FU6832N1 QFN32 Pins

Pin	FU6832N1 QFN32	IO Type	Description
P2.1/ C2P/ A2P/ AD8/ HAL2S	1	DB/ AI/ AI/ AI/ DI	GPIO, configurable as INT1 input CMP2 positive input AMP2 positive input Input of ADC channel 8 Hall-IC2 logic level input after function switching
P2.2/ C2M/ A2M	2	DB/ AI/ AI	GPIO, configurable as INT1 input CMP2 negative input AMP2 negative input
P2.3/ AD1/ A2O/ C4P/ DA1	3	DB/ AI/ AO/ AI/ AO	GPIO, configurable as INT1 input Input of ADC channel 1 AMP2 output CMP4 positive input DAC1 output, without Buffer output
P2.4/ AD2	4	DB/ AI	GPIO, configurable as INT1 input Input of ADC channel 2 for bus voltage sampling
P2.7/ AD4/ C3P/ A0O/ C4M	5	DB/ AI/ AI/ AO/ AI	GPIO, configurable as INT1 input Input of ADC channel 4 for bus current sampling CMP3 positive input AMP0 output CMP4 negative input
P3.0/ A0M	6	DB/ AI	GPIO AMP0 negative input
P3.1/ A0P	7	DB/ AI	GPIO AMP0 positive input
P3.2/ AD5/ VHALF	8	DB/ AI/ AO	GPIO Input of ADC channel 5 VREF/2 output with an external 1μF capacitor
P3.4/ AD7	9	DB/ AI	GPIO Input of ADC channel 7
P0.0/ TIM4S TXD2S/ SDA	10	DB/ DB/ DO/ DB	GPIO, configurable as INT0 input Timer4 input/output after function switching UART2 TXD output after function switching I^2C SDA, configured as collector open-drain output
P0.1/ RXD2S/ DBG/	11	DB/ DB/ DO/	GPIO, configurable as INT0 input UART2 RXD input in two-wire mode or TXD output/RXD input in single-wire mode after function switching Output of debug signal

Pin	FU6832N1 QFN32	IO Type	Description
TIM4/ TIM3S/ SCL		DB/ DB/ DB	Timer4 input/output Timer3 input/output after function switching I^2C SCL, configured as collector open-drain output
P0.5/ TXD/ SCLK	12	DB/ DO/ DB	GPIO, configurable as INT0 input UART1 TXD output SPI SCLK
P0.6/ RXD/ MOSI	13	DB/ DB/ DB	GPIO, configurable as INT0 input UART1 RXD input in two-wire mode or TXD output/RXD input in single-wire mode SPI MOSI, master output or slave input
P0.7/ MISO/ CXO/ TIM2S/ P1.1/ TIM3	14 (P0.7/P1.1)	DB/ DB/ DO/ DB/ DB/ DB	GPIO P0.7 SPI MISO, master input or slave output Test port for comparator output Timer2 input/output after function switching GPIO P1.1, configurable as INT1 input Timer3 input/output
H_PU	15	DO	Pre-driver U-phase high-side output, with built-in 50kΩ pull-up resistor
H_PV	16	DO	Pre-driver V-phase high-side output, with built-in 50kΩ pull-up resistor
H_PW	17	DO	Pre-driver W-phase high-side output, with built-in 50kΩ pull-up resistor
L_U	18	DO	Pre-driver U-phase low-side output, with built-in 25kΩ pull-down resistor
L_V	19	DO	Pre-driver V-phase low-side output, with built-in 25kΩ pull-down resistor
L_W	20	DO	Pre-driver W-phase low-side output, with built-in 25kΩ pull-down resistor
VCC	21	P	Power input. The voltage range is determined by VCC_MODE, with an external filter capacitor of 10μF or larger. ■ High-voltage single-power supply mode: When VCC_MODE = 0, external power supply 5V~18V is connected to VCC pin, and internal LDO supplies VDD5 voltage. ■ Low-voltage single-power supply mode: When VCC_MODE = 1, external power supply 3V~5.5V is connected to VDD5 pin, and VDD5 pin is shorted to VCC pin.
VSS	22	P	Ground
VDD5	23	P	Power input or 5V LDO power output, which is determined by VCC_MODE and provided with an external 1~4.7μF capacitor. See descriptions on VCC pin for more details.
RSTN/ FICEK	24	DI/ DI	Input of external reset; Built-in pull-up resistor FICE SCL
VDD18	25	P	1.8V LDO output with an external 1μF ~ 4.7μF capacitor
P1.2/ FICED	26	DB/ DB	GPIO, configurable as INT1 input FICE SDA
P1.3/ HBIAS/ C1PS/ AD12	27	DB/ DO/ AI/ AI	GPIO, configurable as INT1 input Hall bias power supply, internally connected to VDD5 via a switch to output large current CMP1 positive input after function switching Input of ADC channel 12
P1.4/ C0P/ AD10/ HAL0S	28	DB/ AI/ AI/ DI	GPIO, configurable as INT1 input CMP0 positive input Input of ADC channel 10 Hall-IC0 logic level input after function switching
P1.5/ C0M/ C2PS/ AD13	29	DB/ AI/ AI/ AI	GPIO, configurable as INT1 input CMP0 negative input CMP2 positive input after function switching Input of ADC channel 13
P1.6/ C1P/ A1P/ AD9/ HAL1S	30	DB/ AI/ AI/ AI/ DI	GPIO, configurable as INT1 input CMP1 positive input AMP1 positive input Input of ADC channel 9 Hall-IC1 logic level input after function switching

Pin	FU6832N1 QFN32	IO Type	Description
P1.7/ C1M/ A1M	31	DB/ AI/ AI	GPIO, configurable as INT1 input CMP1 negative input AMP1 negative input
P2.0/ AD0/ A1O	32	DB/ AI/ AO	GPIO, configurable as INT1 input Input of ADC channel 0 AMP1 output

2.2 FU6832N1 QFN32 Pinout Diagram

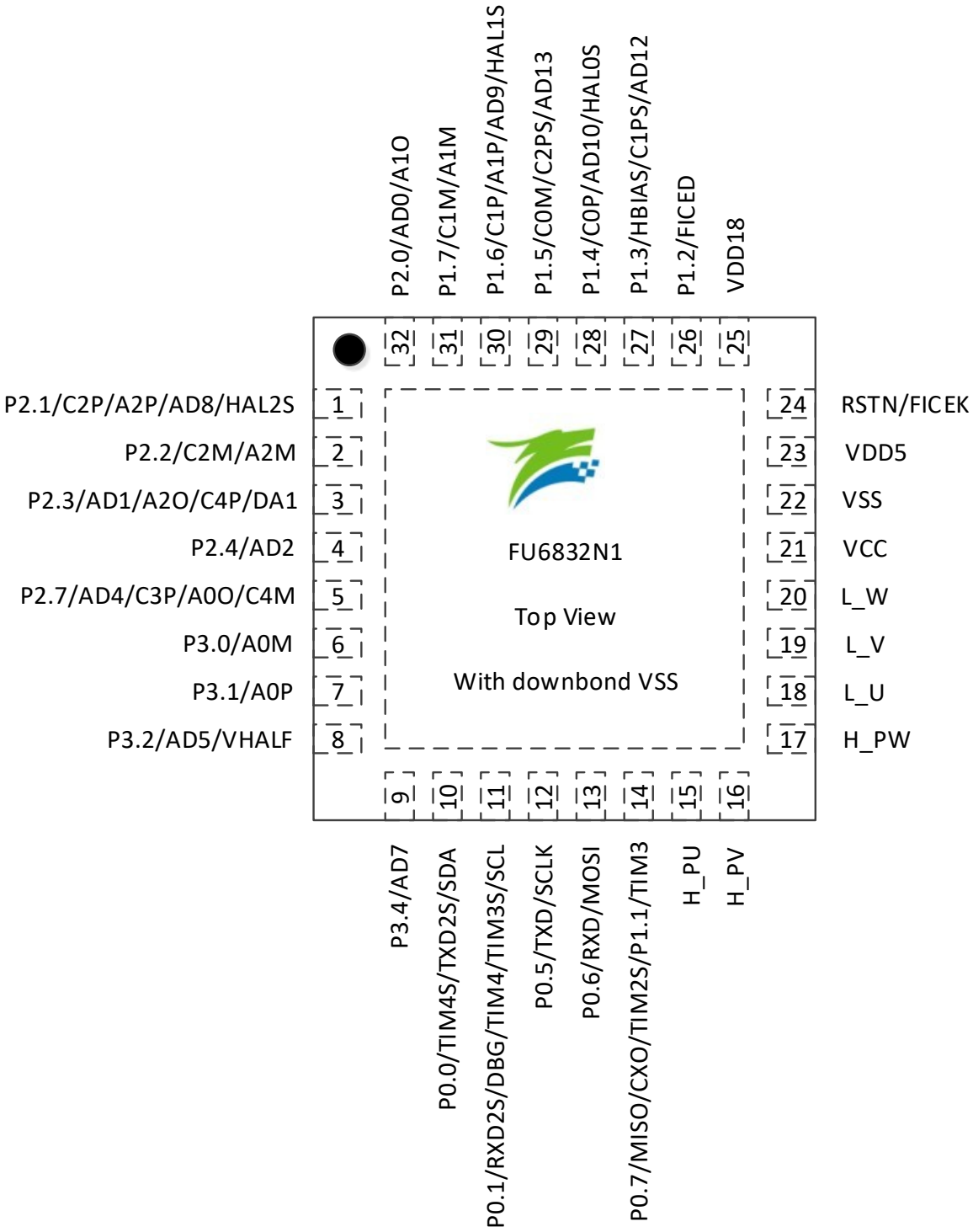
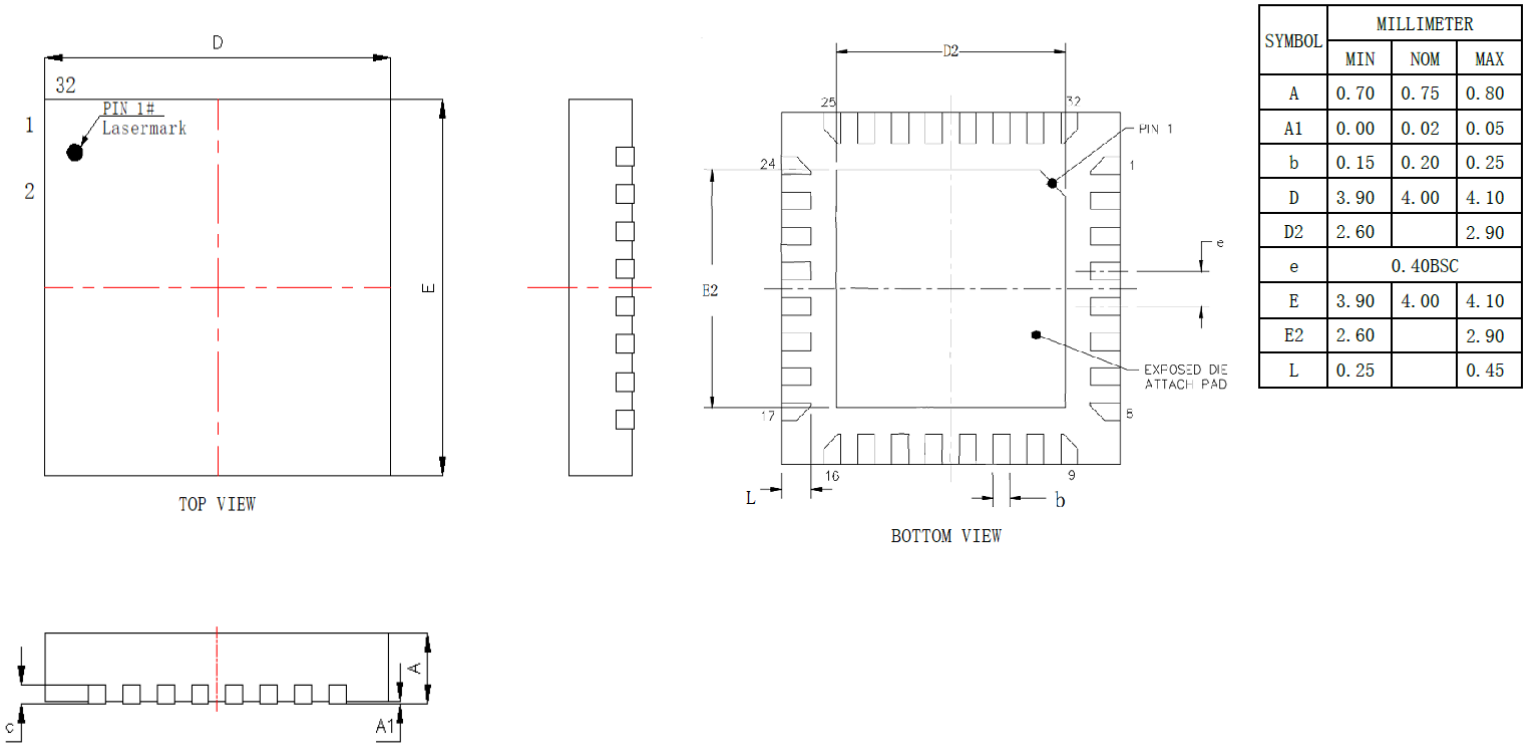


Figure 2-1 FU6832N1 QFN32 Pinout Diagram

3 Package Information

3.1 QFN32_4X4



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
b	0.15	0.20	0.25
D	3.90	4.00	4.10
D2	2.60		2.90
e	0.40BSC		
E	3.90	4.00	4.10
E2	2.60		2.90
L	0.25		0.45

Figure 3-1 QFN32_4X4 Package Dimensions

4 Ordering Information

Table 4-1 Model Selections

Model	MIPS (Peak)	FLASH (kByte)	XRAM (Byte)	Clock Circuit				Driver Interface		Driver type			I^2C/UART/SPI	DMA	GPIO	Timer	Analog Peripherals								Lead-free	Package
																	ADC			DAC						
				Internal Fast Clock	External Fast Clock	Internal Slow Clock	External Slow Clock	3P3N Pre-driver	PWM	Square Wave	Sine Wave	FOC					Number	Channel	Bits	Number	Bits	VREF	Operational Amplifier	Comparator		
FU6832N1	24	16	768	√	—	√	—	√	—	√	—	√	√	√	22	6	1	11	12	2	9\6	√	3	4	√	QFN32 (4x4mm)

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

Table 5-1 Absolute Maximum Ratings

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Operating Ambient Temperature T_A	VCC ≤ 18V	-40	-	125	℃
Operating Junction Temperature T_J		-40	-	150	℃
Storage Temperature		-65	-	150	℃
VCC to VSS Voltage	Peak Duration < 60s	-0.3	-	38	V
VDD5 to VSS Voltage		-0.3	-	6.5	V
RSTN/GPIO to VSS Voltage		-0.3	-	VDD5 + 0.3	V

Notes: Stress values greater than "Absolute Maximum Ratings" listed above may cause irremediable damages to the device. These are stress ratings only, and it is NOT recommended to use your device in conditions that go beyond these stress ratings. Exposure to "Absolute Maximum Ratings" for extended periods may affect device reliability.

5.2 Global Electrical Characteristics

Table 5-2 Global Electrical Characteristics

(T_A = -40℃ ~ 125℃ and VCC = 5V ~ 18V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VCC Operating Voltage[1]	High-voltage Single-power Supply Mode	5	-	18	V
VDD5 Operating Voltage[2]	VCC pin connects with VDD5 pin	3	-	5.5	V
I_{VCC} Operating Current[3]		-	20	-	mA
I_{VCC} Standby Current[3]		-	6	-	mA
I_{VCC} Sleep-mode Current		-	50	150	μA

Notes:

[1] VCC voltage rise rate ranges from 0.5V/μs to 0.1V/s depending on samples batches.

[2] VDD5 must be in the range of 5V~5.5V during Flash write or erase.

[3] Characteristics may vary with different configurations.

5.3 GPIO Electrical Characteristics

Table 5-3 GPIO Electrical Characteristics

(T_A = 25℃ and VCC = 5V ~ 18V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Output Rise Time	50pF load, from 10% to 90%, T_A = 25℃	-	15	-	ns
Output Fall Time	50pF load, from 90% to 10%, T_A = 25℃	-	13	-	ns
V_{OH} Output High Voltage	I_{OH} = 4mA, T_A = -40℃ ~ 125℃	VDD5 - 0.7	-	-	V

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{OL} Output Low Voltage	I_{OL} = 4mA, T_A = -40°C ~ 125°C	-	-	VSS + 0.7	V
V_{IH} Input High Voltage[1]		0.7*VDD5	-	-	V
V_{IL} Input Low Voltage		-	-	0.2*VDD5	V
Pull-up Resistor[2]		-	33	-	kΩ
Pull-up Resistor[3]		-	5.6	-	kΩ
Pull-down Resistor[4]		-	10	-	kΩ

Notes:

[1] When VDD5 = 5V, minimum value of V_{IH} is 0.6*VDD5;

[2] GPIOs except P0[1:0], P1[6:3] and P2[1];

[3] P0[1:0], P1[6:3] and P2[1];

[4] P0[1] and P1[1].

5.4 Pre-driver IO Electrical Characteristics

Table 5-4 Pre-driver IO Electrical Characteristics

(T_A = 25°C, VCC = VDRV = 12V and VCC_MODE = 0 unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
High-side Output Source Current		-	150	-	mA
High-side Output Sink Current		-	90	-	mA
Low-side Output Source Current		-	150	-	mA
Low-side Output Sink Current		-	180	-	mA
Rise Time of High-side Output	1nF load, from 10% to 90%	-	25	-	ns
Fall Time of High-side Output	1nF load, from 90% to 10%	-	90	-	ns
Rise Time of Low-side Output	1nF load, from 10% to 90%	-	115	-	ns
Fall Time of Low-side Output	1nF load, from 90% to 10%	-	60	-	ns
H_PU/V/W High-side High-level Output Voltage	VCC = 5V ~ 18V	-	VCC	-	V
H_PU/V/W High-side Low-level Output Voltage	VCC = 13V ~ 18V	-	VCC - 11	-	V
	VCC = 5V ~ 13V	-	2	-	V
L_PU/V/W Low-side High-level Output Voltage	VCC = 13V ~ 18V	-	11	-	V
	VCC = 5V ~ 13V	-	VCC – 2	-	V
L_PU/V/W Low-side Low-level Output Voltage	VCC = 5V ~ 18V	-	0	-	V

5.5 ADC Electrical Characteristics

Table 5-5 ADC Electrical Characteristics

(T_A = 25°C and VCC = 5V ~ 18V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
INL (Integral Nonlinearity)	12-bit	-	2	-	LSB
DNL (Differential Nonlinearity)	12-bit	-	1.5	-	LSB
OFFSET (Offset Error)	12-bit	-	6	-	LSB

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
SNR (Signal-to-noise Ratio)	f_{IN} = 350kHz	-	70.8	-	dB
ENOB (Effective Number of Bits)	f_{IN} = 350kHz	-	10.5	-	Bit
SFDR (Spurious-free Dynamic Range)	f_{IN} = 350kHz	-	68.2	-	dB
THD (Total Harmonic Distortion)	f_{IN} = 350kHz	-	67	-	dB
R_{IN} Input Resistance		-	800	-	Ω
C_{IN} Input Capacitance		-	30	-	pF
Conversion Time		-	13	-	ADCLK[1]
Sampling Time		3	-	63	ADCLK[1]

Note:

[1] ADCLK=12MHz

5.6 VREF and VHALF Electrical Characteristics

Table 5-6 VREF & VHALF Electrical Characteristics

(T_A = -40°C ~ 125°C and VCC = 5V ~ 18V)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VREF[1]	VREF_CR[VREFVSEL] = 00	4.3	4.5	4.7	V
	VREF_CR[VREFVSEL] = 01 (fixed configuration for FU6832N1)	-	VDD5	-	V
	VREF_CR[VREFVSEL] = 11	-	4	-	V
	VREF_CR[VREFVSEL] = 10	-	3	-	V
VHALF	VREF_CR[VHALFEN] = 1	VREF/2 – 0.2	VREF/2	VREF/2 + 0.2	V

Note:

[1] For FU6832N1, only VDD5 can be used as VREF voltage.

5.7 Operational Amplifier Electrical Characteristics

Table 5-7 Operational Amplifier Electrical Characteristics

(T_A = 25°C and VCC = 5V ~ 18V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{ICMR} Common-mode Input Voltage Range		0	-	VDD5 - 1.5	V
V_{OS} Operational Amplifier Offset Voltage	T_A = 25°C	-	5	10	mV
A_{OL} Open-loop Gain	R_L = 100kΩ	-	80	-	dB
Unity-gain Bandwidth (UGBW)	C_L = 40pF	6	10	-	MHz
Slew Rate (SR)	C_L = 40pF	10	15	-	V/μs
Operational Amplifier Gain	2x	1.88	2	2.12	-
	4x	3.75	4	4.25	-
	8x	7.5	8	8.5	-
	16x	15	16	17	-

Note: With 1kΩ resistors connected in series with both positive and negative inputs of the operational amplifier, the operational amplifier gain can be configured as 2x, 4x, 8x and 16x. The operational amplifier gain varies with external resistors.

5.8 Hall/BEMF Electrical Characteristics

Table 5-8 Hall/BEMF Electrical Characteristics

(T_A = 25°C, VCC = 5V ~ 18V and VCC_MODE = 0 unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
BEMF Built-in Resistor		5.4	6.8	8.2	kΩ
Relative Accuracy between BEMF Built-in Resistors		-	1	-	%

5.9 OSC Electrical Characteristics

Table 5-9 OSC Electrical Characteristics

(T_A = -40°C ~ 125°C, VCC = 5V ~ 18V and VCC_MODE = 0)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
High-speed Oscillator Frequency		23.5	24	24.5	MHz
Low-speed Oscillator Frequency		29	32.8	37	kHz

Note: The frequency of the built-in oscillators refers to the factory test value.

5.10 Reset Electrical Characteristics

Table 5-10 Reset Electrical Characteristics

(T_A = 25°C, VCC = 5V ~ 18V and VCC_MODE = 0 unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Minimum Time for RSTN Released to Low		50	-	-	μs
VDD5 Reset Threshold	Reset Voltage LVR = 2.8V	2.6	2.8	3.0	V
	Reset Voltage LVR = 3.0V	2.8	3.0	3.2	V
	Reset Voltage LVR = 3.5V	3.3	3.5	3.7	V
	Reset Voltage LVR = 3.8V	3.6	3.8	4.0	V

5.11 LDO Electrical Characteristics

Table 5-11 LDO Electrical Characteristics

(T_A = 25°C, VCC = 5V ~ 18V and VCC_MODE = 0 unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD5 Voltage	VCC = 7V ~ 18V, VCC_MODE = 0	4.7	5	5.3	V
VDD18 Voltage		1.65	1.85	2.0	V

5.12 Flash Characteristics

Table 5-12 Flash Characteristics

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Data Storage Duration		-	10	-	year
Byte Programming Time[1]		-	100	-	μs

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Sector Erase Time		-	150	-	ms
Full-chip Erase Time		-	150	-	ms

Note:

[1] The byte programming time may vary slightly depending on the conditions.

5.13 Package Thermal Resistance

Table 5-13 QFN32 Package Thermal Resistance

Parameter	Test Conditions	Value	Unit
Junction-to-ambient Temperature Thermal Resistance θ_{JA}[1]	JEDEC standard, 2S2P PCB	47	°C/W
	JEDEC standard, 1S0P PCB	74	°C/W
Junction-to-case Temperature Thermal Resistance θ_{JC}[1]	JEDEC standard, 2S2P PCB	20	°C/W

Note:

[1] The actual measurements may vary depending on the conditions.

6 Reset Control

6.1 Reset Sources (RST_SR)

The chip includes a reset circuitry with 7 reset sources:

- Power on reset (RSTPOW)
- External reset (RSTEXT)
- Low voltage detector reset (RSTLVD)
- Watchdog timer reset (RSTWDT)
- Flash error detector reset (RSTFED)
- Debug reset (RSTDBG)
- Soft reset (SOFTR)

Reset flag bits can be searched from register RST_SR. Following the last reset, the affected reset flag is set to "1" and all other reset flags are cleared to "0". To clear a reset flag, you can set RST_SR[RSTCLR] to "1" so that RST_SR[7:3] & RST_SR[0] are cleared. After reset, MCU restarts the program from address 0.

6.2 Reset Enable

See the corresponding control registers.

6.3 External Reset and Power-on Reset

The chip resets when RSTN pin remains low for 50μs.

The chip resets when the chip is powered on and VDD5 settles above the reset voltage threshold V_{RST}.

6.4 Low Voltage Detector Reset

The chip's internal circuitry monitors VDD5. When VDD5 drops to a level below V_{RST}, the internal monitor circuitry sends a LVD reset signal to reset the chip.

Configuring corresponding register enables VDD monitor circuitry and sets V_{RST}.

6.5 Watchdog Timer Reset

After the watchdog timer (WDT) is enabled, the software periodically writes 1 to WDT_CR[WDTRF] which initializes watchdog up counter. When watchdog up counter reaches its maximum value, WDT generates an output pulse to reset the chip, which ensures the software runs normally.

6.6 Flash Error Detector Reset

The Flash memory can be programmed by software using the MOVX instruction for read/write/erase operations. A Flash error detector reset (RSTFED) occurs if a Flash erase is attempted targeting the last sector (0x3F80 ~ 0x3FFF) or a Flash write is attempted targeting the last byte (0x3FFF). RSTFED is always enabled and

cannot be disabled.

6.7 Debug Reset

Click **Reset** button of IDE to send a Debug reset signal when the chip enters the debug state.

6.8 Soft Reset

The chip resets immediately when RST_SR[SOFTR] is set to "1". After reset, the flag RST_SR[SOFTR] is set to "1".

6.9 Reset Registers

6.9.1 RST_SR (0xC9)

Bit	7	6	5	4	3	2	1	0
Name	RSTPOW/ RSTCLR	RSTEXT	RSTLVD	RSV	RSTWDT	RSTFED	RSTDBG	SOFTR
Type	R/W1	R	R	—	R	R	R	R/W1
Reset	—	—	—	—	—	—	—	—

Bit	Name	Description
[7]	RSTPOW/ RSTCLR	Power-on Reset Flag Read: 0: Last reset was not a power on reset. 1: Last reset was a power on reset. Write: 0: No effect 1: RST_SR[7:3] & RST_SR[0] are cleared to 0.
[6]	RSTEXT	External RST Pin Reset flag 0: Last reset was not an RST pin reset. 1: Last reset was an RST reset.
[5]	RSTLVD	Low Voltage Detection (LVD) Reset Flag 0: Last reset was not an LVD reset 1: Last reset was an LVD reset
[4]	RSV	Reserved
[3]	RSTWDT	WDT Reset Flag 0: Last reset was not a WDT reset 1: Last reset was a WDT reset
[2]	RSTFED	Flash Error Detector Reset Flag 0: Last reset was not a Flash error detector reset 1: Last reset was a Flash error detector reset
[1]	RSTDBG	Debug Reset Flag 0: Last reset was not a debug reset 1: Last reset was a debug reset
[0]	SOFTR	Soft Reset Flag Read: 0: Last reset was not a soft reset 1: Last reset was a soft reset Write: 0: No effect 1: Generate a soft reset

7 Interrupt

7.1 Interrupt Introduction

The chip supports 16 interrupt sources. Each interrupt source can be individually programmed in IP0 ~ IP3 registers with one of four priority levels. Interrupt flags are located in an SFR or XSFR. The corresponding interrupt flag is set by the hardware to 1 when the internal circuitry or an external signal meets the interrupt conditions. If IE[EA] = 1 and both the associated interrupt EA and IF bits are set to 1, an interrupt request is sent to CPU. If no other interrupt service routine (ISR) of greater priority is currently being serviced, the system enters interrupt state to service the requesting ISR.

Each interrupt source except the Reset Interrupt can be assigned a priority level. A low priority interrupt can be preempted by a high priority interrupt. The low priority interrupt will not be serviced until the ISR for the high priority interrupt completes. An interrupt will not be preempted by another of the same priority level. Each interrupt source can be individually configured to one of four priority levels in the Interrupt Priority (IP) register. Priority level assigned ascends from 0 to 3 and is defaulted to 0. If two interrupt requests are generated at the same time, the interrupt with the higher priority is serviced first. If two interrupt sources have the same priority, a fixed priority order is used to arbitrate. See Table 7-1 for the interrupt sources and default priority orders, where the lower the mark the higher the priority level.

7.2 Interrupt Enable

IE[EA] is the global interrupt enable bit. When EA = 0, the system does not respond to any interrupt request.

Each interrupt can be individually enabled or disabled by configuring the corresponding interrupt enable bit in an SFR or XSFR. When the enable bit of the global interrupt or an interrupt is cleared, the interrupt flag that is set to “1” is held in a pending state. The MCU will immediately enter the interrupt subroutine once the enable bit is set to “1”. Therefore, make sure to clear corresponding interrupt flag bit before enabling the interrupt.

7.3 External Interrupt

The external interrupt has 2 interrupt sources: INT0 and INT1.

The digital input signals from P0.0 ~ P0.6 and the output signals from CMP4 can be used to trigger INT0. The interrupt source is selected through LVSR[EXT0CFG] bit. These interrupt sources share one interrupt entry, one interrupt flag bit TCON[IF0] and one interrupt enable bit IE[EX0]. TCON[IT0] bit selects the interrupt edge. IP0[PX0] bit configures the priority level.

The digital input signals from P1.0 ~ 1.7 and P2.0 ~ 2.7 can be used to trigger INT1. P1IF and P2IF are interrupt flag bits, and P1IE and P2IE are interrupt enable bits. Each trigger source has a corresponding interrupt flag bit and an interrupt enable bit. INT1 can select multiple trigger sources that are recognized by P1IF and P2IF in the interrupt subroutine. These 16 interrupt sources share one interrupt entry and one interrupt enable bit

IE[EX1]. To enable INT1, first set IE[EX1] to 1 and then configure the corresponding enable bit. The interrupt edge is configured by TCON[IT1] bit, and the priority level by IP0[PX1] bit. See 7.5.7 P1_IE (0xD1) ~ 7.5.10 P2_IF (0xD4) for INT1 interrupt flag registers and enable registers.

7.4 Interrupt Summary

Table 7-1 Interrupt Summary

Interrupt Source	Priority Order	Interrupt Vector	Interrupt Flag	Cleared by SW?	Enable Bit	Priority Control
Reset	Highest	0x0000	None	N	Always enabled	Highest
LVW Interrupt/ TSD Interrupt	0	0x0003	LVSR[0] TCON[5]	Y	CCFG1[6]/ IE[1]	IP0[1:0]
INT0	1	0x000B	TCON[2]	Y	IE[0]	IP0[3:2]
INT1	2	0x0013	P1IF[7:0] P2IF[7:0]	Y	IE[2]	IP0[5:4]
FG Interrupt/ DRV Compare Match Interrupt	3	0x001B	DRV_SR[5:4]	Y	DRV_SR[3]/ DRV_SR[2:0]	IP0[7:6]
Timer2 Interrupt	4	0x0023	TIM2_CR1[7:5]	Y	TIM2_CR1[4:3] TIM2_CR0[3]	IP1[1:0]
Timer1 Interrupt	5	0x002B	TIM1_SR[4:0]	Y	TIM1_IER[4:0]	IP1[3:2]
ADC Interrupt	6	0x0033	ADC_CR[0]	Y	ADC_CR[1]	IP1[5:4]
CMP0/1/2 Interrupt	7	0x003B	CMP_SR[6:4]	Y	CMP_CR0[5:0]	IP1[7:6]
RTC Interrupt	8	0x0043	RTC_STA[6]	Y	IE[6]	IP2[1:0]
Timer3 Interrupt	9	0x004B	TIM3_CR1[7:5]	Y	TIM3_CR1[4:3] TIM3_CR0[3]	IP2[3:2]
Systick Interrupt	10	0x0053	DRV_SR[7]	Y	DRV_SR[6]	IP2[5:4]
Timer4 Interrupt	11	0x005B	TIM4_CR1[7:5]	Y	TIM4_CR1[4:3] TIM4_CR0[3]	IP2[7:6]
CMP3 Interrupt	12	0x0063	CMP_SR[7]	Y	CMP_CR0[7:6]	IP3[1:0]
I^2C Interrupt/ UART1 Interrupt	13	0x006B	I2C_SR[0] UT_CR[1:0]	Y	I2C_CR[0]/ IE[4]	IP3[3:2]
SPI Interrupt/ UART2 Interrupt	14	0x0073	SPI_CR0[7:4] SPI_CR1[7] UT2_CR[1:0]	Y	IE[3]/ UT2_BAUDH[5]	IP3[5:4]
DMA Interrupt	15	0x007B	DMA0_CR0[0] DMA1_CR0[0]	Y	DMA0_CR0[2]	IP3[7:6]

Notes:

- UT_CR[RI], UT_CR[TI], DMA0_CR0[DMAIF] and DMA0_CR1[DMAIF] bits can be cleared to “0” or set by software to “1”; and MCU generates an interrupt request when these flags are set to “1”. Other flags can only be cleared to “0” by software, and setting them to “1” has no effect.
- For registers containing serval interrupt flags, you can write a “1” to the active interrupt flags in order to prevent clearing a interrupt flag to 0. For example, use code DRV_SR = (DRV_SR & 0x7F) | 0x30 to clear DRV_SR[SYSTIF].

7.5 Interrupt Registers

7.5.1 IE (0xA8)

Bit	7	6	5	4	3	2	1	0
Name	EA	RTCIE	RSV	ES0	SPIIE	EX1	TSDIE	EX0
Type	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W
Reset	0	0	—	0	0	0	0	0

Bit	Name	Description
[7]	EA	Enable All Interrupts 0: Disable 1: Enable
[6]	RTCIE	RTC Interrupt Enable 0: Disable 1: Enable
[5]	RSV	Reserved
[4]	ES0	UART1 Interrupt Enable 0: Disable 1: Enable
[3]	SPIIE	SPI Interrupt Enable 0: Disable 1: Enable
[2]	EX1	INT1 Interrupt Enable 0: Disable 1: Enable
[1]	TSDIE	TSD Interrupt Enable 0: Disable 1: Enable
[0]	EX0	INT0 Interrupt Enable 0: Disable 1: Enable

7.5.2 IP0 (0xB8)

Bit	7	6	5	4	3	2	1	0
Name	PDRV		PX1		PX0		PLVW_TSD	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	PDRV	FG/DRV Comparison Match Interrupt Priority Control
[5:4]	PX1	INT1 Interrupt Priority Control
[3:2]	PX0	INT0 Interrupt Priority Control
[1:0]	PLVW_TSD	LVW/TSD Interrupt Priority Control

Note: Priority level assigned ascends from 0 to 3, totaling 4 levels

7.5.3 IP1 (0xC0)

Bit	7	6	5	4	3	2	1	0
Name	PCMP		PADC		PTIM1		PTIM2	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	PCMP	CMP0/1/2 Interrupt Priority Control
[5:4]	PADC	ADC Interrupt Priority Control
[3:2]	PTIM1	Timer1 Interrupt Priority Control
[1:0]	PTIM2	Timer2 Interrupt Priority Control

Note: Priority level assigned ascends from 0 to 3, totaling 4 levels.

7.5.4 IP2 (0xC8)

Bit	7	6	5	4	3	2	1	0
Name	PTIM4		PSYSTICK		PTIM3		PRTC	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	PTIM4	Timer4 Interrupt Priority Control
[5:4]	PSYSTICK	Systick Interrupt Priority Control
[3:2]	PTIM3	Timer3 Interrupt Priority Control
[1:0]	PRTC	RTC Interrupt Priority Control

Note: Priority level assigned ascends from 0 to 3, totaling 4 levels.

7.5.5 IP3 (0xD8)

Bit	7	6	5	4	3	2	1	0
Name	PDMA		PSPI_UT2		PI2C_UT1		PCMP3	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	PDMA	DMA Interrupt Priority Control
[5:4]	PSPI_UT2	SPI/UART2 Interrupt Priority Control
[3:2]	PI2C_UT1	I^2C/UART1 Interrupt Priority Control
[1:0]	PCMP3	CMP3 Interrupt Priority Control

Note: Priority level assigned ascends from 0 to 3, totaling 4 levels.

7.5.6 TCON (0x88)

Bit	7	6	5	4	3	2	1	0
Name	RSV		TSDIF	IT1		IF0	IT0	
Type	—	—	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	—	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	TSDIF	TSD Interrupt Flag This bit is set by hardware to 1 when an over-temperature event occurs. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect Note: This flag is often used with the overtemperature status bit LVSR[TSDF]
[4:3]	IT1	INT1 Trigger Level Selection 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on level change (rising or falling)
[2]	IF0	INT0 Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[1:0]	IT0	INT0 Trigger Level Selection 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on level change (rising or falling)

7.5.7 P1_IE (0xD1)

Bit	7	6	5	4	3	2	1	0
Name	P17_IE	P16_IE	P15_IE	P14_IE	P13_IE	P12_IE	P11_IE	P10_IE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	P17_IE	P1.7 INT1 Enable 0: Disable 1: Enable
[6]	P16_IE	P1.6 INT1 Enable 0: Disable 1: Enable
[5]	P15_IE	P1.5 INT1 Enable 0: Disable 1: Enable
[4]	P14_IE	P1.4 INT1 Enable 0: Disable 1: Enable
[3]	P13_IE	P1.3 INT1 Enable 0: Disable 1: Enable
[2]	P12_IE	P1.2 INT1 Enable 0: Disable 1: Enable
[1]	P11_IE	P1.1 INT1 Enable 0: Disable 1: Enable

[0]	P10_IE	P1.0 INT1 Enable 0: Disable 1: Enable

7.5.8 P1_IF (0xD2)

Bit	7	6	5	4	3	2	1	0
Name	P17_IF	P16_IF	P15_IF	P14_IF	P13_IF	P12_IF	P11_IF	P10_IF
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	P17_IF	P1.7 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[6]	P16_IF	P1.6 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[5]	P15_IF	P1.5 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[4]	P14_IF	P1.4 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[3]	P13_IF	P1.3 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[2]	P12_IF	P1.2 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[1]	P11_IF	P1.1 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[0]	P10_IF	P1.0 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending

Note: The interrupt flag is cleared upon a write of “0” by the software.

7.5.9 P2_IE (0xD3)

Bit	7	6	5	4	3	2	1	0
Name	P27_IE	P26_IE	P25_IE	P24_IE	P23_IE	P22_IE	P21_IE	P20_IE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	P27_IE	P2.7 INT1 Enable 0: Disable 1: Enable
[6]	P26_IE	P2.6 INT1 Enable 0: Disable 1: Enable

[5]	P25_IE	P2.5 INT1 Enable 0: Disable 1: Enable
[4]	P24_IE	P2.4 INT1 Enable 0: Disable 1: Enable
[3]	P23_IE	P2.3 INT1 Enable 0: Disable 1: Enable
[2]	P22_IE	P2.2 INT1 Enable 0: Disable 1: Enable
[1]	P21_IE	P421 INT1 Enable 0: Disable 1: Enable
[0]	P20_IE	P2.0 INT1 Enable 0: Disable 1: Enable

7.5.10 P2_IF (0xD4)

Bit	7	6	5	4	3	2	1	0
Name	P27_IF	P26_IF	P25_IF	P24_IF	P23_IF	P22_IF	P21_IF	P20_IF
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	P27_IF	P2.7 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[6]	P26_IF	P2.6 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[5]	P25_IF	P2.5 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[4]	P24_IF	P2.4 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[3]	P23_IF	P2.3 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[2]	P22_IF	P2.2 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[1]	P21_IF	P2.1 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[0]	P20_IF	P2.0 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending

Note: The interrupt flag is cleared upon a write of “0” by the software.

8 I²C

8.1 I²C Introduction

The I²C module provides an industry standard two-wire serial interface and is a simple bi-directional synchronous serial bus for communication between MCU and external I²C devices as shown in Figure 8-1. The bus consists of two serial lines: SDA and SCL. P0.0 serves as SDA port and P0.1 as SCL port. P0.0 and P0.1 automatically shifts into open-drain outputs when I²C is enabled.

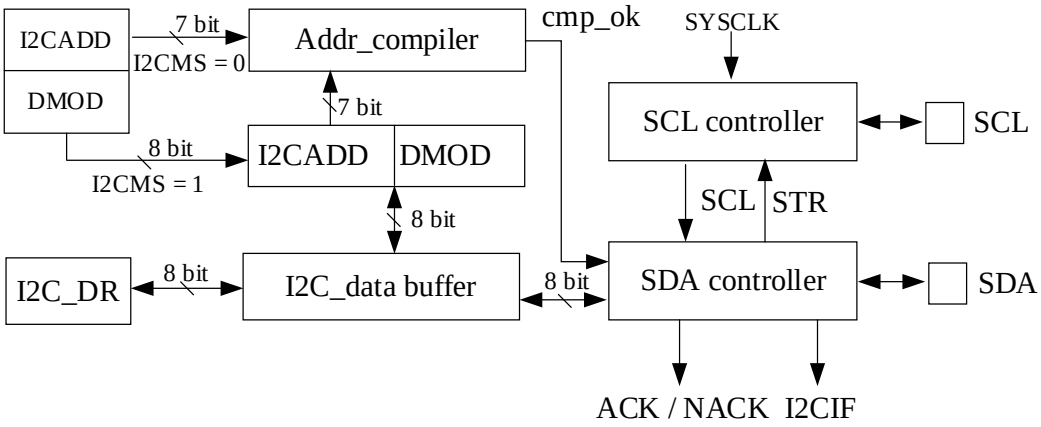


Figure 8-1 I²C Block Diagram

Features:

- Support standard mode (up to 100kHz), fast mode (up to 400kHz) and fast plus mode (up to 1MHz)
- Support master mode and slave mode
- Support 7-bit address mode and general call address mode
- Support DMA data transfer

Both SDA and SCL lines are high when the bus is idle, which is the only basis for detecting whether the bus is idle or not. Only one master device and at least one slave device are active on the bus during the transmission. When the bus is occupied, other devices must wait for the bus idle to start an I²C communication. The master starts the bus to transfer data. Clock signal is sent to all devices via SCL and the slave address and read/write mode are sent via SDA. When a device on the bus matches the address, it acts as a slave. The relationships between masters and slaves or data transfer direction on the bus are not constant. The process for the master to send data to the slave is shown in Figure 8-2. The master first addresses the slave device and waits for the slave response. And then it sends data to the slave. Finally, the master terminates the data transmission. The process for the master to receive data from the slave is shown in Figure 8-3. The master first addresses the slave and waits for the slave response. And then, it receives the data from the slave. Finally, the master terminates the data transmission. In this case, the master generates the timing clock and stops the data transmission.

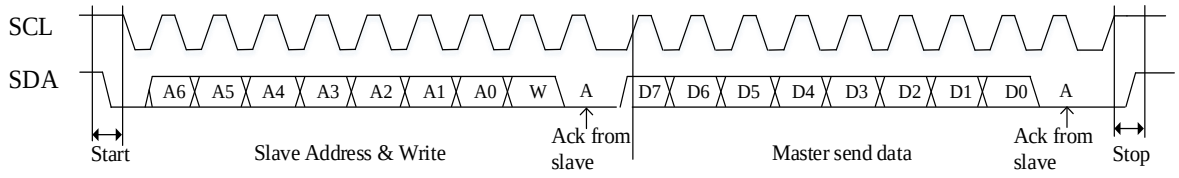


Figure 8-2 Master Sends Data to Slave

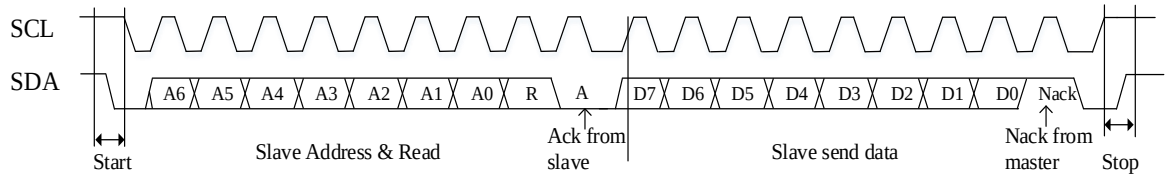


Figure 8-3 Master Receives Data from Slave

8.2 I²C Operations

8.2.1 Master Mode

1. Set I2C_CR[I2CMS] to "1" to select master mode;
2. Configure I2C_CR [I2CSPD] to set the clock rate of SCL;
3. Configure I2C_ID[I2CADD] to set the slave address;
4. Configure I2C_SR[DMOD] to set the read/write direction;
5. Set I2C_CR[I2CEN] to "1"to enable I²C;
6. Set I2C_SR[I2CSTA] to "1" to send START and address. After ACK/NACK is received, I2C_SR[STR] is set to "1" by hardware and SCL is pulled LOW by the master
7. Data Transmission: Write the data to I2C_DR register. The master starts to send data after I2C_SR[STR] is reset and SCL is released. After the data is transmitted and ACK/NACK is received, I2C_SR[STR] is to "1" by hardware and SCL is pulled to LOW by the master;
8. Data Reception: The master starts to receive data after I2C_SR[STR] is reset and SCL is released. When the data is received, I2C_SR[STR] is set to "1" by hardware and SCL is pulled LOW by the master. Configure ACK/NACK via I2C_SR[NACK], and then clear I2C_SR[STR] to release SCL to send ACK/NACK signal. After the data is received, I2C_SR[STR] is set to "1" by hardware and SCL is pulled LOW by the master
9. Stop Communication: Set I2C_SR[I2CSTP] to "1" when I2C_SR[STR] is "1". Stop signal is sent after I2C_SR[STR] is cleared.

8.2.2 Slave Mode

1. Set I2C_CR[I2CMS] to "0" to select slave mode;
2. Configure I2C_ID[I2CADD] to set the slave address or set I2C_ID[GC] to "1" to enable general call

mode;

3. Set I2C_CR[I2CEN] to "1" to enable I²C;
4. After START signal and the correct address are received, I2C_SR[I2CSTA] and I2C_SR[STR] are set to "1" by hardware and SCL is pulled LOW by the slave. ACK/NACK is configured via I2C_SR[NACK] and the slave determines whether to receive or send the data via I2C_SR[DMOD].
5. Data Transmission: Write the data to I2C_DR register and clear I2C_SR[STR] to release SCL. The data is sent after ACK/NACK is transmitted. After the data is sent and ACK/NACK is received from the master, I2C_SR[STR] is set by hardware to "1" and SCL is pulled LOW by the slave
6. Data Reception: Clear I2C_SR[STR] to release SCL to receive data. After the data is received, I2C_SR[STR] is set to "1" by hardware and SCL is pulled LOW by the slave. ACK/NACK is configured via I2C_SR[NACK] to reset I2C_SR[STR] to release SCL for ACK/NACK transmission. If new data is received, I2C_SR[STR] is set by hardware to "1" and SCL is pulled LOW by the slave.
7. RESTART: If the slave is processing a service when receiving START signal, it halts the current routine and waits for receiving address.

8.2.3 I²C Interrupt Sources

The interrupt sources of I²C include:

- I2C_SR[STR] = 1 generates an interrupt. This interrupt source is valid in both master and slave modes.
- I2C_SR[I2CSTP] = 1 generates an interrupt. This interrupt source is only valid in slave mode.

8.3 I²C Registers

8.3.1 I2C_CR (0x4028)

Bit	7	6	5	4	3	2	1	0
Name	I2CEN	I2CMS	RSV			I2CSPD		I2CIE
Type	R/W	R/W	—	—	—	R/W	R/W	R/W
Reset	0	0	—	—	—	0	0	0

Bit	Name	Description
[7]	I2CEN	I²C Enable Enable the corresponding GPIO and switch to I²C mode, serving as collector open-drain output. The pull-up setting decides whether to pull the I²C HIGH. 0: Disable 1: Enable
[6]	I2CMS	Master/slave Mode Selection 0: Slave mode 1: Master mode
[5:3]	RSV	Reserved
[2:1]	I2CSPD	I²C transfer rate settings, valid only in master mode 00: 100kHz 01: 400kHz 10: 1MHz 11: Reserved

[0]	I2CIE	I²C Interrupt Enable 0: Disable 1: Enable

8.3.2 I2C_ID (0x4029)

Bit	7	6	5	4	3	2	1	0
Name	I2CADD							GC
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	1	0	1	0

Bit	Name	Description
[7:1]	I2CADD	Slave address
[0]	GC	General call mode, valid only in slave mode 0: General call is disabled 1: General call is enabled, i.e., the receiving device also reads an ACK at address 0x00

8.3.3 I2C_DR (0x402A)

Bit	7	6	5	4	3	2	1	0
Name	I2C_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	I2C_DR	I²C Data Register Read: Data to be sent or received Write: Data to be sent

8.3.4 I2C_SR (0x402B)

Bit	7	6	5	4	3	2	1	0
Name	I2CBSY	DMOD	RSV	I2CSTA	I2CSTP	STR	NACK	I2CIF
Type	R	R/W	—	R/W	R/W	R/W0	R/W	R
Reset	0	0	—	0	0	0	0	0

Bit	Name	Description
[7]	I2CBSY	I²C busy flag I2C_SR[I2CBSY] is cleared to “0” by hardware when I2C_CR[I2CEN] is set to 0. Master mode: This bit is set to “1” by hardware after START is sent, and cleared to “0” by hardware after STOP is sent. Slave mode: This bit is set to “1” by hardware after START is received and address matches, and cleared to “0” by hardware after STOP is received.
[6]	DMOD	I²C R/W flag 0: WRITE(master sends the data, slave receives the data) 1: READ (master receives the data, slave sends the data) Note: Read only in slave mode
[5]	RSV	Reserved

<table>
<tr><td>[4]</td><td>I2CSTA</td><td>Master mode:
When this bit is configured with “1” by the software, START and address bytes are transmitted after both SCL and SDA are HIGH confirmed by the hardware. This bit is cleared to “0” by hardware automatically when the transmission is completed, and I2C_SR[I2CSTA] writing is forbidden during data transmission. After the data is sent or received, I2C_SR[I2CSTA] is set to “1” to transmit RESTART.
0: Not START and address
1: Send START or RESTART and address

Slave mode:
This bit is set to “1” after hardware receives START and address matches, and cleared to “0” by software.

Table 8-1 Relationship between I2C_SR[I2CSTA/I2CSTP] and I²C Data Type
<table><tr><th>I2CSTA</th><th>I2CSTP</th><th>I²C Data Type</th></tr><tr><td>0</td><td>0</td><td>Data byte</td></tr><tr><td>0</td><td>1</td><td>STOP</td></tr><tr><td>1</td><td>0</td><td>START + address</td></tr><tr><td>1</td><td>1</td><td>STOP received first, then START + address bytes</td></tr></table>Note: When I2C_CR[I2CEN] is “0”, I2C_SR[I2CSTA] is automatically cleared to “0”.</td></tr>
<tr><td>[3]</td><td>I2CSTP</td><td>Master mode:
This bit cannot be written to “1” by software unless I2C_SR[I2CBSY] = 1. STOP is transmitted after I2C_SR[STR] is cleared to release SCL. After the transmission, this bit is cleared to “0” automatically by hardware. If I2CSTA and I2CSTP are written to “1” at the same time and I2C_SR[I2CBSY] is “1”, I²C first sends STOP, then START and address bytes. After START and address bytes are transmitted, I2C_SR[STR] is set to “1” by hardware. I2C_SR[I2CSTP] writing is forbidden during data transmission.
0: STOP is not transmitted.
1: STOP is transmitted.

Slave mode:
This bit is set to 1 by hardware after STOP is received, and cleared to “0” by software. Refer to Table 8-1 for status flags

Note: When I2C_CR[I2CEN] is 0, I2C_SR[I2CSTP] is automatically cleared to “0” by hardware.</td></tr>
<tr><td>[2]</td><td>STR</td><td>I²C Bus Pending Flag
Master Mode:
After START and address or DATA byte are transmitted, I2C_SR[STR] are set to “1” by hardware and SCL is pulled LOW. SCL is released after I2C_SR[STR] is cleared by software. When I2C_SR[I2CSTA] and I2C_SR[I2CSTP] are both “1”, I2C_SR[STR] is set to “1” only after hardware sends STOP and START & address bytes.

Slave mode:
After DATA byte is received or START receives and address matches, I2C_SR[STR] is set to “1” and SCL is pulled LOW. SCL is released after I2C_SR[STR] is cleared by software.

Note: This bit is set to “1” by hardware and cleared to “0” by software. When I2C_SR[I2CEN] = 0, I2C_SR[STR] is automatically cleared to “0”.</td></tr>
<tr><td>[1]</td><td>NACK</td><td>This bit refers to the feedback from a receiver to a sender after a byte is transferred via I²C. It is automatically cleared to “0” when I2C_SR[I2CEN] = 0.
0: ACK, indicating that the receiver can continue to receive data
1: NACK, indicating that the receiver attempts to stop data transmission
When the device is in read mode, I2C_SR[NACK] is configured to send ACK/NACK after the 8th bit of data is received.
0: Bit9 sends ACK
1: Bit9 sends NACK</td></tr>
</table>

		When the device is in write mode, I2C_SR[NACK] is read to receive ACK/NACK after the 8th bit of data is sent. 0: Bit9 receives ACK 1: Bit9 receives NACK
[0]	I2CIF	I²C Interrupt Flag 0: No interrupt pending 1: Interrupt pending In master and slave mode, an interrupt generates when I2C_SR[STR] = 1 In slave mode, an interrupt generates when I2C_SR[I2CSTP] = 1

9 SPI

9.1 SPI Introduction

SPI provides access to a high-speed, full-duplex synchronous serial bus, with its block diagram shown in Figure 9-1. SPI can operate as a master/slave in 3/4-wire mode, and supports multiple masters and slaves on a single SPI bus.

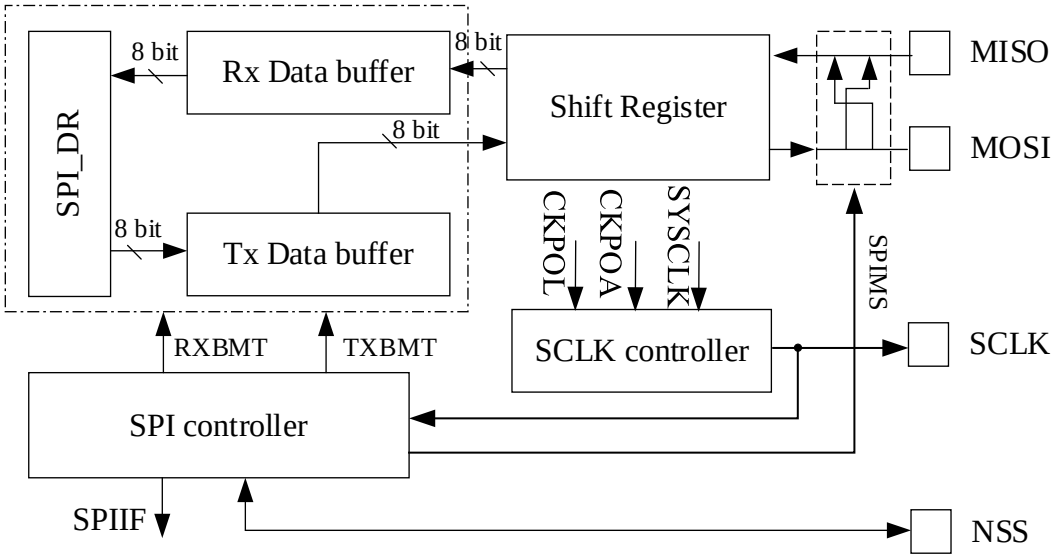


Figure 9-1 SPI Block Diagram

9.2 SPI Operations

9.2.1 Signal Descriptions

The four signals for SPI are MOSI, MISO, SCLK, and NSS.

9.2.1.1 Master Out, Slave In (MOSI)

The MOSI signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. Data is transferred with most-significant bit (MSB) first, namely, the master begins its transmission by driving the MSB of the shift register on its MOSI pin.

9.2.1.2 Master In, Slave Out (MISO)

The MISO signal is an output from a slave device and an input to the master device. The MISO pin is placed in a high-impedance state when the SPI module is disabled or when the SPI operates in 4-wire mode as a slave that is not selected. When the SPI acts as a slave in 3-wire mode or operates in 4-wire mode as a slave that is selected, MISO is used to serially transfer data from the slave to the master. Data is transferred with most-significant bit (MSB) first, namely, the master begins its transmission by driving the MSB of the shift register on its MISO pin.

9.2.1.3 Serial Clock (SCLK)

The SCLK signal is an output from the master device and an input to slave devices. It is used to synchronize serial data transfer between the master and slave. SCLK is generated by SPI operating as a master.

The SCLK signal is ignored when the slave is not selected (NSS = 1) in 4-wire slave mode.

9.2.1.4 Slave Select (NSS)

NSS is dependent on the configuration of SPI_CR1[NSSMOD], i.e., the SPI operating mode. SPI may operate in 3-Wire Mode, 4-Wire Slave/Multi-Master Mode or 4-Wire Master Mode. When SPI operates in 4-Wire Slave/ Multi-Master Mode, NSS is enabled as an input. In this mode, a particular SPI master function is disabled to prevent SPI bus collision where two or more masters simultaneously initiate data transfer. When SPI operates in 4-Wire Single Master Mode, the master NSS is configured as chip select output. When SPI operates in 3-Wire Mode, NSS is disabled. When SPI operates as a master, multiple addressed slave devices can be selected using general-purpose I/O pins.

When SPI_CR1[NSSMOD] = 00, SPI operates in 3-Wire Mode. NSS port is not necessary in this mode and there is only one master and one slave on the SPI bus. The connection diagram is shown in Figure 9-2.

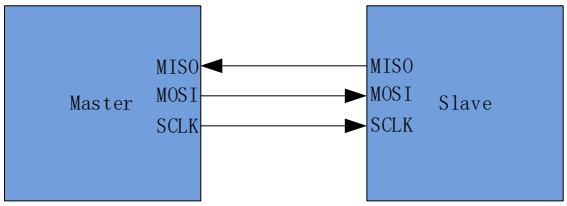


Figure 9-2 Connection Diagram of 3-Wire SPI Mode

When SPI_CR1[NSSMOD] = 01, SPI operates in 4-Wire Slave/ Multi-Master Mode. In this mode, NSS pins on the SPI bus are all configured as inputs, waiting to be addressed by the master. When SPI_CR0[SPIMS] = 0, SPI operates in 4-Wire Slave Mode. If NSS is set to “0”, the slave is selected; while NSS is set to “1”, the slave is not selected. When SPI_CR0[SPIMS] = 1, SPI operates in Master Mode and defaults to multi-master mode. In this mode, NSS pin is configured as input to disable the master SPI. When NSS pin of a master on the bus is pulled low, SPI_CR0[SPIMS] and SPI_CR1[SPIEN] is cleared by hardware to disable SPI, and SPI_CR1[MODF] is set to 1. In this case, SPI communication remains halted before the SPI is re-enabled by software. In this mode, multiple masters are allowed for communication on the SPI bus. The connection diagram is shown in Figure 9-3.

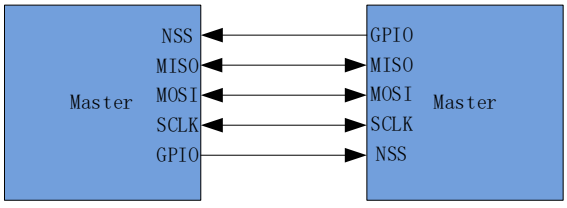


Figure 9-3 Connection Diagram of 4-Wire Multi-Master Mode

When SPI_CR1[NSSMOD] = 1x, SPI operates in 4-Wire Single Master Mode. In this mode, NSS pin of the master on the bus is configured as an output, and NSS pin of the slave devices are configured as inputs. SPI_CR1[NSSMOD0] setting decides the output level of NSS pin serving as signal to select a slave. If SPI_CR1[NSSMOD0]=1, NSS pin of the master outputs high level and if SPI_CR1[NSSMOD0]=0, it outputs low level. The connection diagram is shown as below.

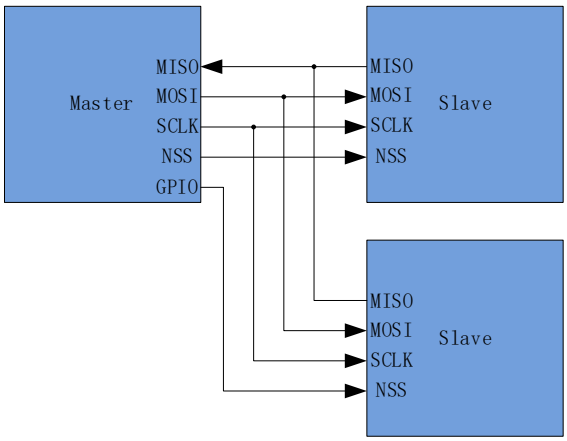


Figure 9-4 Connection Diagram of 4-Wire Single Master Mode

9.2.2 SPI Master Mode

When SPI_CR0[SPIMS] = 1, SPI operates in master mode, which provides SCLK signal for the bus. When the data is written to SPI_DR, it is firstly written to the transmit buffer and SPI_CR1[TXBMT] is cleared to “0”. If the shift register is empty, the data in the transmit buffer will be transferred to the shift register for the transmission. The master SPI begins its transmission by driving the MSB of the shift register on its MOSI pin. After the transmission is completed, SPI_CR1[SPIIF] and SPI_CR1[TXBMT] are set to “1”. While the SPI master transfers data to a slave on the MOSI line, the addressed SPI slave simultaneously transfers data in the shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, SPI_CR1[SPIIF] serves as both a transmit-complete flag and a receive-ready flag, and the data in the shift register is that received by MISO, which is transferred to the receive buffer. The data from SPI_DR is that of the receive buffer. If the data is written to SPI_DR when SPI_CR1[TXBMT] is “0”, the write conflict flag bit SPI_CR1[WCOL] is set to “1” and the data in the transmit buffer keeps unchanged.

9.2.2.1 Master Mode Configuration

1. Configure SPI_CR1[NSSMOD] to set the SPI operating mode;
2. Configure SPI_CR0[CPOL] to set the clock polarity;
3. Configure SPI_CR0[CPHA] to set the clock phase;
4. Set SPI_CR0[SPIMS] to “1” to select master mode;
5. Configure SPI_CLK to set the SCLK rate;

6. Set SPI_CR1[SPIEN] to “1” to enable SPI;
7. Write the data to SPI_DR. SPI transmits the data for each write;
8. After SPI_CR1[SPIIF] is set to 1, SPI_DR is read to receive the data.

9.2.3 SPI Slave Mode

When SPI_CR0[SPIMS] = 0, SPI operates in slave mode. In this mode, SCLK signal is sent by the master SPI. The data is shifted in from MOSI pin and shifted out from MISO pin. If no SCLK signal is input, shift register of the slave remains in the stop state. If the SCLK signal is input, shift register of the slave starts to receive and send data through MOSI and MISO pins. The slave device cannot initiate data transfer. The data sent to the master device is preloaded into the shift buffer by writing to SPI_DR. If the shift register is empty, the data in the transmit buffer is transferred to the shift register. After the transmission is completed, SPI_CR1[SPIIF] and SPI_CR1[TXBMT] are set to 1. The received data that is transferred to receive buffer and the receive buffer empty flag bit SPI_CR0[RXBMT] is cleared, indicating the new data has not been read. If SPI_CR0[RXBMT] is cleared and there is new data ready to be sent to the receive buffer, SPI_CR1[RXOVRN] is set to 1 and the data in the receive buffer remains unaffected. When data is written to SPI_DR, SPI_CR1[TXBMT] is cleared. If data is written in this case, the write conflict flag bit SPI_CR1[WCOL] is set to 1 and the data in the transmit buffer keeps unchanged.

9.2.3.1 Slave Mode Configuration

1. Configure SPI_CR1[NSSMOD] to set the SPI operating mode;
2. Configure SPI_CR0[CPOL] to set the clock polarity;
3. Configure SPI_CR0[CPHA] to set the clock phase;
4. Set SPI_CR0[SPIMS] to 0 to select slave mode;
5. Set SPI_CR1[SPIEN] to 1 to enable SPI;
6. Write data to SPI_DR and wait for the master to send the clock signal.

9.2.4 SPI Interrupt Sources

The interrupt sources of SPI include:

- SPI interrupt flag SPI_CR1[SPIIF] is set to 1 each time after the byte is transferred.
- If SPI_DR is written when the data in transmit buffer has not been transferred to the shift register, the write conflict flag SPI_CR1[WCOL] is set to 1 and the write operation will not be implemented.
- When SPI works as a master in a multi-master system and NSS pin is pulled LOW, the mode error flag SPI_CR1[MODF] is set to 1. When a mode error occurs, SPI_CR0[SPIMS] and SPI_CR1[SPIEN] are cleared. SPI is forbidden to allow another master to control the bus.
- The receive overflow flag SPI_CR1[RXOVRN] is set to 1 when SPI operates in slave mode and a

transmission is completed while the receive buffer still holds unread data from a previous transfer. And the received data will not be transferred to the receive buffer.

9.2.5 Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the CPHA and CPOL bits in the SPI_CR0 Register. SPI_CR0[CPHA] selects the clock phase (the edge of the SCLK signal used to latch the data in shift register). SPI_CR0[CPOL] selects the polarity. Both master and slave devices must be configured with the same clock phase and polarity. When the clock phase and polarity is configured, SPI shall be disabled (SPI_CR1[SPIEN] = 0). The timing relationships of SCL and SDA in clock phase and polarity combinations are shown in Figure 9-5 and Figure 9-6.

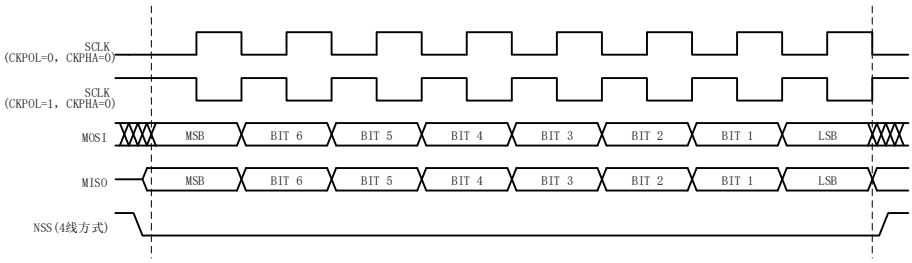


Figure 9-5 SDA/SCL Line Timing Diagram (SPI_CR0[CPHA] = 0)

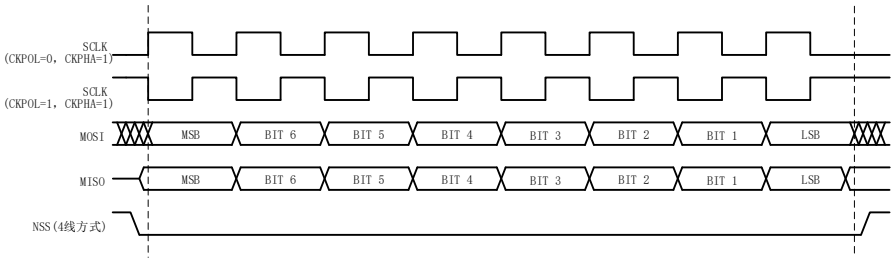


Figure 9-6 SDA/SCL Line Timing Diagram (SPI_CR0[CPHA] = 1)

9.3 SPI Registers

9.3.1 SPI_CR0 (0x4030)

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	SPIMS	CPHA	CPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

Bit	Name	Description
[7]	SPIBSY	Busy Flag 0: No data is transferring via SPI. 1: Data is transferring via SPI.
[6]	SPIMS	Master/Slave Mode Selection 0: Slave 1: Master

[5]	CPHA	Clock Phase 0: Data received on leading edge and transmitted on trailing edge of active SCLK 1: Data transmitted on leading edge and received on trailing edge of active SCLK
[4]	CPOL	Clock Idle Polarity 0: Low level in idle state 1: High level in idle state
[3]	SLVSEL	NSS Select Flag This bit is set to 1 when the filtered signal of NSS is low, indicating that the device is selected as slave. When NSS is high, this bit is cleared to "0", indicating that the device is not selected as slave. 0: Not selected as slave 1: Selected as slave
[2]	NSSIN	NSS real-time signal, unfiltered.
[1]	SRMT	Shift Register Empty Flag (valid only in slave mode) 0: Data has been shifted out of the Transit Buffer into the shift register or SCLK changes. 1: There is no data in the shift register or transmit and receive buffers Note: SPI_CR0[SRMT] = 1 in master mode
[0]	RXBMT	Receive Buffer Empty Flag (valid only in slave mode) 0: New data in the receive buffer has not been read 1: Data has been read and there is no new data in the receive buffer Note: SPI_CR0[RXBMT] =1 in the master mode

Note: Clock phase and polarity modes SPI_CR0[CPHA:CPOL]:

- 00: Receive data on rising edge, and send data on falling edge. Idle level is low.
- 01: Send data on rising edge, and receive data on falling edge. Idle level is high.
- 10: Send data on rising edge, and receive data on falling edge. Idle level is low.
- 11: Receive data on rising edge, and send data on falling edge. Idle level is high.

9.3.2 SPI_CR1 (0x4031)

Bit	7	6	5	4	3	2	1	0
Name	SPIIF	WCOL	MODF	RXOVRN	NSSMOD1	NSSMOD0	TXBMT	SPIEN
Type	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	1	0

Bit	Name	Description
[7]	SPIIF	SPI Interrupt Flag This bit is set to "1" by hardware each time after a data frame (8-bit) is transferred. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[6]	WCOL	Write Conflict Interrupt Flag When TXBMT is 0, a write to SPI_DR sets this bit to 1. This bit can be cleared to "0" by software only. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect

[5]	MODF	Master Mode Error Interrupt Flag This bit is set to "1" when a master mode conflict is detected (SPI_CR0[NSSIN] = 0, SPI_CR0[SPIMS] = 1 and SPI_CR1[NSSMOD] = 01) This bit can be cleared to "0" by software only. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[4]	RXOVRN	Receive Overflow Interrupt Flag (valid only in slave mode) This bit is set to 1 by hardware (and generates a SPI interrupt) when the last bit of the current transfer has been shifted into the shift register and the receive buffer still holds unread data from the previous transfer. This bit cannot be cleared to "0" automatically by hardware and can be cleared by software only. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[3:2]	NSSMOD	SPI Mode Selection 00: 3-Wire Slave Mode or 3-wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (default value). NSS pin is configured as an input. 1X: 4-Wire Single Master Mode, NSS pin is configured as output and outputs the SPI_CR1[2] value
[1]	TXBMT	Transmit Buffer Empty Flag This bit is cleared when new data is written to the transmit buffer. It is set to "1" when the data in the transmit buffer is transferred to the SPI shift register, indicating that it is safe to write a new byte to the transmit buffer. 0: A new byte is written to the transmit buffer. 1: Data in the transmit buffer has been transferred to the shift register.
[0]	SPIEN	SPI Enable 0: Disable 1: Enable

9.3.3 SPI_CLK (0x4032)

Bit	7	6	5	4	3	2	1	0
Name	SPI_CLK							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	SPI_CLK	SPI Baud Rate Setting This bit is valid in master mode only, and can be written only when SPI_CR1[SPIEN] = 0. Baud rate = SYSCLK/2/(SPI_CLK + 1) For example, if baud rate = 2400kHz, then SPI_CLK = (24M/2/2400k) - 1 = 4, i.e., 0x04 Note: When PI/PID and slave SPI are active at the same time (using DMA transfer), the master SPI Baud Rate shall be less than 600kHz to prevent erroneous data transmitted from the slave SPI.

9.3.4 SPI_DR (0x4033)

Bit	7	6	5	4	3	2	1	0
Name	SPI_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	SPI_DR	SPI Data Registers SPI_DR register is used to send and receive SPI data Read: Data in the receive buffer Write: Write data to the transmit buffer and initiate a transfer

10 UART

10.1 Introduction

UART is a full-duplex or half-duplex serial data exchange interface as shown in Figure 9-7. The baud rate is configurable and supports DMA transmission. Figure 9-8 depicts the UART timing.

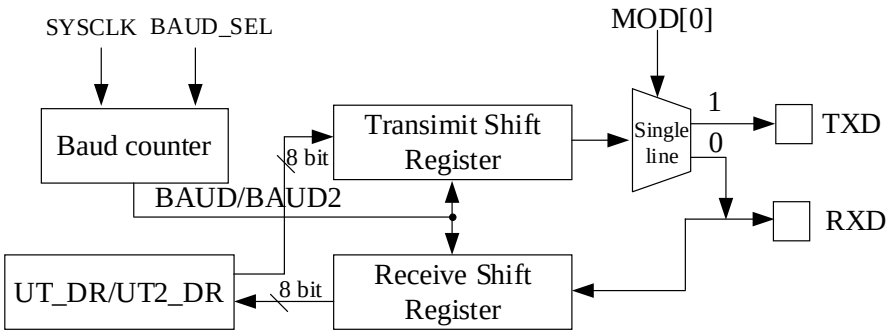


Figure 9-7 UART Block Diagram

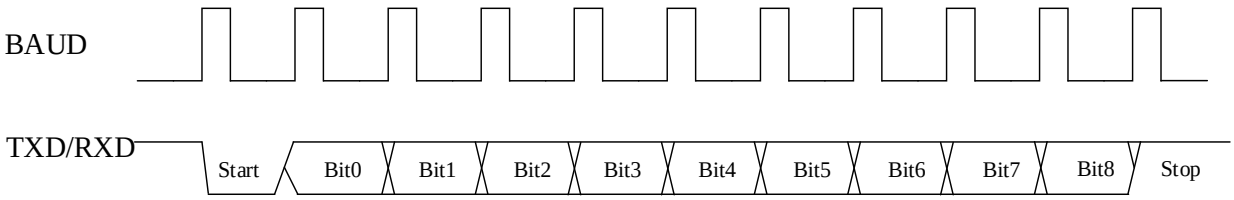


Figure 9-8 UART Communication Timing Diagram

10.2 UART Operations

10.2.1 UART1 Operating Instructions

10.2.1.1 UART1 Mode0

UART1 mode0 works in single-wire half-duplex mode. RXD pin is configured as both an output (Transmit Data Bus) and an input (Receive Data Bus). It uses a total of 10 bits (1-bit start, 8-bit data, 1-bit stop) to receive or transmit data. The baud rate is configured by UT_BAUD[BAUD].

Sending Data: Write the data to UT_DR and clear UT_CR[TI]. RXD outputs 10-bit data. UT_CR[TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT_CR[REN] to "1" to receive the data and clear UT_CR[RI]. The data is received via RXD. After the data is received, UT_CR[RI] is set to “1” and UT_DR is read to obtain the data.

10.2.1.2 UART1 Mode1

UART1 mode1 works in full/half duplex mode. TXD pin is configured as an output (Transmit Data Bus), and RXD as an input (Receive Data Bus). It uses a total of 10 bits (1-bit start, 8-bit data, 1-bit stop) to receive or transmit data. The baud rate is configured by UT_BAUD[BAUD].

Sending Data: Write the data to UT_DR and clear UT_CR[TI]. TXD outputs 10-bit data. UT_CR[TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT_CR[REN] to "1" to receive the data and clear UT_CR[RI]. The data is received via RXD. After the data is received, UT_CR[RI] is set to "1" and UT_DR is read to obtain the data.

10.2.1.3 UART1 Mode2

UART1 mode2 works in single-wire half-duplex mode. RXD pin is configured as both an output (Transmit Data Bus) and an input (Receive Data Bus). It uses a total of 11 bits (1-bit start, 9-bit data and 1-bit stop) to receive or transmit data. The baud rate is configured by UT_BAUD[BAUD].

Sending Data: Write the first 8 low bits of the data to UT_DR and the 9th bit to UT_CR[TB8], and clear UT_CR[TI]. TXD outputs 11-bit data. UT_CR[TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT_CR[REN] to "1" to receive the data and clear UT_CR[RI]. The data is received via RXD. After the data is received, UT_CR[RI] is set to "1". UT_CR[RB8] stores the 9th bit of the data, and UT_DR stores the first 8 low bits.

10.2.1.4 UART1 Mode3

UART1 mode3 works in full/half duplex mode. TXD pin is configured as an output (Transmit Data Bus), and RXD as an input (Receive Data Bus). It uses a total of 11 bits (1-bit start, 9-bit data, 1-bit stop) to receive or transmit data. The baud rate is configured by UT_BAUD[BAUD].

Sending Data: Write the first 8 low bits of the data to UT_DR and the 9th bit to UT_CR[TB8], and clear UT_CR[TI]. TXD outputs 11-bit data. UT_CR[TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT_CR[REN] to "1" to receive the data and clear UT_CR[RI]. The data is received via RXD. After the data is received, UT_CR[RI] is set to "1". UT_CR[RB8] stores the 9th bit of the data, and UT_DR stores the first 8 low bits.

10.2.1.5 UART1 Interrupt Sources

UART1 interrupt sources include:

- After UART1 sends the data, UT_CR[TI] is set to 1 by hardware
- After UART1 receives the data and STOP, UT_CR[RI] is set to 1 by hardware

10.2.2 UART2 Operating Instructions

10.2.2.1 UART2 Mode0

UART2 mode0 works in single-wire half-duplex mode. RXD pin is configured as both an output (Transmit Data Bus) and an input (Receive Data Bus). It uses a total of 10 bits (1-bit start, 8-bit data, 1-bit stop) to receive or transmit data. The baud rate is configured by UT2_BAUD[BAUD2].

Sending Data: Write the data to UT2_DR and clear UT2_CR[UT2TI]. RXD outputs 10-bit data. UT2_CR[UT2TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT2_CR[UT2REN] to "1" to receive the data and clear UT2_CR[UT2RI]. The data is received via RXD. After the data is received, UT2_CR[UT2RI] is set to "1" and UT2_DR is read to obtain the data.

10.2.2.2 UART2 Mode1

UART2 mode1 works in full/half duplex mode. TXD pin is configured as an output (Transmit Data Bus), and RXD as an input (Receive Data Bus). It uses a total of 10 bits (1-bit start, 8-bit data, 1-bit stop) to receive or transmit data. The baud rate is configured by UT2_BAUD[BAUD2].

Sending Data: Write the data to UT2_DR and clear UT2_CR[UT2TI]. TXD outputs 10-bit data. UT2_CR[UT2TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT2_CR[UT2REN] to "1" to receive the data and clear UT2_CR[UT2RI]. The data is received via RXD. After the data is received, UT2_CR[UT2RI] is set to "1" and UT2_DR is read to obtain the data.

10.2.2.3 UART2 Mode2

UART2 mode2 works in single-wire half-duplex mode. RXD pin is configured as both an output (Transmit Data Bus) and an input (Receive Data Bus). It uses a total of 11 bits (1 start bit, 9 data bits, and 1 stop bit) to receive or transmit data. The baud rate is configured by UT2_BAUD[BAUD2].

Sending Data: Write the first 8 low bits of the data to UT2_DR and the 9th bit to UT2_CR[UT2TB8], and clear UT2_CR[UT2TI]. TXD outputs 11-bit data. UT2_CR[UT2TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT2_CR[UT2REN] to "1" to receive the data and clear UT2_CR[UT2RI]. The data is received via RXD. After the data is received, UT2_CR[UT2RI] is set to "1". UT2_CR[UT2RB8] stores the 9th bit of the data, and UT2_DR stores the first 8 low bits.

10.2.2.4 UART2 Mode3

UART2 mode3 works in full/half duplex mode. TXD pin is configured as an output (Transmit Data Bus), and RXD as an input (Receive Data Bus). It uses a total of 11 bits (1-bit start, 9-bit data, 1-bit stop) to receive or transmit data. The baud rate is configured by UT2_BAUD[BAUD2].

Sending Data: Write the first 8 low bits of the data to UT2_DR and the 9th bit to UT2_CR[UT2TB8], and clear UT2_CR[UT2TI]. TXD outputs 11-bit data. UT2_CR[UT2TI] is set to "1" after the transmission is completed.

Receiving Data: Set UT2_CR[UT2REN] to "1" to receive the data and clear UT2_CR[UT2RI]. The data is

received via RXD. After the data is received, UT2_CR[UT2RI] is set to "1". UT2_CR[UT2RB8] stores the 9th bit of the data, and UT2_DR stores the first 8 low-order bits.

10.2.2.5 UART2 Interrupt Sources

UART2 interrupt sources include:

- After UART2 sends data, UT2_CR[UT2TI] is set to "1" by hardware.
- After UART2 receives data and STOP, UT2_CR[UT2RI] is set to "1" by hardware.

10.3 UART1 Registers

10.3.1 UT_CR (0x98)

Bit	7	6	5	4	3	2	1	0
Name	MOD		SM2	REN	TB8	RB8	TI	RI
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	MOD	Mode Selection 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
[5]	SM2	Communication Mode 0: Single-device communication 1: Multi-device communication
[4]	REN	Serial input enable 0: Disable 1: Enable
[3]	TB8	Bit9 of the sent data in mode 2 and mode 3
[2]	RB8	Bit9 of the received data in mode 2 and mode 3
[1]	TI	Data Sending Completed Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to "0". 1: The interrupt is generated.
[0]	RI	Data Reception Completed Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to "0". 1: The interrupt is generated.

10.3.2 UT_DR (0x99)

Bit	7	6	5	4	3	2	1	0
Name	UT_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	UT_DR	Send/Receive Data Read: Data received Write: Data to be sent Note: The UART1 data buffer consists of two independent buffers, i.e., a receive buffer and a transmit buffer, which can send and receive data at the same time. The transmit buffer can be written only but not read, while the receive buffer can be read only but not written. Both buffers share a same address.

10.3.3 UT_BAUD (0x9A, 0x9B)

UT_BAUDH(0x9B)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	BAUD_SEL	RSV			BAUD[11:8]			
Type	R/W	—	—	—	R/W	R/W	R/W	R/W
Reset	0	—	—	—	0	0	0	0
UT_BAUDL(0x9A)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BAUD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	1	0	1	1

Bit	Name	Description
[15]	BAUD_SEL	Frequency Multiplier Enable 0: Disable 1: Enable
[14:12]	RSV	Reserved
[11:0]	BAUD	Baud Rate Setting Baud rate = SYSCLK/(16/(1 + UT_BAUD[BAUD_SEL]))/(UT_BAUD[BAUD] + 1) For example, baud rate = 9600, UT_BAUD[BAUD_SEL] = 0, UT_BAUD[BAUD] = (24M/16/9600/(1 + 0)) - 1 = 155 (0x9B)

10.4 UART2 Registers

10.4.1 UT2_CR (0x8A)

Bit	7	6	5	4	3	2	1	0
Name	UT2MOD		UT2SM2	UT2REN	UT2TB8	UT2RB8	UT2TI	UT2RI
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W0	R/W0
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	UT2MOD	Mode Selection 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
[5]	UT2SM2	Communication Mode 0: Single-device communication 1: Multi-device communication
[4]	UT2REN	Serial Input Enable 0: Disable 1: Enable
[3]	UT2TB8	Bit9 of the sent data in mode 2 and mode 3

[2]	UT2RB8	Bit9 of the received data in mode 2 and mode 3
[1]	UT2TI	Data Sending Completed Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	UT2RI	Data Reception Completed Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect

10.4.2 UT2_DR (0x89)

Bit	7	6	5	4	3	2	1	0
Name	UT2_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	UT2_DR	Send/Receive Data Read: Data received Write: Data to be sent Note: The data buffer of UART2 contains two independent receive and send buffers, which can send and receive data at the same time. The send buffer can only be written but not read while the receive buffer can only be read but not written. The two buffers share a same address.

10.4.3 UT2_BAUD (0x4042, 0x4043)

UT2_BAUDH(0x4042)								
Bit	15	14	13	12	11	10	9	8
Name	BAUD2_SEL	UART2CH	UART2IEN	RSV	BAUD2[11:8]			
Type	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W
Reset	0	0	0	—	0	0	0	0
UT2_BAUDL(0x4043)								
Bit	7	6	5	4	3	2	1	0
Name	BAUD2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	1	0	1	1

Bit	Name	Description
[15]	BAUD2_SEL	Frequency Multiplier Enable 0: Disable 1: Enable
[14]	UART2CH	UART2 Function Switching Enable 0: UART2 port function not transferred, P3.6 for RXD; P3.7 for TXD 1: UART2 port function switching, P0.1 for RXD; P0.0 for TXD
[13]	UART2IEN	UART2 Interrupt Enable 0: Disable 1: Enable

[12]	RSV	Reserved
[11:0]	BAUD2	Baud Rate Setting Baud rate = SYSCLK/(16/(1 + UT2_BAUD[BAUD_SEL]))/(UT2_BAUD[BAUD2] + 1) For example, baud rate = 9600, UT2_BAUD[BAUD_SEL] = 0; UT2_BAUD[BAUD2] = (24M/16/9600/(1 + 0)) - 1 = 155 (0x9B)

11 MDU

MDU is a computational co-processing unit, which assists the CPU to process complex operations efficiently. MDU provides multiplication, division, trigonometric functions, low-pass filter and PID operation. MDU module can be invoked in different interrupt services and master programs, and the results are independent from each other.

11.1 MDU Features

- Support invocation with nested interrupt
- Hardware acceleration to reduce CPU load
- Support the following modes:
  - 16-bit signed multiplication
  - 16-bit signed multiplication (result shifted with one-bit left)
  - 16-bit unsigned multiplication
  - 32-bit/16-bit unsigned division
  - Low-pass filter
  - Coordinate transformation
  - Arctangent

11.2 MDU Instructions

11.2.1 MDU Operations

MDU is operated as follows.

1. Set MDU_CR[MDURUN] to 1;
2. Configure MUD_MD to select the operation mode;
3. Write data to MDU_A, MDU_B, MDU_C and MDU_D. The operation starts after MDU_C[7:0] is written;
4. Wait for MDU_CR[MDUBUSY] to be cleared by hardware;
5. Set MDU_CR[MDUDONE] to 1.

Notes:

- Before using MDU, MDU_CR[MDURUN] must be set to 1. After the operation is completed, MDU_CR[MDUDONE] must be set to 1. These two steps ensure the data of MDU is not affected by nested calls of different interrupts and the main function.
- Make sure the operation mode and other registers have been written before MDU_C[7:0] is written.

11.2.2 16-bit Signed Multiplication with the Result Shifted Left by 1 Bit

When MDU_MD[2:0] = 000, MDU module works in 16-bit signed multiplication mode with the result

shifted left by 1 bit. As shown in Table 11-1, after 16-bit signed data is written to MDU_A and MDU_C as the multiplied number and multiplier respectively, 32-bit signed data is obtained by the product shifting left by 1 bit. The 16 high-order bits of this data are read by MDU_A, and the 16 low-order bits are read by MDU_B.

Table 11-1 Register Definitions in16-bit Signed Multiplication Mode with the Result Shifted Left by One-bit

Data Register	Input Data	Output Data
MDU_A	Multiplied number	High-order 16 bits of the product
MDU_B	—	Low-order 16 bits of the product
MDU_C	Multiplier	—
MDU_D	—	—

11.2.3 16-bit Signed Multiplication

When MDU_MD[2:0] = 001, MDU module works in 16-bit signed multiplication mode. As shown in Table 11-2, 31-bit signed data is obtained after 16-bit signed data is written to MDU_A and MDU_C as the multiplied number and multiplier respectively. The high 16 bits of the data is read by MDU_A, and the low 16 bits is read by MDU_B.

Table 11-2 Register Definitions in 16-Bit Signed Multiplication Mode

Data Register	Input Data	Output Data
MDU_A	Multiplied number	High-order 16 bits of the product
MDU_B	—	Low-order 16 bits of the product
MDU_C	Multiplier	—
MDU_D	—	—

11.2.4 16-bit Unsigned Multiplication

When MDU_MD[2:0] = 010, MDU module works in 16-bit unsigned multiplication mode. As shown in Table 11-3, 32-bit unsigned data is obtained after 16-bit unsigned data is written to MDU_A and MDU_C as the multiplied number and multiplier respectively. The high 16 bits of the data is read by MDU_A, and the low 16 bits is read by MDU_B.

Table 11-3 Meaning of the Register in 16-Bit Unsigned Multiplication Mode

Data Register	Input Data	Output Data
MDU_A	Multiplied number	High-order 16 bits of the product
MDU_B	—	Low-order 16 bits of the product
MDU_C	Multiplier	—
MDU_D	—	—

11.2.5 32-bit/16-bit Unsigned Division

When MDU_MD[2:0] = 011, the MDU works in 32-bit/16-bit unsigned division mode. As shown in Table 11-4, the dividend is 32-bit unsigned data, and the divisor is 16-bit unsigned data. 32-bit unsigned quotient with

16-bit unsigned remainder is obtained after high 16 bits of the dividend is written to MDU_A, low 16 bits of the dividend is written to MDU_B, and divisor is written to MDU_C. The high 16 bits of the quotient is read by MDU_A, the low 16 bits is read by MDU_B, and the remainder is read by MDU_C.

Table 11-4 Meaning of the Register in Unsigned Division Mode

Data Register	Input Data	Output Data
MDU_A	High-order 16 bits of the dividend	High 16 bits of the quotient
MDU_B	Low-order 16 bits of the dividend	Low 16 bits of the quotient
MDU_C	Divisor	Remainder
MDU_D	—	—

11.2.6 Low-Pass Filter

When MDU_MD[2:0] = 110, MDU works in LPF mode.

The calculation formula of LPF is:

$$Y_k = Y_{k-1} + K \times (X_k - Y_{k-1})$$

Where,

Y_k: Filtered value

Y_{k-1}: Previous filtered value

K: Filter coefficient

X_k: Value to be filtered

As shown in Table 11-5, Y_k and Y_{k-1} are 32-bit signed data, X_k is 16-bit signed data, and K is 8-bit unsigned data. Y_k is obtained after high 16 bits of Y_{k-1} is written to MDU_B, low 16 bits of Y_{k-1} to MDU_C, K to MDU_D and X_k to MDU_A. High 16 bits of Y_k is read by MDU_B, and low 16 bits is read by MDU_C.

Table 11-5 Meaning of the Register in LPF Mode

Data Register	Input Data	Output Data
MDU_A	X_k	—
MDU_B	Y_{k-1}[31:16]	Y_k[31:16]
MDU_C	Y_{k-1}[15:0]	Y_k[15:0]
MDU_D	K	—

11.2.7 Coordinate Transformation

When MDU_MD[2:0] = 100, MDU works in coordinate transformation mode. As shown in Figure 11-1, the coordinate transformation converts the components cos_i and sin_i of vector A under the x-y axis to the components cos_o and sin_o under the x'-y' axis, with the x'-y' axis lagging the x-y axis by θ.

The formula for coordinate transformation is:

$$cos_o = cos_i \times cos\,\theta - sin_i \times sin\,\theta$$

$$sin_o = cos_i \times sin\,\theta + sin_i \times cos\,\theta$$

In particular, when $sin_i = 0$, the coordinate transformation is a sine and cosine calculation with cos_i as the amplitude, calculated as:

$$cos_o = cos_i \times cos\,\theta$$

$$sin_o = cos_i \times sin\,\theta$$

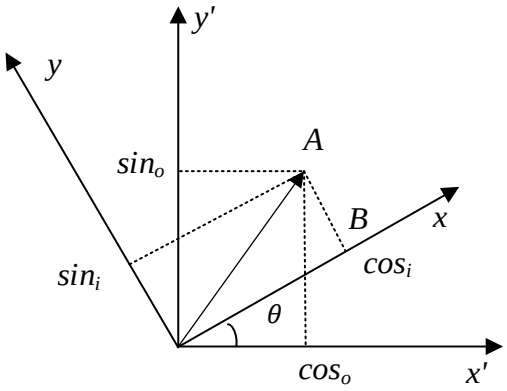


Figure 11-1 Coordinate Transformation

As shown in Table 11-6, cos_i, sin_i, θ, cos_o and sin_o are all 16-bit signed data. cos_i is written to MDU_A, θ to MDU_B and sin_i to MDU_C to calculate cos_o and sin_o. cos_o is read by MDU_A and sin_o by MDU_C.

Table 11-6 Register Definitions in Coordinate Transformation Mode

Data Register	Input Data	Output Data
MDU_A	cos_i	cos_o
MDU_B	θ	—
MDU_C	sin_i	sin_o
MDU_D	—	—

11.2.8 Arctangent

When MDU_MD[2:0] = 101, MDU module works in arctangent mode.

Arctangent function calculates the magnitude and angle of the vector based on the input sine and cosine values. The calculation formula is:

$$U = \sqrt{(Usin\,\theta)^2 + (U\,cos\,\theta)^2}$$

$$\theta = tan^{-1}\left(\frac{U\,sin\,\theta}{Ucos\,\theta}\right)$$

Where,

$Usin\theta$: Sine component of vector

$Ucos\theta$: Cosine component of vector

θ: Calculated vector angle

U: Calculated vector magnitude

As shown in Table 11-7, $Ucos_\theta$, $Usin_\theta$, U and θ are 16-bit signed data. $Ucos_\theta$ is written to MDU_A and $Usin_\theta$ to MDU_C to calculate U and θ. U is read by MDU_A, and θ is read by MDU_B.

Table 11-7 Register Definitions in Arctangent Mode

Data Register	Input Data	Output Data
MDU_A	$Ucos\theta$	U
MDU_B	—	θ
MDU_C	$Usin\theta$	—
MDU_D	—	—

11.3 MDU Registers

11.3.1 MDU_CR (0xC1)

Bit	7	6	5	4	3	2	1	0
Name	MDUBUSY	MDUDONE	MDURUN	RSV				
Type	R	W1	W1	—	—	—	—	—
Reset	0	0	0	—	—	—	—	—

Bit	Name	Description
[7]	MDUBUSY	MDU Busy Flag MDU starts after MDU_C[7:0] is written. 0: MDU idle 1: MDU operating
[6]	MDUDONE	MDU Operation End Bit 0: No effect 1: This bit is set to 1 by software after MDU ends its operation. This operation ensures that MDU calculation is correct when it is invoked in different interrupt services and master programs.
[5]	MDURUN	MDU Start Operation Bit 0: No effect 1: This bit is set to 1 by software before MDU starts. This operation ensures that the MDU calculation is correct when it is invoked in different interrupt services and master programs.
[4:0]	RSV	Reserved

11.3.2 MDU_MD (0xCA)

Bit	7	6	5	4	3	2	1	0
Name	RSV					MDUMOD		
Type	—	—	—	—	—	R/W	R/W	R/W
Reset	—	—	—	—	—	0	0	0

Bit	Name	Description
[7:3]	RSV	Reserved
[2:0]	MDUMOD	MDU Mode Selection 000: 16-bit signed multiplication with the result shifted left by 1 bit 001: 16-bit signed multiplication 010: 16-bit unsigned multiplication 011: 32-bit/16-bit unsigned division 100: Coordinate transformation (sin/cos calculation) 101: Arctangent function 110: Low-pass filter 111: Reserved

11.3.3 MDU_A (0xC7, 0xC6)

MDU_AH(0xC7)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MDU_A[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MDU_AL(0xC6)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MDU_A[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MDU_A	Data register A of MDU. The contents of this register in different modes are shown as below. Table 11-8 MDU_A contents in Different Modes

MDU_MD[2:0]	Write	Read
000	Multiplied number	High-order 16 bits of the product
001	Multiplied number	High-order 16 bits of the product
010	Multiplied number	High-order 16 bits of the product
011	High-order 16 bits of dividend	High-order 16 bits of quotient
100	cos_i	cos_o
101	$Ucos\theta$	U
110	X_k	—

11.3.4 MDU_B (0xC5, 0xC4)

MDU_BH(0xC5)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MDU_B[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MDU_BL(0xC4)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MDU_B[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MDU_B	Data register B of MDU. The contents of this register in different modes are shown as below. Table 11-9 MDU_B contents in Different Modes

MDU_MD[2:0]	Write	Read
000	—	Low-order 16 bits of the product
001	—	Low-order16 bits of the product
010	—	Low-order 16 bits of the product
011	Low-order 16 bits of the dividend	Low-order 16 bits of the quotient
100	θ	—
101	—	θ
110	Y_{k-1}[31:16]	Y_k[31:16]

11.3.5 MDU_C (0xC3, 0xC2)

MDU_CH(0xC3)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MDU_C[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MDU_CL(0xC2)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MDU_C[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MDU_C	Data register C of MDU. The contents of this register in different modes are shown as below.

Table 11-10 MDU_C contents in Different Modes

MDU_MD[2:0]	Write	Read
000	Multiplier	—
001	Multiplier	—
010	Multiplier	—
011	Divisor	Remainder
100	sin_i	sin_o
101	$Usin\theta$	—
110	Y_{k-1}[15:0]	Y_k[15:0]

11.3.6 MDU_D (0xCB)

Bit	7	6	5	4	3	2	1	0
Name	MDU_D							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	MDU_D	D data register of MDU MDU_MD[2:0] = 110: *K* of LPF

12 PI/PID

12.1 PI/PID Introduction

PI/PID regulator is a linear controller, where the output is generated by linear combination of error proportional (P), integral (I) and differential (D) actions, and then implemented by an actuator. In motor control systems, it is used for speed and position control.

PI formula:

$$U_k = U_{k-1} + Kp \times (E_k - E_{k-1}) + Ki \times E_k$$

PID formula:

$$U_k = U_{k-1} + Kp \times (E_k - E_{k-1}) + Ki \times E_k + Kd \times (E_k - 2 \times E_{k-1} + E_{k-2})$$

Where,

U_k: Output for round k of calculation

U_{k-1}: Output for round of k - 1 calculation

E_k: Deviation for round k of input

E_{k-1}, E_{k-2}: Deviations for round k-1 and round k-2 of inputs

Kp, Ki, Kd: Proportional, integral and differential coefficients of regulator

Maximum value of U_k is presented as PIx_UKMAX (x = 0 ~ 3) and the minimum value as PIx_UKMIN.

12.2 PI/PID Features

- Parameter range is configurable
- Support multiple invocations but not with nested interrupt
- Produce a 32-bit result PIx_UK
- Read the results after the busy flag is reset to" "0

12.3 PI/PID Operations

1. Initialize MDU before the operations, and configure Kp, Ki, Kd and the maximum and minimum values of U_k;
2. Set PI_CR[PIxSTA] = 1 to start PI/PID operation. The busy flag PI_CR[PIBSY] is automatically set to "1";
3. Read PI_CR[PIBSY] in software. PI_CR[PIBSY] reading of 0 indicates that the calculation is completed and calculation result PIx_UK is updated;
4. Read PIx_UK to obtain the output.

Notes:

- The data format of PI_KP is Q12 and that of other registers is Q15.
- PIx_UK and PIx_EK1 values default to the last calculated UK and Ek. The related values change after

PIx_EK1 and PIx_UK are written.

- When PI controller is invoked repeatedly, related parameters shall be saved after each PI operation and initialized before next PI operation. Initialization codes are shown as below:

```
PIx_KP = KP;                    //Initialize Kp
PIx_KI = KI;                    //Initialize Ki
PIx_KD = KD;                    //Initialize Kd
PIx_UKMAX = UKMAX;              //Initialize maximum output
PIx_UKMIN = UKMIN;              //Initialize minimum output
PIx_EK1 = X;                    //Initialize Ek-1
PIx_UKH = Y1;                   //Initialize 16 high bits of Uk-1
PIx_UKL = Y2;                   //Initialize 16 low bits of Uk-1
```

12.4 PI/PID Registers

12.4.1 PI_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2TSS	RSV		PIBSY	PI3STA	PI2STA	PI1STA	PI0STA
Type	R/W	—	—	W	W	W	W	W
Reset	0	—	—	0	0	0	0	0

Bit	Name	Description
[7]	T2TSS	Input Mode Selection for Timer2 Step Motor 0: P1.0 is the direction line, and P0.7 is the pulse counting line 1: P1.0 is reverse pulse counting line, and P0.7 is forward pulse counting line
[6:5]	RSV	Reserved
[4]	PIBSY	PI Busy Flag 0: PI idle 1: PI active
[3]	PI3STA	PI3 Enable 0: Disable 1: Enable
[2]	PI2STA	PI2 Enable 0: Disable 1: Enable
[1]	PI1STA	PI1 Enable 0: Disable 1: Enable
[0]	PI0STA	PI0 Enable 0: Disable 1: Enable

12.4.2 PI0_KP (0x02E0, 0x02E1)

PI0_KPH(0x02E0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_KPL(0x02E1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_KP	Proportional coefficient of PI0

12.4.3 PI0_KI (0x02E2, 0x02E3)

PI0_KIH(0x02E2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_KIL(0x02E3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_KI	Integral coefficient of PI0

12.4.4 PI0_UKMAX (0x02E4, 0x02E5)

PI0_UKMAXH(0x02E4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKMAXL(0x02E5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_UKMAX	Maximum output of PI0

12.4.5 PI0_UKMIN (0x02E6, 0x02E7)

PI0_UKMINH(0x02E6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKMINL(0x02E7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_UKMIN	Minimum output of PI0

12.4.6 PI0_EK1 (0x02E8, 0x02E9)

PI0_EK1H(0x02E8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_EK1L(0x02E9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_EK1	Previous input deviation of PI0

12.4.7 PI0_EK (0x02EA, 0x02EB)

PI0_EKH(0x02EA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_EKL(0x02EB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_EK	Present input deviation of PI0

12.4.8 PI0_UKH (0x02EC, 0x02ED)

PI0_UKHH(0x02EC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKHL(0x02ED)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_UKH	16 high-order bits of PI0 output

12.4.9 PI0_UKL (0x02EE, 0x02EF)

PI0_UKLH(0x02EE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI0_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKLL(0x02EF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI0_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI0_UKL	16 low-order bits of PI0 output

12.4.10 PI1_KP (0x02D0, 0x02D1)

PI1_KPH(0x02D0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_KPL(0x02D1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_KP	Proportional coefficient of PI1

12.4.11 PI1_KI (0x02D2, 0x02D3)

PI1_KIH(0x02D2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_KIL(0x02D3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_KI	Integral coefficient of PI1

12.4.12 PI1_UKMAX (0x02D4, 0x02D5)

PI1_UKMAXH(0x02D4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKMAXL(0x02D5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_UKMAX	Maximum output of PI1

12.4.13 PI1_UKMIN (0x02D6, 0x02D7)

PI1_UKMINH(0x02D6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKMINL(0x02D7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_UKMIN	Minimum output of PI1

12.4.14 PI1_EK1 (0x02D8, 0x02D9)

PI1_EK1H(0x02D8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_EK1L(0x02D9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_EK1	Previous input deviation of PI1

12.4.15 PI1_EK (0x02DA, 0x02DB)

PI1_EKH(0x02DA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_EKL(0x02DB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_EK	Present input deviation of PI1

12.4.16 PI1_UKH (0x02DC, 0x02DD)

PI1_UKHH(0x02DC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKHL(0x02DD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_UKH	16 high-order bits of PI1 output

12.4.17 PI1_UKL (0x02DE, 0x02DF)

PI1_UKLH(0x02DE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI1_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKLL(0x02DF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI1_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI1_UKL	16 low-order bits of PI1 output

12.4.18 PI2_KP (0x02BC, 0x02BD)

PI2_KPH(0x02BC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KPL(0x02BD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_KP	Proportional coefficient of PI2

12.4.19 PI2_KI (0x02BE, 0x02BF)

PI2_KIH(0x02BE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KIL(0x02BF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_KI	Integral coefficient of PI2

12.4.20 PI2_UKMAX (0x02C0, 0x02C1)

PI2_UKMAXH(0x02C0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKMAXL(0x02C1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_UKMAX	Maximum output of PI2

12.4.21 PI2_UKMIN (0x02C2, 0x02C3)

PI2_UKMINH(0x02C2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKMINL(0x02C3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_UKMIN	Minimum output of PI2

12.4.22 PI2_EK1 (0x02C4, 0x02C5)

PI2_EK1H(0x02C4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EK1L(0x02C5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_EK1	Previous input deviation of PI2

12.4.23 PI2_EK (0x02C6, 0x02C7)

PI2_EKH(0x02C6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EKL(0x02C7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_EK	Present input deviation of PI2

12.4.24 PI2_UKH (0x02C8, 0x02C9)

PI2_UKHH(0x02C8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKHL(0x02C9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_UKH	16 high-order bits of PI2 output

12.4.25 PI2_UKL (0x02CA, 0x02CB)

PI2_UKLH(0x02CA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKLL(0x02CB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_UKL	16 low-order bits of PI2 output

12.4.26 PI2_KD (0x02CC, 0x02CD)

PI2_KDH(0x02CC)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_KD[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KDL(0x02CD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_KD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_KD	Differential coefficient of PI2

12.4.27 PI2_EK2 (0x02CE, 0x02CF)

PI2_EK2H(0x02CE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_EK2[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EK2L(0x02CF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_EK2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI2_EK2	Previous input deviation of PI2

12.4.28 PI3_KP (0x02A8, 0x02A9)

PI3_KPH(0x02A8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KPL(0x02A9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_KP	Proportional coefficient of PI3

12.4.29 PI3_KI (0x02AA, 0x02AB)

PI3_KIH(0x02AA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KIL(0x02AB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_KI	Integral coefficient of PI3

12.4.30 PI3_UKMAX (0x02AC, 0x02AD)

PI3_UKMAXH(0x02AC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKMAXL(0x02AD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_UKMAX	Maximum output of PI3

12.4.31 PI3_UKMIN (0x02AE, 0x02AF)

PI3_UKMINH(0x02AE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKMINL(0x02AF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_UKMIN	Minimum output of PI3

12.4.32 PI3_EK1 (0x02B0, 0x02B1)

PI3_EK1H(0x02B0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EK1L(0x02B1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_EK1	Previous input deviation of PI3

12.4.33 PI3_EK (0x02B2, 0x02B3)

PI3_EKH(0x02B2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EKL(0x02B3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_EK	Present input deviation of PI3

12.4.34 PI3_UKH (0x02B4, 0x02B5)

PI3_UKHH(0x02B4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKHL(0x02B5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_UKH	16 high-order bits of PI3 output

12.4.35 PI3_UKL (0x02B6, 0x02B7)

PI3_UKLH(0x02B6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKLL(0x02B7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_UKL	16 low-order bits of PI3 output

12.4.36 PI3_KD (0x02B8, 0x02B9)

PI3_KDH(0x02B8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	**15**							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KDL(0x02B9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_KD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_KD	Differential coefficient of PI3

12.4.37 PI3_EK2 (0x02BA, 0x02BB)

PI3_EK2H(0x02BA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_EK2[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EK2L(0x02BB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_EK2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	PI3_EK2	Previous input deviation of PI3

13 FOC

13.1 FOC Overview

13.1.1 FOC Introduction

FOC module is used in sensorless and sensored FOC motor drive applications and SVPWM-based motor control applications. When DRV_CR[FOC_EN] = 0, FOC module is inactivated and FOC clock stops. The relevant registers are forced into the reset state and cannot be written.

FOC module consists of angle estimator, PI controller, coordinate transformation module, current sampling module and PWM output module, which implements current closed loop in hardware. The angle estimator uses the sampling motor current to estimate the rotor position and implement sensorless FOC-based motor control. MCU can also process signals from the position sensor to obtain the rotor position and implement sensored FOC-based motor control.

- Sensorless FOC: Angle for coordinate transformation is obtained by angle estimator, and the motor speed is estimated for speed closed-loop control.
- Sensored FOC: FOC module provides the angle input interface. MCU samples position sensor signal and calculates the electrical angle of motor. Software sends the result to FOC module for coordinate transformation.

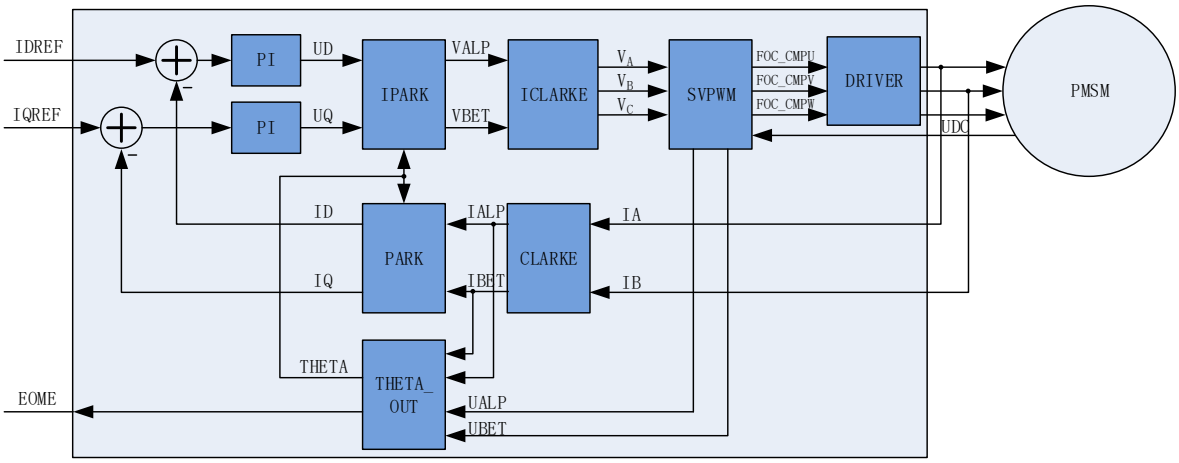


Figure 13-1 Schematic Diagram of FOC

13.1.2 Reference Voltage (VREF) Input

The current loop of FOC module uses the d-axis current reference value FOC_IDREF and the q-axis current reference value FOC_IQREF as the reference, and uses the d-axis current sampling value FOC__ID and the q-axis current sampling value FOC__IQ as the feedback. FOC module provides real-time estimated motor speed FOC__EOME. MCU can use FOC__EOME as the feedback to build speed loop and send the output of speed loop to FOC_IQREF to implement the speed-current dual closed loop control.

13.1.3 PI Controller

FOC module integrates 2 PI controllers:

1. Flux control: PI controller of d-axis current, with current reference FOC_IDREF minus feedback current FOC__ID as the error input, proportional coefficient FOC_DQKP and the integral coefficient FOC_DQKI for adjustment of PI performance, and FOC_DMAX and FOC_DMIN for limiting of the output amplitude. The output is voltage reference of d-axis FOC__UD;
2. Torque control: PI controller of q-axis current, with current reference FOC_IQREF minus feedback current FOC__IQ as the error input, proportional coefficient FOC_DQKP and the integral coefficient FOC_DQKI for adjustment of PI performance, and FOC_QMAX and FOC_QMIN for limiting of the output amplitude. The output is voltage reference of q-axis FOC__UQ.

13.1.4 Coordinate Transformations

13.1.4.1 Inverse Park Transformation

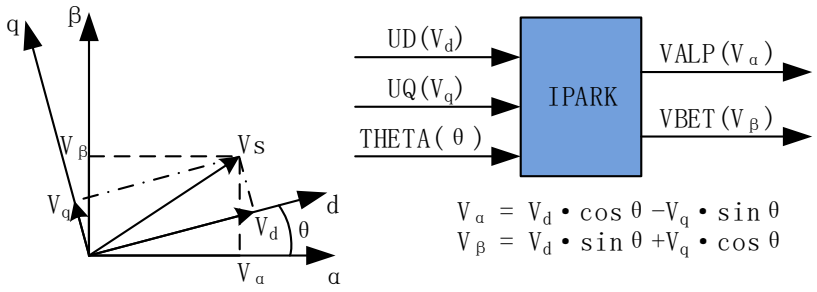


Figure 13-2 Inverse Park Transformation

Inverse Park transformation is used to transform two voltage vectors obtained by PI controller, FOC__UD and FOC__UQ, from dq-axis coordinate to αβ-axis coordinate.

13.1.4.2 Inverse Clarke Transformation

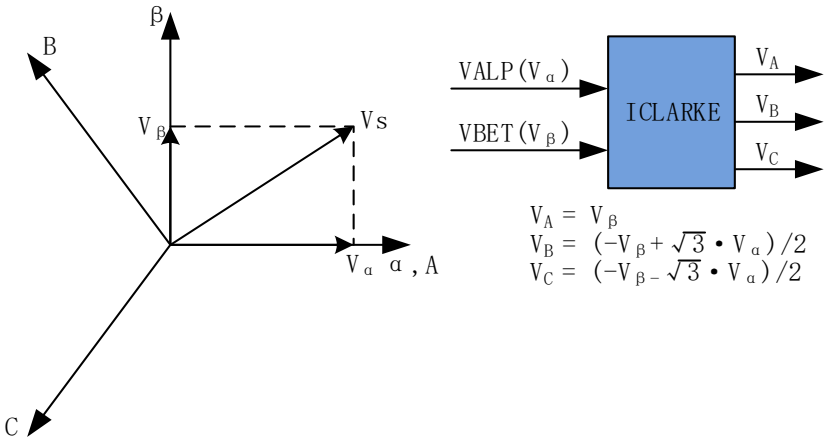


Figure 13-3 Inverse Clarke Transformation

Inverse Clarke transformation is used to transform voltage vector from αβ-axis coordinate to 3-phase stationary coordinate.

13.1.4.3 Clarke Transformation

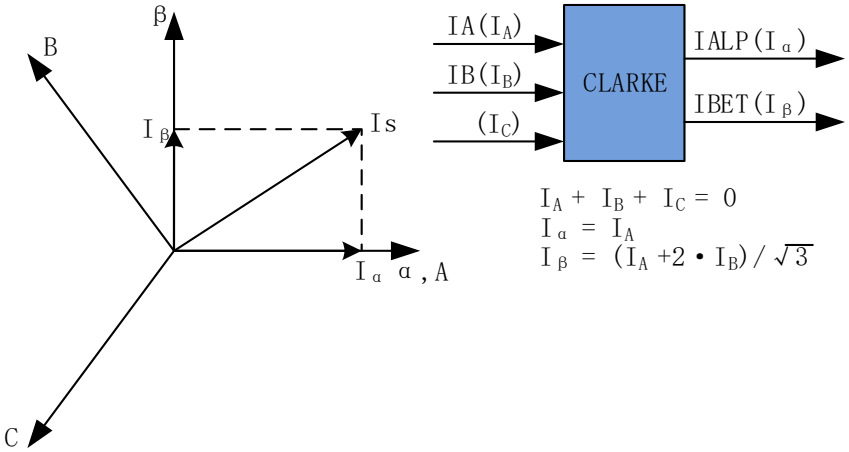


Figure 13-4 Clarke Transformation

Clarke transformation is used to transform the sampled current from 3-phase stationary coordinate to αβ-axis coordinate.

13.1.4.4 Park Transformation

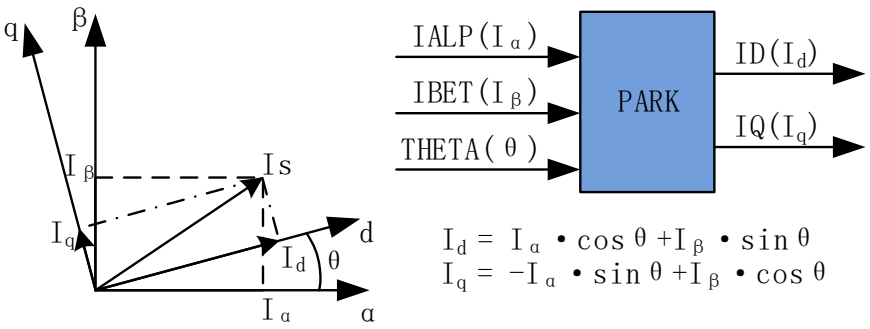


Figure 13-5 Park Transformation

Park transformation is used to transform the current vectors, obtained after Clarke transformation, from αβ-axis coordinate to dq-axis coordinate to get the sampled dq-axis current FOC__ID and FOC__IQ.

13.1.5 SVPWM

SVPWM algorithm is an important part of FOC. The main idea is to obtain quasi-circular rotating magnetic field by switching the inverter space voltage vectors. This method decreases harmonic components of the phase current, harmonic losses of the motor and torque ripple, and achieves high voltage utilization.

SVPWM generates pulse-width modulation signals for the 3-phase motor voltage control, whose process can be reduced to a few simple equations. Since high side and low side of the inverter cannot be turned on simultaneously, there are two states for a phase, i.e., phase connected to bus voltage (represented by 1) or phase connected to ground (represented by 0). Therefore, voltage vector output of the inverter has a total of $2^3 = 8$ possible states. $X_C X_B X_A$ represents the voltage vectors, where X_C represents the state of C phase, X_B represents the state of B phase and X_A represents the state of A phase. For example, 100 represents the state that C phase voltage is connected to bus voltage and A, B phases are connected to ground. When the states of 3-phase are all 1 or 0, there is no voltage drop between two phases and the state is called inactive state or zero voltage vector. The other 6 states which have voltage output are active voltage vectors with an adjacent state rotation offset of 60 degrees.

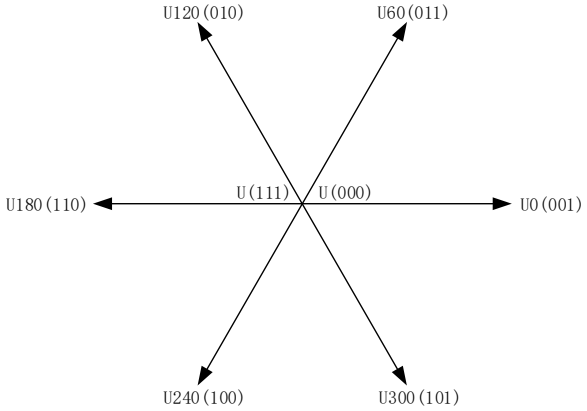


Figure 13-6 SVPWM Voltage Vector

SVPWM uses the sum of two adjacent vectors to generate any voltage vector located in the voltage vector space. As shown in Figure 13-7, U_{OUT} is the desired vector and it is in the sector between U60 and U0. Based on the principle of equal impulse, the effect, U0 applied 2*T1 time and U60 applied 2*T2 time, is equivalent to the U_{OUT}. The rest of time (T0) is applied by zero voltage vector.

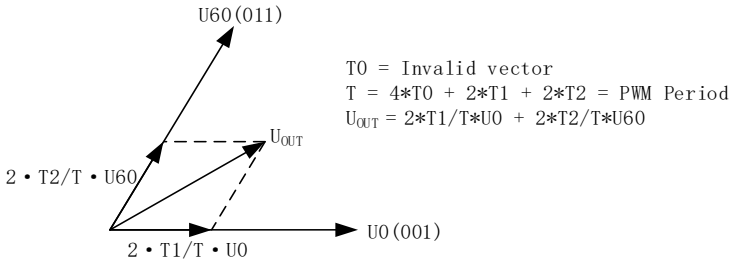


Figure 13-7 Voltage Vector Synthesis

Table 13-1 Inverter States of SVPWM

C phase	B phase	A phase	U_{ALP}	U_{BET}	State
0	0	0	0	0	000
0	0	1	$2/3*U_{DC}$	0	001
0	1	1	$1/3*U_{DC}$	$1/3*U_{DC}$	011
0	1	0	$-1/3*U_{DC}$	$1/3*U_{DC}$	010
1	1	0	$-2/3*U_{DC}$	0	110
1	0	0	$-1/3*U_{DC}$	$-1/3*U_{DC}$	100
1	0	1	$1/3*U_{DC}$	$-1/3*U_{DC}$	101
1	1	1	0	0	111

13.1.5.1 Continuous SVPWM

In single-shunt current sampling mode, Continuous SVPWM is always used. In dual/triple-shunt current

sampling mode, FOC_CR2[F5SEG] is set to “0” to select Continuous SVPWM as the output mode.

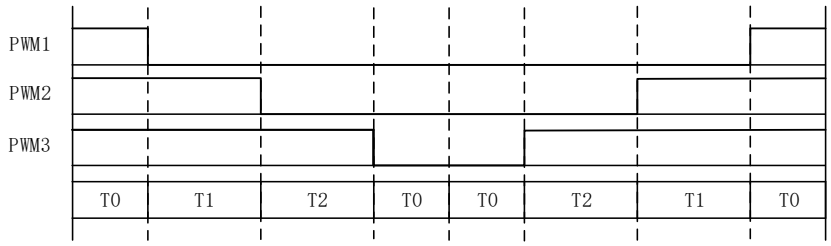


Figure 13-8 Continuous SVPWM Output Waveform

13.1.5.2 Discontinuous SVPWM

Discontinuous SVPWM is available in the dual/triple-shunt current sampling mode. FOC_CR2[F5SEG] is set to 1 to activate this mode.

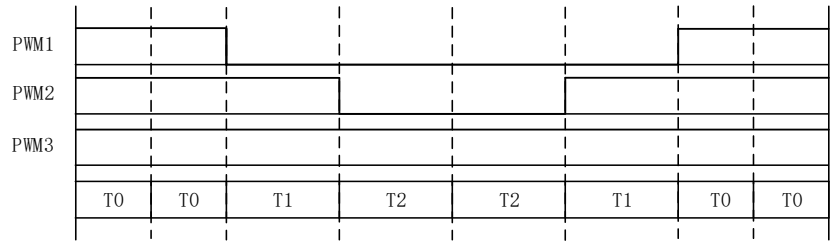


Figure 13-9 Discontinuous SVPWM Output Waveform

13.1.6 Overmodulation

Overmodulation is available in single/dual/triple-shunt current sampling mode. FOC_CR1[OVMDL] is set to 1 to activate this mode. The output, FOC__UD and FOC__UQ, and limit value, FOC_DMAX and FOC_QMIN, will be multiplied by 1.15 in this mode.

13.1.7 Deadtime Compensation

Deadtime compensation is available in dual/triple-shunt current sampling mode. The compensation value of deadtime is configured by FOC_TSMIN. This mode improves the quality of phase current at low speed.

13.1.8 Current and Voltage Sampling

In FOC mode, bus voltage and phase current are sampled by hardware automatically. Before the FOC module operates, ADC and operational amplifier shall be enabled and the corresponding control registers be configured. No configuration is required for ADC channel and mode. Current sampling mode, single/dual/triple-shunt, is selected by setting FOC_CR1[CSM]. In single-shunt mode, default sampling channel of the bus current (itrip) is ADC channel 4. In dual-shunt mode, default sampling channels of A phase current (ia) and B phase current (ib) are ADC channel 0 and channel 1 respectively. In triple-shunt mode, default sampling channels of ia, ib and C

phase current (ic) are ADC channel 0, channel 1 and channel 4 respectively. Channel 14, with built-in voltage divider to sample VCC directly, or channel 2 can be selected for bus voltage sampling.

13.1.8.1 Single-shunt Current Sampling Mode

FOC_CR1[CSM] is set to 00 to select the single-shunt current sampling mode. In this mode, FOC module samples itrip twice during the DRV timer counting-up operation, and samples the bus voltage during the DRV timer counting-down operation after FOC module completes the calculation.

Since deadtime affects the accuracy of current sampling, FOC module samples within T1' and T2', which is the applied time of active voltage vector with deadtime removed. FOC_TRGDLY is the register which moves the current sampling time, and this register shall be configured reasonably to ensure sampling is completed within T1' and T2'. For example, FOC_TRGDLY = 5, the sampling time is delayed for 5*T = 208ns; FOC_TRGDLY = 0xFB(–5), the sampling time is advanced for 5*T = 208ns.

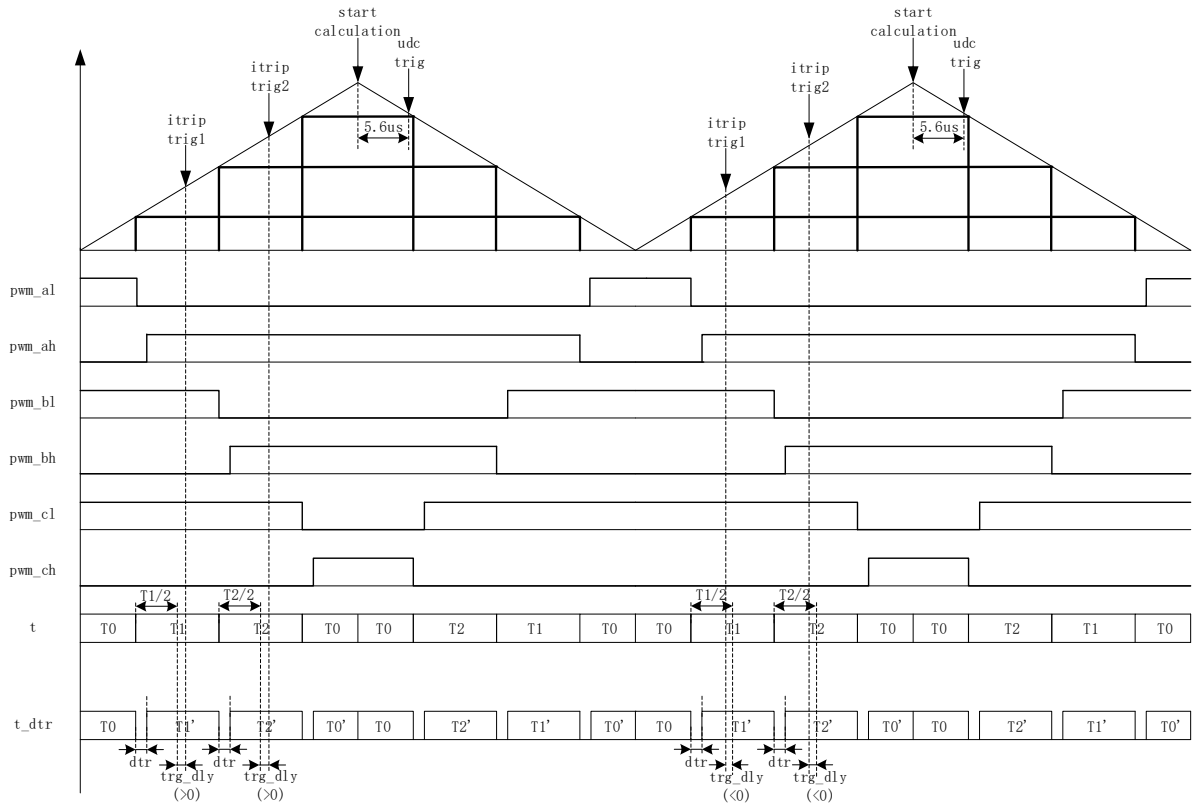


Figure 13-10 Single-shunt Current Sampling Timing Diagram

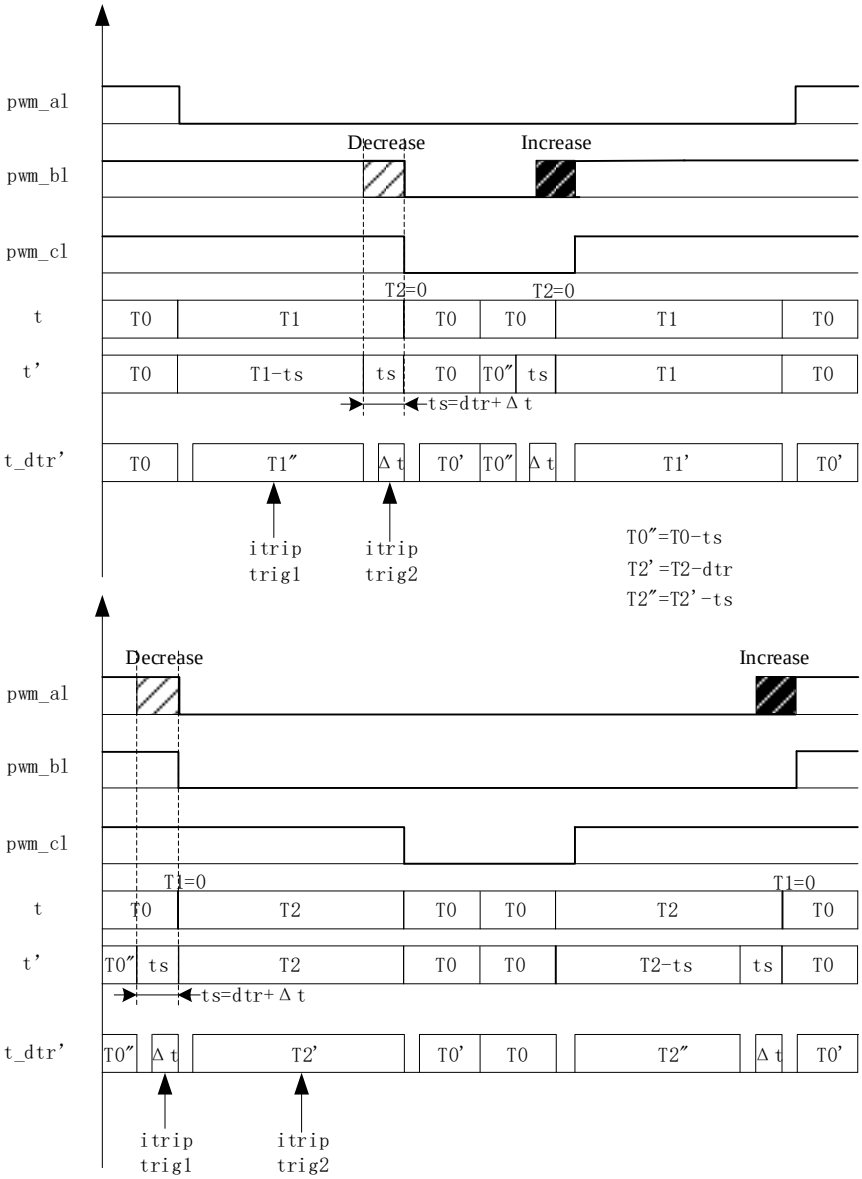


Figure 13-11 PWM Wave Shift in Single-shunt Current Sampling Mode

The time of sampling window may be not enough to sample the current in low modulation index and sector switching area. PWM waveform shall be adjusted to ensure the minimum sampling window required in the case. FOC_TSMIN (FOC_TSMIN = minimum sampling window + deadtime) is used to configure the minimum active voltage vector applied time, and FOC module adjusts the PWM waveform automatically.

13.1.8.2 Dual/Triple-shunt Current Sampling Mode

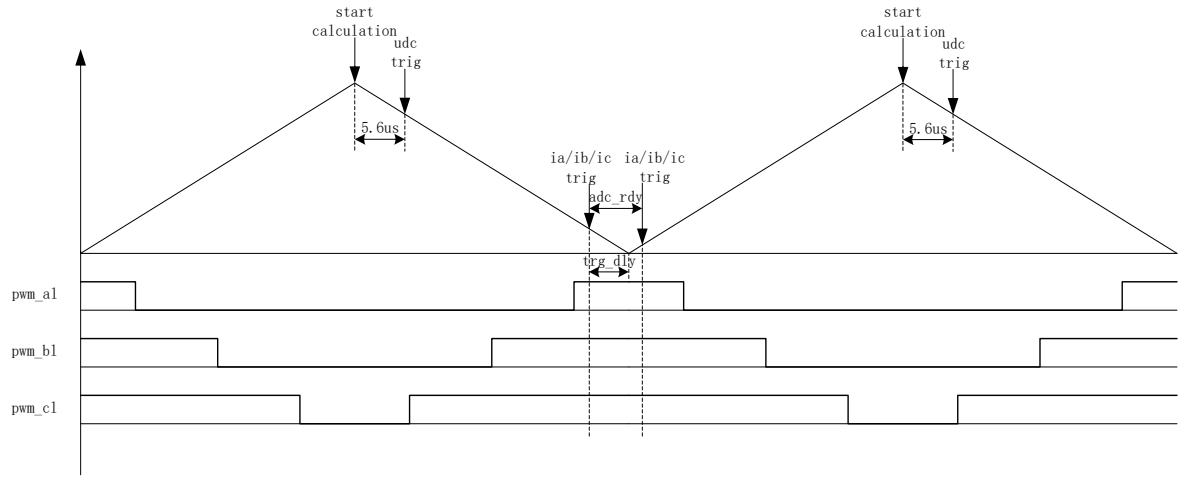


Figure 13-12 Dual/Triple-shunt Sequential Current Sampling Mode

FOC_CR1[CSM] is set to 10/11 and FOC_CR2[DSS] to 0 to select dual/triple-shunt concurrent current sampling mode. In triple-shunt mode, FOC_TRGDLY is used to configure the sampling time of a phase current (ia/ib/ic is determined according to the sector), and other phases are sampled at the end of previous sampling. In dual-shunt mode, FOC_TRGDLY is used to configure the sampling time of ia, and ib is sampled at the end of ia sampling. TRG_DLY shall be configured reasonably to ensure current sampling time is within zero voltage vector (000). For example, when FOC_TRGDLY = 0xB2 and FOC counter counts down, ia/ib/ic is sampled at 50*T = 2.08μs before an underflow event, and then the other phases of ia/ib/ic are sampled.

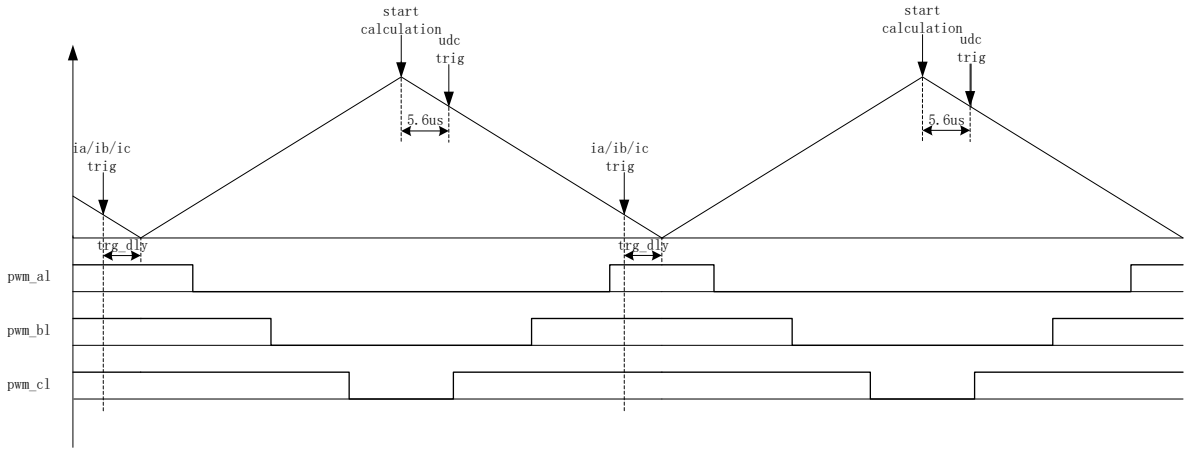


Figure 13-13 Dual/Triple-Shunt Alternating Current Sampling Mode

FOC_CR1[CSM] is set to 10/11 and FOC_CR2[DSS] to 1 to select dual/triple-shunt alternating current sampling mode. In this mode, FOC module performs calculation in every PWM cycle. However, only one phase current is sampled at each PWM cycle (ia/ib/ic is determined according to the sector). The first carrier cycle samples one phase of the ia/ib/ic, and the second carrier cycle samples the current of the other phase,

so as to alternately sample the current of two phases in three phases. FOC_TRGDLY is used to configure the sampling time of ia (channel 0), ib (channel 1) and ic (channel 4). TRG_DLY shall be configured reasonably to ensure sampling time for the current of ia (Channel 0), ib (Channel 1) or ic (Channel 4) is within zero voltage vector (000). For example, when FOC_TRGDLY = 0xB2 and FOC counter counts-down, phase current is sampled at 50*T = 2.08μs before an underflow event.

In dual/triple-shunt current sampling mode, bus voltage is sampled when driver counter is down-counting and FOC module completes the calculation.

13.1.8.3 Current Sampling Offset

The current sampling offset voltage shall be added to sample full range of current due to the existence of the positive and negative phase current. When phase current is 0, ADC result is the offset value. ADC result minus this value, 0x4000 default, is the sampling current. Since ADC reference voltage and hardware are nonideal, there is a deviation between the default value and the real value. Therefore, it is necessary to calibrate the offset. The calibration procedure is as follows. When there is no current in three phases, MCU starts to sample the corresponding channel and the results, averaging all the sampled value, is written to FOC_CSO. Providing ADC sampling range is 0 ~ 5V and the offset is 2.5V, FOC_CSO = 2.5V/5V*32768= 16384 (0x4000).

- When FOC_CHC[CSOC] = 00/11, FOC_CSO is written to modify the offset of itrip and ic.
- When FOC_CHC[CSOC] = 01, FOC_CSO is written to modify the offset of ia.
- When FOC_CHC[CSOC] = 10, FOC_CSO is written to modify the offset of ib.

13.1.9 Angle Mode

Angle module includes angle estimation module, ramping module and estimated angle smooth switching module. The sources of angle are as follows:

- Forced ramping angle
- Forced pulling angle
- Estimated angle of estimator
- Forced angle of estimator

Table 13-2 Sources of Angle

FOC_CR1[RFAE]	FOC_CR1[ANGM]	FOC_CR1[EFAE]	Sources of Angle
1	X	X	Forced ramping angle
0	0	X	Forced pulling angle
0	1	0	Estimated angle of estimator
0	1	1	omega > FOC_EFREQMIN: Estimated angle of estimator omega < FOC_EFREQMIN: Forced angle of estimator

13.1.9.1 Forced Ramping Angle

Forced ramping angle is controlled by angle register FOC__THETA, speed register FOC__RTHESTEP, acceleration register FOC_RTHEACC and ramping timer FOC_RTHECNT. The formula is:

FOC__RTHESTEP (32-bit) = FOC__RTHESTEP (32-bit) + FOC_RTHEACC (low 16 bits)

THETA_OL (16-bit) = THETA_OL (16-bit) + FOC__RTHESTEP (high 16 bits)

Where, THETA_OL is an internal variable of the chip. In forced ramping angle mode, THETA_OL is written to FOC__THETA as the used angle. If the software writes a value to FOC__THETA, this value is written to THETA_OL as well.

Forced ramping angle has the highest priority. Ramping module makes a ramping operation in every PWM cycle and the counter is added by 1. When the value of the counter reaches the set value by FOC_RTHECNT, FOC_CR1[RFAE] is cleared by hardware, and then the ramping is completed.

Thereafter, according to the value of FOC_CR1[ANGM], the angle comes from estimator (FOC_CR1[ANGM] = 1) or forced pulling angle (FOC_CR1[ANGM] = 0).

13.1.9.2 Forced Pulling Angle

Forced pulling angle is controlled by angle register FOC__THETA, speed register FOC__RTHESTEP. The formula is:

THETA_OL (16-bit) = THETA_OL (16-bit) + FOC__RTHESTEP (high 16 bits)

Where, THETA_OL is an internal variable of the chip. In forced pulling angle mode, THETA_OL is written to FOC__THETA as the used angle. If the software writes a value to FOC__THETA, this value will be written to THETA_OL as well.

- When FOC_CR1[RFAE] is set to 1 and FOC_CR1[ANGM] to 0, MCU switches to forced pulling angle mode starts after forced ramping angle mode. The speed is the cumulative result after ramping. This mode implements a forced uniform speed control.
- When FOC_CR1[RFAE] is set to 0 and FOC_CR1[ANGM] to 0, the angle is the forced pulling angle and the speed of FOC__RTHESTEP is the initial speed written by software. Configuring FOC__RTHESTEP to “0” enables the pre-position feature. The sensor-based FOC is implemented after the motor speed is set with RTHESTEP. (Principle of Sensor-based FOC: The angle and speed are written to FOC__THETA and FOC__RTHESTEP by software, and FOC module generates an angle in each PWM cycle based on the written values.)

13.1.9.3 Estimator Angle

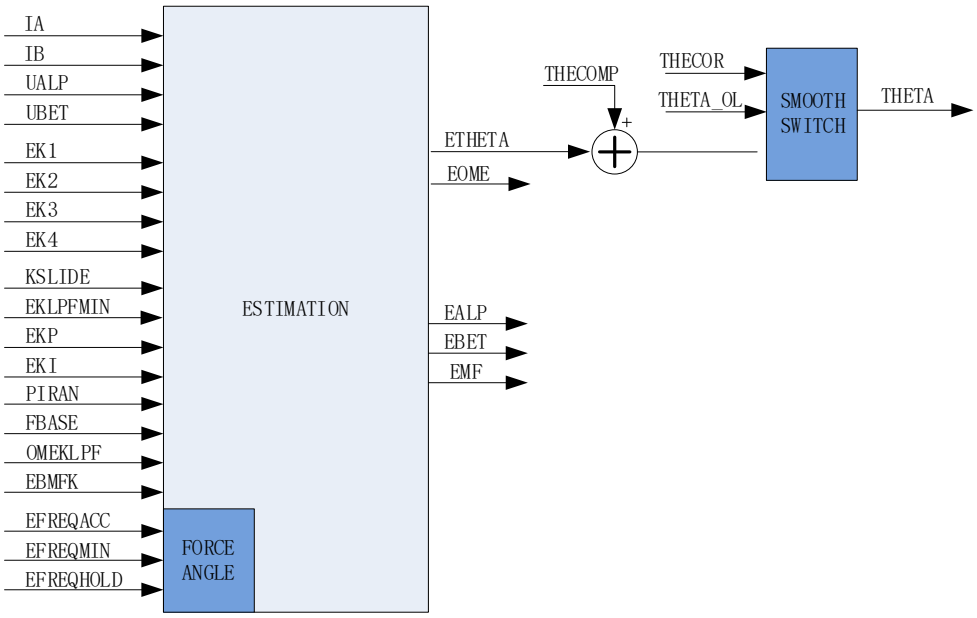


Figure 13-14 Schematic Block Diagram of Estimator

13.1.9.3.1 Estimated Angle of Estimator

The estimator builds the motor model based on the motor parameters and control parameters, and outputs the estimated angle based on the sampling current and output voltage. The estimator operates in PLL mode or SMO mode by configuring FOC_CR2[ESEL] bit.

13.1.9.3.2 Forced Angle of Estimator

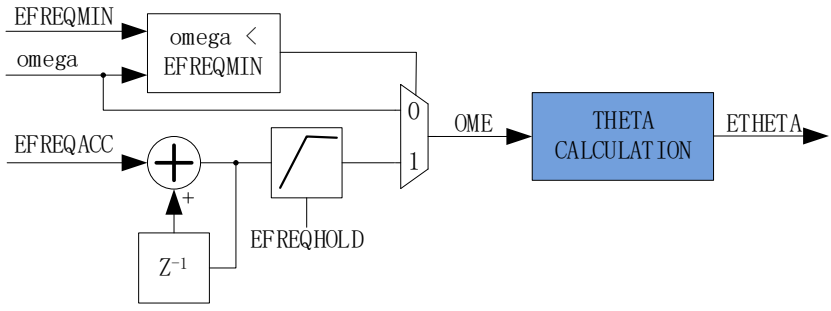


Figure 13-15 Schematic Block Diagram of Forced Angle of Estimator

This feature is similar to the ramping feature. Due to the low speed at motor starting process, there may be a deviation in angle and speed estimation with the small effective signal, resulting in startup failure. In this case, the estimator outputs the forced angle to ensure the motor start normally.

The forced angle feature of the estimator is enabled when FOC_CR1[RFAE] is set to 0, FOC_CR1[ANGM] to 1 and FOC_CR1[EFAE] to 1.As shown in Figure 13-15, the estimator compares the value of real-time estimated speed (omega) and FOC_EFREQMIN to determine omega or forced speed as the used speed (OME). When omega < FOC_EFREQMIN, the forced speed is selected as EOME. The forced speed starts with 0 and increases by FOC_EFREQACC in each PWM cycle, with the maximum value FOC_EFREQHOLD. When omega ≥ FOC_EFREQMIN, omega is selected as OME.

Estimated speed of the estimator FOC__EOME is the low-pass filtering result of OME with the coefficient set by FOC_OMEKLPF.

13.1.9.3.3 Angle Smooth Switching

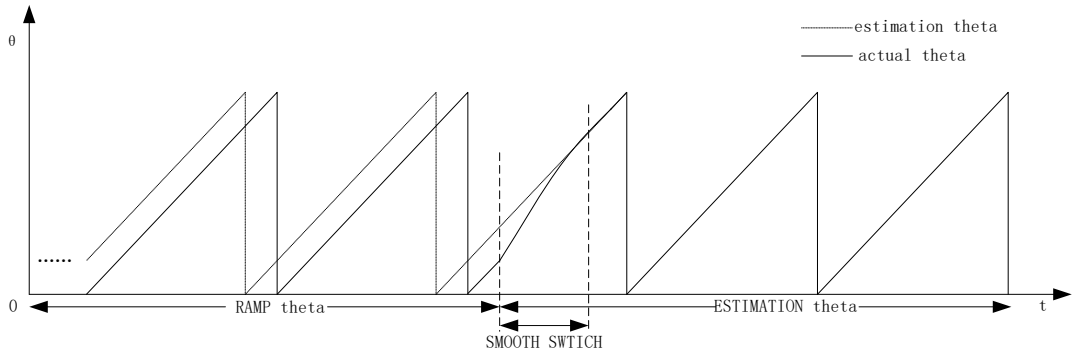


Figure 13-16 Angle Smooth Switching Waveform

When FOC_CR1[RFAE] is set to 1 and FOC_CR1[ANGM] to 1, the motor starts with ramping feature. At the end of the ramping, it switches to estimator angle mode. However, there is usually a deviation between the estimated angle (FOC__ETHETA) and the forced ramping angle (THETA_OL). If the angle is switched from forced ramping angle to estimated angle directly, motor jitter may occur due to such a sudden change. To deal with this problem, a smooth switching is preferred.

At the end of the ramping, if the deviation between FOC__ETHETA and THETA_OL is less than or equal to FOC_THECOR, FOC__ETHETA is selected as the output angle. But if the deviation is larger than FOC_THECOR, THETA_OL is modified smoothly with the step of FOC_THECOR at every PWM cycle until it is close to FOC__ETHETA. After the deviation is less than THECOR, FOC__ETHETA is selected as the output angle.

13.1.9.3.4 Angle Compensation

Angle compensation value FOC_THECOMP is used to compensate for the estimated angle FOC__ETHETA. If FOC_THECOMP is negative, the lagged angle is compensated; if it is positive, the advanced angle is compensated.

13.1.10 Motor Real Time Parameters

MCU monitors the state of motor using the following real time variables provided by FOC module:

- Used angle FOC__THETA
- Estimated angle FOC__ETHETA, Estimated speed FOC__EOME
- d-axis voltage FOC__UD, q-axis voltage FOC__UQ
- d-axis current FOC__ID, q-axis current FOC__IQ
- α-axis voltage FOC__VALP, β-axis voltage FOC__VBET
- Bus voltage FOC__UDCFLT
- Phase current FOC__IA, FOC__IB, FOC__IC and maximum phase current FOC__IAMAX, FOC__IBMAX, FOC__ICMAX
- α-axis current (equal to FOC__IA), β-axis current FOC__IBET
- α-axis BEMF FOC__EALP, β-axis BEMF FOC__EBET
- Magnitude of BEMF FOC__EMF
- Motor power FOC__POW

13.1.10.1 RSD

FOC provides RSD (detection of rotating state: upwind and downwind) feature. FOC module starts to operate when FOC_CR0[ESCMS] is set to 1, FOC_IDREF to 0, and FOC_IQREF to 0. Motor's rotor state is detected by FOC__ETHETA and FOC__EOME. If FOC__ETHETA decreases or FOC__EOME is a negative value, the motor rotates upwind and it is necessary to brake first and then start the motor with ramping forced angle mode. If FOC__ETHETA increases or FOC__EOME is a positive value, the motor rotates downwind and can be started using estimated angle directly.

13.1.10.2 BEMF Detection

Estimator estimates α-axis BEMF FOC__EALP and β-axis BEMF FOC__EBET with the motor parameters, and calculates the magnitude of BEMF (FOC__EMF) , which implements protection features, such as lock protection or phase unconnected protection.

13.1.10.3 Motor Power

FOC module calculates motor power based on the sampling current, modulation index of SVPWM and bus voltage.

13.1.11 FG Generation

FG signal is generated by FOC module and Timer4. FOC module calculates an FG value based on frequency base fbase, low-pass filtered speed FOC_EOMELPF and FG coefficient FOC_KFG in every PWM cycle. The

result is updated to TIM4_ARR automatically and half of the result (TIM4_ARR/2) to TIM4_DR by hardware. It shall be noted that Timer4 must work in output mode and the clock division factor of Timer4 shall be configured according to the motor maximum speed. FOC_KFG is set as: FOC_KFG = SYSCLK/(2^ TIM4_CR0[T4PSC]*fbase*x), where, x refers to the expected number of FG signal in one electric cycle. If the result exceeds 65535, the clock division factor TIM4_CR0[T4PSC] shall be adjusted.

When FOC_KFG = 0, this feature is disabled, and TIM4__ARR and TIM4__DR will not be updated.

13.2 FOC Registers

13.2.1 FOC_CR0 (0x409F)

Bit	7	6	5	4	3	2	1	0
Name	OMIF	OMAF	MERRS		UCSEL	OMAS	ESCMS	EDIS
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	OMIF	omega < FOC_EFREQMIN flag. This bit is valid even if FOC_CR1[EFAE] = 0 0: omega ≥ FOC_EFREQMIN 1: omega < FOC_EFREQMIN
[6]	OMAF	omega > FOC_EFREQMAX flag 0: omega ≤ FOC_EFREQMAX 1: omega > FOC_EFREQMAX
[5:4]	MERRS	The maximum error in SMO 00: 0.5 01: 0.25 10: 0.125 11: 1.0
[3]	UCSEL	Sampling Channel for Bus Voltage (UDC) In FOC mode, bus voltage is sampled automatically by hardware after Driver counter is enabled. This bit selects the ADC channel for bus voltage sampling 0: ADC channel 2 1: ADC channel 14 ADC channel 14 is the internal channel dedicated for bus voltage sampling, and the voltage division ratio is selected by configuring ADC_CR[ADCRATIO]. ADC channel 2 is the external bus voltage sampling channel, which uses external voltage resistor division Note: It is not necessary to set the associated Enable Bit in ADC_MASK register to “1”.
[2]	OMAS	Output selection when omega is too large When omega[15:8] is larger than FOC_EFREQMAX, OME is set as: 0: FOC_EFREQMAX*256 1: FOC_EFREQHOLD
[1]	ESCMS	Angle Output Mode Selection 0: Test mode 1: Recommended mode
[0]	EDIS	FOC__EALP/FOC__EBET Auto-computation Disabled 0: Not forbid 1: Forbid

13.2.2 FOC_CR1 (0x40A0)

Bit	7	6	5	4	3	2	1	0
Name	OVMDL	EFAE	RFAE	ANGM	CSM		RSV	SVPWMEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	—	R/W
Reset	0	0	0	0	0	0	—	0

Bit	Name	Description
[7]	OVMDL	Overmodulation Enable 0: Disable 1: Enable
[6]	EFAE	Forced Angle of Estimator Enable When this feature is enabled, angle mode is determined by the estimator, and it switches to estimated angle mode automatically. 0: Disable 1: Enable
[5]	RFAE	Forced Ramping Angle Enable When this feature is enabled, angle mode is determined by the ramping module. After ramping, it switches to estimated mode or forced pulling mode according to FOC_CR1[ANGM]. FOC_CR1[RFAE] is cleared to 0 by hardware as well. 0: Disable 1: Enable
[4]	ANGM	Angle Mode When FOC_CR1[RFAE] = 0, angle mode is determined by this bit. When FOC_CR1[RFAE] = 1, angle mode is determined by this bit after ramping.0: Forced pulling angle mode 1: Estimated Angle of Estimator mode
[3:2]	CSM	Current Sampling Mode 00: Single-shunt current sampling mode 01: Dual-shunt current sampling mode 10: Reserved 11: Triple--shunt current sampling mode
[1]	RSV	Reserved
[0]	SVPWMEN	SVPWM Enable 0: Disable 1: Enable

13.2.3 FOC_CR2 (0x40A1)

Bit	7	6	5	4	3	2	1	0
Name	ESEL	ICLR	F5SEG	DSS	CSOC		UQD	UDD
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	ESEL	Angle Estimator Mode Selection 0: SMO 1: PLL. FOC_KSILDE register is FOC_PLLKP of PI controller, and FOC_KLPFMIN register is FOC_PLLKI of PI controller.
[6]	ICLR	Clear FOC_IAMAX/FOC_IBMAX/FOC_ICMAX to “0” 0: No effect. 1: This bit is cleared to 0 by hardware after FOC_IAMAX/ FOC_IBMAX/ FOC_ICMAX is cleared to 0.
[5]	F5SEG	SVPWM Mode Selection 0: Continuous SVPWM 1: Discontinuous SVPWM (cannot be selected in single-shunt current sampling mode)
[4]	DSS	Current Sampling Discontinuous Mode Selection in Dual/Triple-shunt Mode 0: Sequential sampling mode, where current values of two phases are sampled in

		each carrier period. 1: Alternating sampling mode. FOC module completes the calculation in every PWM cycle. The current of one phase is sampled in each PWM cycle, and the current of two phases are sampled alternately in two adjacent PWM cycles.
[3:2]	CSOC	Current Sampling Offset Calibration Selection This bit is written to select the offset of FOC_CSO. In single-shunt sampling, 00 or 11 is written to calibrate itrip offset. In dual-shunt sampling, 01 is written to calibrate ia offset and 10 to calibrate ib offset. In triple-shunt sampling, 01 is written to calibrate ia offset, 10 to calibrate ib offset and 00 or 11 to calibrate ic offset. 00: itrip and ic 01: ia 10: ib 11: itrip and ic
[1]	UQD	q-Axis PI controller disabled, where FOC__UQ value is no longer updated by the PI controller. 0: Not forbid 1: Forbid
[0]	UDD	d-Axis PI controller disabled, where the FOC__UD value is no longer updated by the PI controller. 0: Not forbid 1: Forbid

13.2.4 FOC_TSMIN (0x40A2)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TSMIN							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_TSMIN	Single-shunt sampling mode: Minimum window for sampling Dual/triple-shunt sampling mode: Deadtime compensation Range: [0,255] TSMIN = sampling window T_{window} + deadtime T_{DT} Providing T_{window} = 1μs, T_{DT} = 1μs, TSMIN = 2μs, Ts = 62.5μs, FOC_TSMIN = (1 + 1)/62.5*4096 = 131

13.2.5 FOC_TGLI (0x40A3)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TGLI							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_TGLI	Narrow Pulse Elimination for High Side of the Bridge This feature is used in high voltage applications. The high side of bridge must be longer than a certain time. After this bit is configured, high side of the bridge is not turned on when the conducting time is less than this value. Range: [0,255] Providing to eliminate pulse narrower than 1μs, T_{DT} = 1μs, Ts = 62.5μs FOC_TGLI = (1 + 1)/62.5*4096 = 131

13.2.6 FOC_TBLO (0x40A4)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TBLO							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_TBLO	Sampling Masking Time in Triple-shunt Sampling Mode When low side of the bridge is turned on for less than FOC_TBLO, the current of this phase is not sampled and obtained through special process. Range: [0,255] Providing the current is not sampled when low side of the bridge is turned on for less than 1μs, FOC_TBLO = 1000ns/41.67ns = 24

13.2.7 FOC_TRGDLY (0x40A5)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TRGDLY							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_TRGDLY	Time Configuration for Current Sampling When FOC_TRGDLY is set to 0, FOC module samples the current as follows. Single-shunt sampling mode: Midpoint between deadtime and applied time of active voltage vector. Dual/triple-shunt sampling mode: Midpoint of voltage vector 000 (minima of Driver counter) Range: [–128,127] Single-shunt sampling mode: If FOC_TRGDLY = 5, it delays 5*T = 208ns to sample the current, and if FOC_TRGDLY = –5, it advances 5*T=208ns. Dual/triple-shunt sampling mode: If FOC_TRGDLY = -5 and Driver counter counts downward, it samples the current at 5*T = 208ns before an overflow event occurs. If OC_TRGDLY = 5 and Driver counter counts upward, it samples the current at 5*T = 208ns after an overflow event occurs.

13.2.8 FOC_CSO (0x40A6, 0x40A7)

FOC_CSOH(0x40A6)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_CSO[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0
FOC_CSOL(0x40A7)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_CSO[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_CSO	Current Sampling Offset FOC_CR2[CSOC] is configured to select the current, and FOC_CSO is written to calibrate current sampling offset of itrip in single-shunt mode, ia, ib in dual-shunt mode and ia, ib and ic in triple-shunt mode. Range: [0,32767]; MSB is always 0. Providing ADC sampling voltage range is 0V ~ 5V and offset is 2.5V, FOC_CSO = 2.5V/5V*32768 = 16384 (0x4000)

13.2.9 FOC__RTHESTEP (0x40A8, 0x40A9)

FOC__RTHESTEPH(0x40A8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__RTHESTEP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__RTHESTEPL(0x40A9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__RTHESTEP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__RTHESTEP	Speed of Ramping Module. FOC__RTHESTEP is an internal 32-bit variable. MSB is sign bit. High-order 16 bits are written by software. Range: [–32768,32767] FOC__RTHESTEP (32-bit) = FOC__RTHESTEP (32-bit) + FOC_RTHEACC (low 16 bits) THETA_OL (16-bit) = THETA_OL (16-bit) + FOC__RTHESTEP (high 16 bits)

13.2.10 FOC_RTHEACC (0x40AA, 0x40AB)

FOC_RTHEACCH(0x40AA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_RTHEACC[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_RTHEACCL(0x40AB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_RTHEACC[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_RTHEACC	Ramping Acceleration. FOC_RTHEACC is an internal 32-bit variable. MSB is sign bit. Low-order 16 bits are written by software, and high-order 16 bits are always 0. Range: [–32768,32767] FOC__RTHESTEP (32-bit) = FOC__RTHESTEP (32-bit) + FOC_RTHEACC (low 16 bits) THETA_OL (16-bit) = THETA_OL (16-bit) + FOC__RTHESTEP (high 16 bits)

13.2.11 FOC_EOMELPF (0x40AA, 0x40AB)

FOC_EOMELPFH(0x40AA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EOMELPF[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC_EOMELPFL(0x40AB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EOMELPF[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EOMELPF	Filtered Estimated Speed of Estimator The filter coefficient is FOC_EOMEKLPF and f the LPF frequency is the carrier period. Range: [–32768,32767]

13.2.12 FOC_RTHECNT (0x40AC)

Bit	7	6	5	4	3	2	1	0
Name	FOC_RTHECNT							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_RTHECNT	Ramping times = RTHECNT*256 When ramping feature is enabled, the ramping angle increases in each PWM cycle. After RTHECNT*256 times, ramping feature is disabled.

13.2.13 FOC_THECOR (0x40AD)

Bit	7	6	5	4	3	2	1	0
Name	FOC_THECOR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset	0	0	0	0	0	0	0	1

Bit	Name	Description
[7:0]	FOC_THECOR	Step Value of Angle Smooth Switching The step value of angle smooth switching after ramping. The format is the same as FOC__THETA. Range: [0,255]

13.2.14 FOC__EMF (0x40AE, 0x40AF)

FOC__EMFH(0x40AE)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__EMF[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__EMFL(0x40AF)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__EMF[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__EMF	BEMF Magnitude of Estimator This value is the root of sum of square of FOC__EALP and square of FOC__EBET; Range: [0,32767]

13.2.15 FOC_THECOMP (0x40AE, 0x40AF)

FOC_THECOMPH(0x40AE)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_THECOMP[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

FOC_THECOMPL(0x40AF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_THECOMP[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_THECOMP	Angle Compensation Value (write only) The estimated angle (FOC__ETHETA) plus the compensation value is the output angle of estimator (FOC__THETA). The format is the same as FOC__THETA. Range: [–32768,32767]

13.2.16 FOC_DMAX (0x40B0, 0x40B1)

FOC_DMAXH(0x40B0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_DMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DMAXL(0x40B1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_DMAX	Maximum Output of d-axis PI Controller Range: [–32768,32767]

13.2.17 FOC_DMIN (0x40B2, 0x40B3)

FOC_DMINH(0x40B2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_DMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DMINL(0x40B3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_DMIN	Minimum Output of d-axis PI Controller Range: [–32768,32767]

13.2.18 FOC_QMAX (0x40B4, 0x40B5)

FOC_QMAXH(0x40B4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_QMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_QMAXL(0x40B5)								

Bit	7	6	5	4	3	2	1	0
Name	FOC_QMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_QMAX	Maximum Output of q-axis PI Controller Range: [–32768,32767]

13.2.19 FOC_QMIN (0x40B6, 0x40B7)

FOC_QMINH(0x40B6)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_QMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_QMINL(0x40B7)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_QMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_QMIN	Minimum Output of q-axis PI Controller Range: [–32768,32767]

13.2.20 FOC__UD (0x40B8, 0x40B9)

FOC__UDH(0x40B8)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UD[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__UDL(0x40B9)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__UD	Voltage by d-axis PI Controller; Range: [–32768,32767]

13.2.21 FOC__UQ (0x40BA, 0x40BB)

FOC__UQH(0x40BA)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__UQL(0x40BB)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__UQ	Voltage by q-axis PI Controller Range: [–32768,32767]

13.2.22 FOC__ID (0x40BC, 0x40BD)

FOC__IDH(0x40BC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__ID[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IDL(0x40BD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__ID[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__ID	d-axis Current from Park Transformation; Range: [–32768,32767]

13.2.23 FOC__IQ (0x40BE, 0x40BF)

FOC__IQH(0x40BE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__IQ[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IQL(0x40BF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__IQ[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__IQ	q-axis Current from Park Transformation Range: [–32768,32767]

13.2.24 FOC__IBET (0x40C0, 0x40C1)

FOC__IBETH(0x40C0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__IBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IBETL(0x40C1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__IBET[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__IBET	β-axis Current from Clarke Transformation Range: [–32768,32767]

13.2.25 FOC__VBET (0x40C2, 0x40C3)

FOC__VBETH(0x40C2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__VBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__VBETL(0x40C3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__VBET[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__VBET	β-axis Output Voltage of FOC Module Range: [–32768,32767]

13.2.26 FOC__VALP (0x40C4, 0x40C5)

FOC__VALPH(0x40C4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__VALP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__VALPL(0x40C5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__VALP[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__VALP	α-axis Output Voltage of FOC Module Range: [–32768,32767]

13.2.27 FOC_UDCPS (0x40C2, 0x40C3)

FOC_UDCPSH(0x40C2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_UDCPS[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_UDCPSL(0x40C3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_UDCPS[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_UDCPS	Compensation Value for d-axis Voltage The result of d-axis PI controller (FOC__UD) added to FOC_UDCPS is used for inverse Park transformation. Range: [–32768,32767]

13.2.28 FOC_UQCPS (0x40C4, 0x40C5)

FOC_UQCPSH(0x40C4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_UQCPS[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_UQCPSL(0x40C5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_UQCPS[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_UQCPS	Compensation Value for q-axis Voltage The result of q-axis PI controller (FOC__UQ) added to FOC_UQCPS is used for inverse Park transformation. Range: [–32768,32767]

13.2.29 FOC__IC (0x40C6, 0x40C7)

FOC__ICH(0x40C6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__IC[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__ICL(0x40C7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__IC[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__IC	Sampled C-phase Current Range: [–32768,32767]

13.2.30 FOC__IB (0x40C8, 0x40C9)

FOC__IBH(0x40C8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__IB[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IBL(0x40C9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__IB[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__IB	Sampled B-phase Current Range: [–32768,32767]

13.2.31 FOC__IA (0x40CA, 0x40CB)

FOC__IAH(0x40CA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__IA[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IAL(0x40CB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__IA[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__IA	Sampled A-phase Current Range: [–32768,32767]

13.2.32 FOC__THETA (0x40CC, 0x40CD)

FOC__THETAH(0x40CC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__THETA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__THETAL(0x40CD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__THETA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__THETA	FOC output angle Range: [–32768,32767] –32768 ~ 32767 mapping –180°~ 180° Providing FOC__THETA = 8192, the angle is 8192/32768*180° = 45°

13.2.33 FOC__ETHETA (0x40CE, 0x40CF)

FOC__ETHETAH(0x40CE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__ETHETA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__ETHETAL(0x40CF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__ETHETA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__ETHETA	Read: The angle estimated by estimator (before compensation). The format is the same as FOC__THETA Write: The initial angle in estimator Range: [–32768,32767]

13.2.34 FOC__EALP (0x40D0, 0x40D1)

FOC__EALPH(0x40D0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__EALP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__EALPL(0x40D1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__EALP[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__EALP	α-axis Estimated BEMF Range: [–32768,32767]

13.2.35 FOC__EBET (0x40D2, 0x40D3)

FOC__EBETH(0x40D2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__EBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__EBETL(0x40D3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__EBET[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__EBET	β-axis Estimated BEMF Range: [–32768,32767]

13.2.36 FOC__EOME (0x40D4, 0x40D5)

FOC__EOMEH(0x40D4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__EOME[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__EOMEL(0x40D5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__EOME[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__EOME	Estimator Output Speed Range: [–32768,32767]

13.2.37 FOC__UQEX (0x40D6, 0x40D7)

FOC__UQEXH(0x40D6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__UQEX[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__UQEXL(0x40D7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__UQEX[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__UQEX	Overflow Value of q-axis PI Controller. FOC_UQEX = FOC__UQ – FOC_QMAX When FOC_UQ > FOC_QMAX, FOC_UQEX is positive When FOC_UQ < FOC_QMAX, FOC_UQEX is negative FOC__UQEX can be used to as the input of field weakening controller. Range: [–32768,32767]

13.2.38 FOC_KFG (0x40D6, 0x40D7)

FOC_KFGH(0x40D6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_KFG[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_KFGL(0x40D7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_KFG[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_KFG	Coefficient of FG Calculation FOC module performs the calculation based on FOC_EOMELPF and FOC_KFG in each PWM cycle. The result is updated to TIM4__ARR and half of the result (TIM4__ARR/2) to TIM4__DR by hardware. See FG Generation for more details. Range: [0,65535] Note: The clock division facto of Timer4 shall be configured if FOC_KFG overflows. When FOC_KFG = 0, this feature is disabled.

13.2.39 FOC__POW (0x40D8, 0x40D9)

FOC__POWH(0x40D8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__POW[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__POWL(0x40D9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__POW[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__POW	Motor Power; Range: [–32768,32767]

13.2.40 FOC_EOMEKLPF (0x40D8)

Bit	7	6	5	4	3	2	1	0
Name	FOC_EOMEKLPF							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_EOMEKLPF	LPF coefficient of estimated speed FOC_EOMELPF of the estimator LPF is calculated in every PWM cycle Range 1 ~ 255 mapping 1/32768 ~ 255/32768

13.2.41 FOC__IAMAX (0x40DA, 0x40DB)

FOC__IAMAXH(0x40DA)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IAMAX[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IAMAXL(0x40DB)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IAMAX[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__IAMAX	Maximum A-phase Current Recorded maximum value of A phase current. This value may be unreliable unless the motor rotates in a full electrical period. This maximum value will not be cleared automatically unless FOC_CR2[ICLR] is set to 1. Range: [–32768,32767]

13.2.42 FOC__IBMAX (0x40DC, 0x40DD)

FOC__IBMAXH(0x40DC)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IBMAX[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IBMAXL(0x40DD)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IBMAX[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__IBMAX	Maximum B-phase Current Recorded maximum value of B phase current. This value may be unreliable unless the motor rotates in a full electrical period. This value will not be cleared automatically unless FOC_CR2[ICLR] is set to 1. Range: [–32768,32767]

13.2.43 FOC__ICMAX (0x40DE, 0x40DF)

FOC__ICMAXH(0x40DE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__ICMAX [15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__ICMAXL(0x40DF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__ICMAX [7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__ICMAX	Maximum Value of C-phase Current Recorded maximum value of C phase current. This value may be unreliable unless the motor rotates in a full electrical period. This value will not be cleared automatically unless FOC_CR2[ICLR] is set to 1. Range: [–32768,32767]

13.2.44 FOC_EFREQMAX (0x406F)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EFREQMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	1	1	1	1

Bit	Name	Description
[7:0]	FOC_EFREQMAX	Maximum of omega When omega[15:8] is bigger than this value, the output speed OME is : FOC_CR0[OMAS] = 0: FOC_EFREQMAX*256 FOC_CR0[OMAS] = 1: FOC_EFREQHOLD Range: [0,127] 0 ~ 127 mapping the speed range 0 ~ 32767 Note: This bit is invalid when MSB is 1.

13.2.45 FOC_EKP (0x4074, 0x4075)

FOC_EKPH(0x4074)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EKPL(0x4075)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EKP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EKP	KP coefficient of PI controller for estimated angle of the estimator. MSB is always 0.Q12 format. Range: [0,32767]

13.2.46 FOC_EKI (0x4076, 0x4077)

FOC_EKIH(0x4076)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EKI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EKIL(0x4077)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EKI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EKI	KI coefficient of PI controller for estimated angle of the estimator. MSB is always 0. Q15 format Range: [0,32767]

13.2.47 FOC_EBMFK (0x407C, 0x407D)

FOC_EBMFKH(0x407C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EBMFK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EBMFKL(0x407D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EBMFK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EBMFK	LPF Coefficient (EKLPF) of BEMF Low-pass Filter. Q15 format Range: [0,32767] EKLPF = FOC_EBMFK*FOC__EOME FOC_EBMFK = 2*π*fbase*Ts

13.2.48 FOC_KSLIDE (0x4078, 0x4079)

FOC_KSLIDEH(0x4078)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_KSLIDE/FOC_PLLKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_KSLIDEL(0x4079)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_KSLIDE/FOC_PLLKP [7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_KSLIDE /FOC_PLLKP	FOC_CR2[ESEL] = 0: Gain coefficient of SMO; Q15 format FOC_CR2[ESEL] = 1: KP coefficient of PI controller on PLL; Q12 format Range: [0,32767]. MSB is always 0.

13.2.49 FOC_EKLPFMIN (0x407A, 0x407B)

FOC_EKLPFMINH(0x407A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EKLPFMIN/FOC_PLLKPI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EKLPFMINH(0x407B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EKLPFMIN/FOC_PLLKPI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EKLPFMIN /FOC_PLLKI	FOC_CR2[ESEL] = 0: The minimum value of BEMF low pass filter. EKLPF is forced to be this value if it is lower than this value. Q15 format FOC_CR2[ESEL] = 1: KI coefficient of PI controller on PLL; Q12 format; Range: [0,32767]; MSB is always 0.

13.2.50 FOC_OMEKLPF (0x407E, 0x407F)

FOC_OMEKLPFH(0x407E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_OMEKLPF[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_OMEKLPFL(0x407F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_OMEKLPF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_OMEKLPF	LPF Coefficient of FOC__EOME; MSB is always 0; Q15 format Range: [0,32767]

13.2.51 FOC_FBASE (0x4080, 0x4081)

FOC_FBASEH(0x4080)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_FBASE[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_FBASEL(0x4081)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_FBASE[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_FBASE	Frequency Base of Estimator Range: [0,32767] FOC_FBASE = fbase*Ts*32768 Providing fbase = 200Hz, Ts = 62.5μs, FOC_FBASE = 200*0.0000625*32768 = 409(0x199)

13.2.52 FOC_EFREQACC (0x4082, 0x4083)

FOC_EFREQACCH(0x4082)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EFREQACC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EFREQACCL(0x4083)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EFREQACC[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EFREQACC	Speed Increment of the Forced Angle Mode. FOC_EFREQACC is an internal 24-bit variable and MSB is sign bit. Low-order 16 bits are written by software. Range: [0,65535] Providing fbase = 200Hz, pole pairs (pp) = 4, speed_base = 60*fbase/pp = 3000rpm. To set speed increment to 3rpm, FOC_EFREQACC = 3rpm/speed_base*32768*256 = 8388 (0x20C4)

13.2.53 FOC_EFREQMIN (0x4084, 0x4085)

FOC_EFREQMINH(0x4084)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EFREQMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EFREQMINL(0x4085)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EFREQMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EFREQMIN	Switch Threshold of the Estimated Angle FOC_EFREQMIN is an internal 24–bit variable, and MSB is sign bit. High-order 16 bits are written by software. With Forced Angle of Estimator Mode enabled, FOC module outputs forced angle when the estimated angle is smaller than the bit value. Range: [–32768,32767] Providing fbase = 200Hz, pole pairs pp = 4, speed_base = 60*fbase/pp = 3000rpm. To set switch speed to 30rpm, FOC_EFREQMIN = 30rpm/speed_base*32768 = 327 (0x147)

13.2.54 FOC_EFREQHOLD (0x4086, 0x4087)

FOC_EFREQHOLDH(0x4086)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EFREQHOLD[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EFREQHOLDL(0x4087)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Name	FOC_EFREQHOLD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EFREQHOLD	Maximum Value of Forced Speed of the Estimator FOC_EFREQHOLD is an internal 24-bit variable, and MSB is sign bit. High 16 bits are written by the software. The forced speed increases until it reaches this value. Range: [–32768,32767] Providing fbase = 200Hz, pole pairs pp = 4, speed_base = 60*fbase/pp = 3000rpm. To set maximum forced speed to 60rpm, FOC_EFREQHOLD = 60rpm/speed_base*32768 = 655 (0x028F)

13.2.55 FOC_EK3 (0x4088, 0x4089)

FOC_EK3H(0x4088)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EK3[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK3L(0x4089)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EK3[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EK3	The 3rd coefficient of the current model in estimator; MSB is always 0; Q15 format; Range: [–32768,32767]

13.2.56 FOC_EK4 (0x408A, 0x408B)

FOC_EK4H(0x408A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EK4[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK4L(0x408B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EK4[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EK4	The 4th coefficient of the current model in estimator; Q15 format Range: [–32768,32767]

13.2.57 FOC_EK1 (0x408C, 0x408D)

FOC_EK1H(0x408C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EK1[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

FOC_EK1L(0x408D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EK1	The 1st coefficient of the current model in estimator; MSB is always 0; Q15 format Range: [0,32767]

13.2.58 FOC_EK2 (0x408E, 0x408F)

FOC_EK2H(0x408E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_EK2[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK2L(0x408F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_EK2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_EK2	The 2nd coefficient of the current model in estimator; MSB is always 0; Q15 format Range: [0,32767]

13.2.59 FOC_IDREF (0x4090, 0x4091)

FOC_IDREFH(0x4090)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_IDREF[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_IDREFL(0x4091)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_IDREF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_IDREF	User-defined d-axis Current Range: [–32768,32767]

13.2.60 FOC_IQREF (0x4092, 0x4093)

FOC_IQREFH(0x4092)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_IQREF[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_IQREFL(0x4093)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Name	FOC_IQREF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_IQREF	User-defined q-axis Current Range: [–32768,32767]

13.2.61 FOC_DQKP (0x4094, 0x4095)

FOC_DQKPH(0x4094)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_DQKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DQKPL(0x4095)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DQKP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_DQKP	KP Coefficient of dq-axis PI Controller; MSB is always 0; Q12 format Range: [0,32767], corresponding to range of Q12: [0,8]

13.2.62 FOC_DQKI (0x4096, 0x4097)

FOC_DQKIH(0x4096)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_DQKI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DQKIL(0x4097)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DQKI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_DQKI	KI Coefficient of dq-axis PI Controller; MSB is always 0; Q15 format Range: [0,32767], corresponding to range of Q15: [0,1]

13.2.63 FOC__UDCFLT (0x4098, 0x4099)

FOC__UDCFLTH(0x4098)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__UDCFLT[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__UDCFLTL(0x4099)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__UDCFLT[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__UDCFLT	Filtered Bus Voltage FOC module samples the bus voltage and filters it to obtain FOC__UDCFLT. ADC channel 2 (external voltage divider) or channel 14 (internal voltage divider) can be selected. Range: [0,32767] Providing the sampled bus voltage is divided by 6 and ADC voltage range is 0V ~ 5V, the sampling range of bus voltage is 0V ~ 30V. Providing FOC_UDCFLT is 19661(0x4CCD), the bus voltage is 19661/32768*5V*6 = 18V.

14 Timer1

14.1 Timer1 Operations

Timer1 consists of a 16-bit up-counting Basic Timer and a 16-bit up counting Reload Timer. Timer1 can be used in the applications of square-wave controlled BLDC motor drive. Timer1 features as follows.

- The 16-bit up counting Basic Timer is used to record the time between two position detected events or two phases commutations (60 degree time) and also can be used for forced commutation control when phase detection fails.
- The 16-bit up counting Reload Timer is used to control the time from position detected to phase commutation, as well as masking time for diode freewheeling, in which position detection is disabled
- The 3-bit programmable frequency prescaler divides the system clock. The divided clock is used as the clock source of the two counters
- Configurable filtering signals and sampling delay for position detection
- Position detection module generates the position signal required for phase commutation according to the input signal
- 7 groups state register control comparator and pre-driver output
- 6 interrupt sources

The internal structure of Timer1 is shown in Figure 14-1.

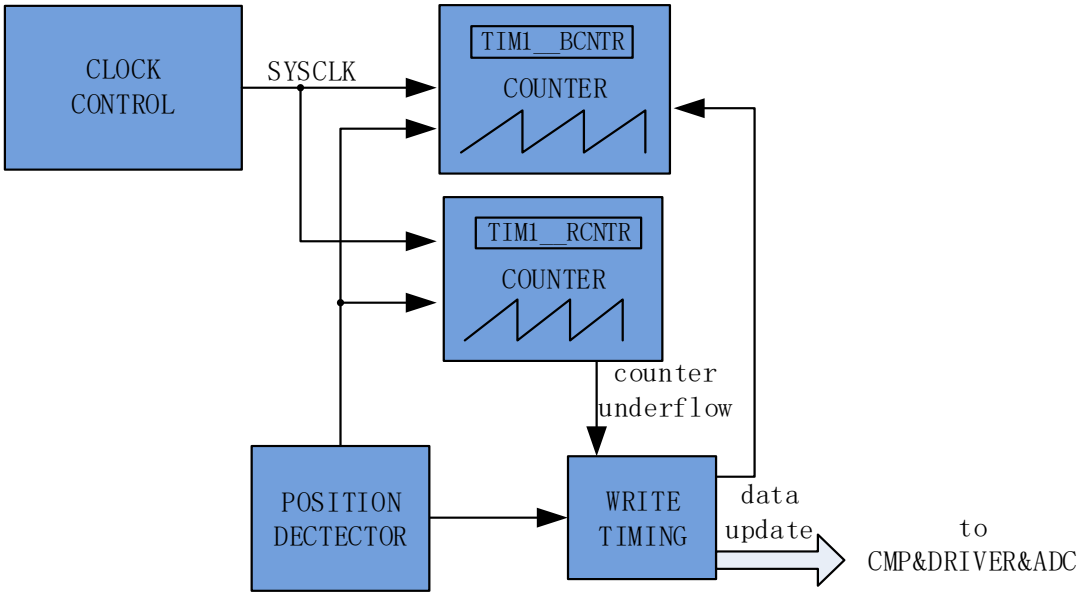


Figure 14-1 Internal Structure of Timer1

14.1.1 Timer1 Counter Module

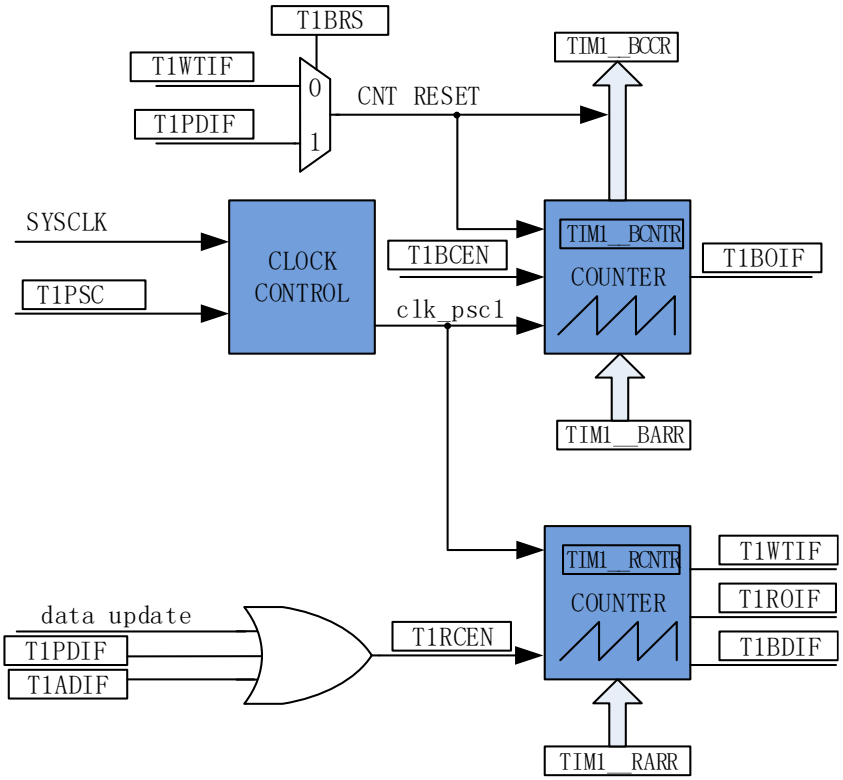


Figure 14-1 Timer Counting Units

Timer1 consists of a frequency prescaler, an 16-bit up-counting Base Timer and an 16-bit up-counting Reload Timer.

14.1.1.1 Prescaler

Prescaler is used to divide the system clock frequency and generate clock source for Basic Timer and Reload Timer. It offers 8 division coefficients and can be selected with TIM1_CR3[T1PSC]. Since this register has no buffer, the clock rate is immediately updated after the division coefficient is written. Therefore, the division coefficient shall be configured when both the Basic Timer and Reload Timer are not working. The clock rate clk_psc1 = SYSCLK/(2^TIM1_CR3[T1PSC]). The clock rate corresponding to TIM1_CR3[T1PSC] is shown in Table 14-1.

Table 14-1 Mapping between Clock Rate and TIM1_CR3[T1PSC]

TIM1_CR3[T1PSC]	Coefficient	clk_psc1(Hz)	TIM1_CR3[T1PSC]	Coefficient	clk_psc1(Hz)
000	1	24M	100	16	1.5M
001	2	12M	101	32	750k
010	4	6M	110	64	375k
011	8	3M	111	128	187.5k

14.1.1.2 Basic Timer

The Basic Timer is a 16-bit up counter with its count value held in the TIM1__BCNTR register. TIM1__BCNTR value is loaded into Capture Register TIM1__BCCR upon a Position Detection Interrupt TIM1_SR[T1PDIF] or a Write Timing Interrupt TIM1_SR[T1WTIF] (selected by TIM1_CR2[T1BRS] bit). Meanwhile, TIM1__BCNTR bit is cleared to 0 and restarts the counter cycle. TIM1__BCCR captures the time between two position detection interrupts or two write timing interrupts, i.e. 60° commutation time. These time inputs are averaged multiple times (programmed by the TIM1_CR0[T1CFLT] bit) before loading the average as a 60° commutation base into the TIM1_BCOR register. When TIM1__BARR auto load feature is enabled (TIM1_CR1[BAPE] is set to 1), TIM1__BARR loads the value of TIM1_BCOR by hardware. When count value of TIM1__BCNTR increases to TIM1__BARR, overflow interrupt flag TIM1_SR[T1BOIF] of the Basic Timer is set to 1. If forced commutation feature is enabled, phase commutation occurs and the Basic Timer Register is cleared to 0. Otherwise, the Basic Timer Register will not be cleared until it counts up to 0xFFFF and becomes overflowed.

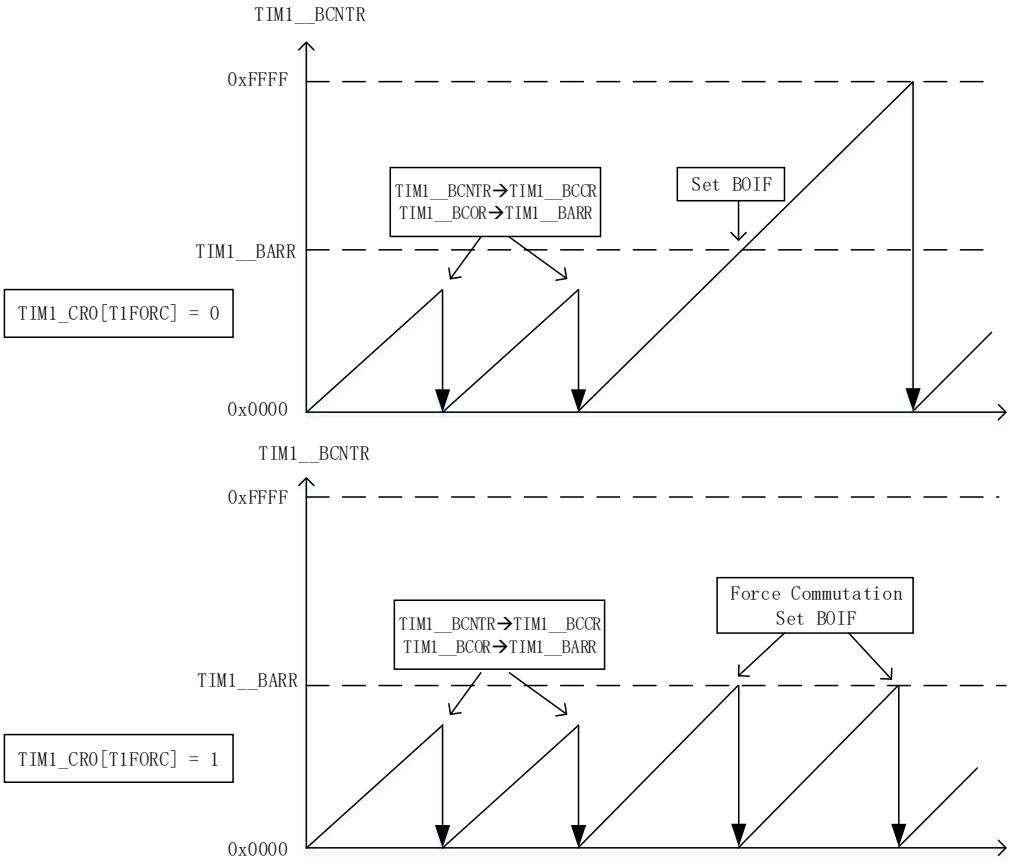


Figure 14-2 Waveform of Basic Timer

In Manual mode (TIM1_IER[T1MAME] = 1), TIM1__BCNTR is cleared by Basic Timer overflow event instead of TIM1_CR2[T1BRS].

14.1.1.3 Reload Timer

The reload timer is a 16-bit up counter with its count value held in TIM1__RCNTR register. The timer overflows when TIM1__RCNTR increases to TIM1__RARR. It stops counting when TIM1_SR[T1ROIF] (overflow interrupt flag of the reload counter) is set to 1, and TIM1__RCNTR and TIM1_CR0[T1RCEN] are cleared to 0. TIM1_CR0[T1RCEN] is set to 1 to restart reload timer when position detection interrupt or write timing interrupt is generated.

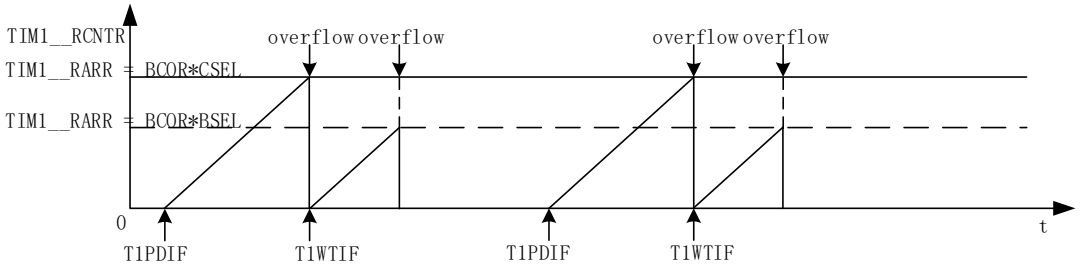


Figure 14-3 Waveform of Reload Timer

14.1.2 Position Detection

14.1.2.1 Position Detection Signal

TIM1_CR3[T1TIS] bit selects the source of position detection signal, including CMP0/1/2 (CMP based position detection), GPIO (Hall effected position detection) or ADC (ADC based position detection). The CMP_CR1[HALLSEL] bit is used to configure GPIO sourced by P1.4/P1.6/P2.1 (after function switching) or P0.2/P3.7/P3.6. The TIM1_CR3[T1INM] bit decides whether the CMP/GPIO signal is filtered. A Position Detection Interrupt is generated upon the completion of position detection. Position Detection Interrupts are divided into CMP/GPIO Position Detection Interrupt and ADC Position Detection Interrupt.

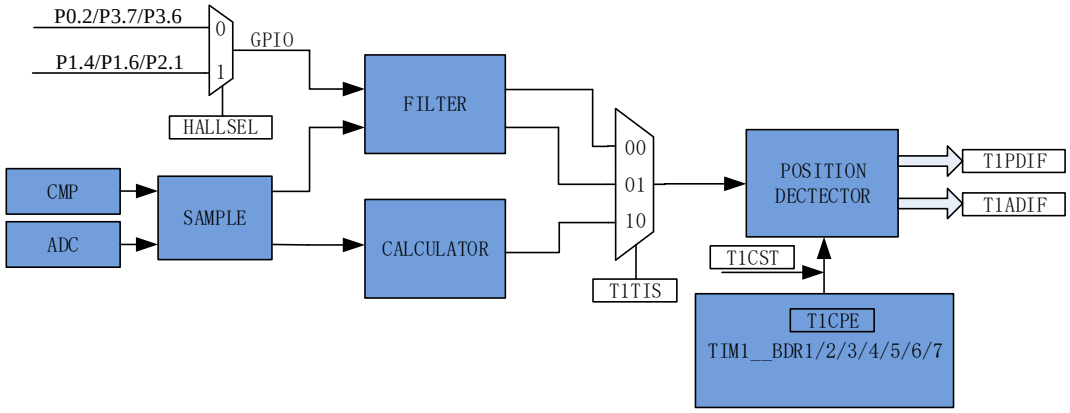


Figure 14-4 Functional Block Diagram of Position Detection

14.1.2.2 CMP/GPIO Position Detection Event

The register bank TIM1_DBR1/2/3/4/5/6/7[T1CPE] is configured to select the active edge of position detection signal. When an active edge of CMP/GPIO position detection signal is detected, it indicates the position detection is successfully done, allowing CMP/GPIO Position Detection Interrupt Flag TIM1_SR[T1PDIF] bit to become logic 1. TIM1_CR4[T1CST] bit selects TIM1_DBR1/2/3/4/5/6/7[T1CPE] timing.

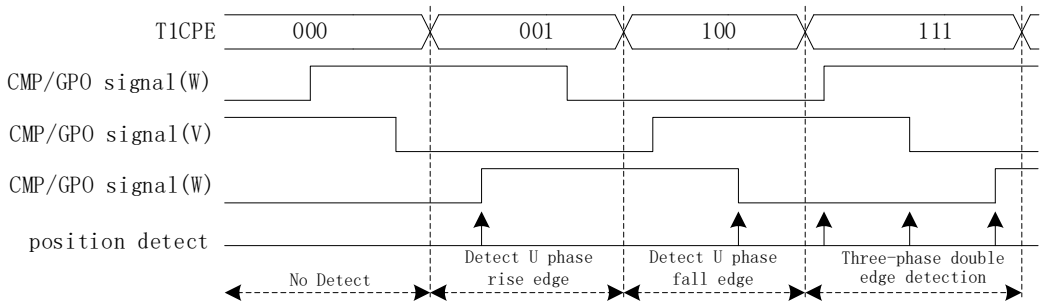


Figure 14-5 Timing Diagram of CMP/GPIO Position Detection

The relation between active edge and TIM1_DBR1/2/3/4/5/6/7[T1CPE] is shown in Table 14-2.

Table 14-2 Mapping between Active Edge and TIM1_DBR1/2/3/4/5/6/7[T1CPE]

T1CPE	Description	T1CPE	Description
000	0	100	U-phase corresponding comparator is enabled when falling edge of U-phase is detected.
001	U-phase corresponding comparator is enabled when rising edge of U-phase is detected.	101	W-phase corresponding comparator is enabled when rising edge of W-phase is detected.
010	W-phase corresponding comparator is enabled when falling edge of W-phase is detected.	110	V-phase corresponding comparator is enabled when falling edge of V-phase is detected.
011	V-phase corresponding comparator is enabled when rising edge of V-phase is detected.	111	U+W+V-phase corresponding comparator is enabled when rising falling edge of U+W+V-phase is detected.

14.1.2.3 ADC Position Detection Event

TIM1_CR3[T1TIS] is configured to select the position detection signal from ADC. Timer1 controls ADC to sample the voltage of active phase and floating phase. To get the position information, which are calculated as the following equation:

$$TIM1\_URES = K \times TIM1\_UCOP - TIM1\_UFLP$$

Where,

K: ADC position detection factor

TIM1__UCOP: ADC sampling value of active phase

TIM1__UFLP: ADC sampling value of floating phase

K, TIM1__UCOP and TIM1__UFLP are determined by TIM1_DBR1/2/3/4/5/6/7[T1CPE] bit, as detailed in Table 14-3.

Table 14-3 Relation between TIM1_DBR1/2/3/4/5/6/7[T1CPE] and K, TIM1__UCOP, TIM1__UFLP

T1CPE	Description
000	Reserved
001	TIM1_KR for K, W-phase voltage for TIM1__UCOP, and U-phase voltage for TIM1__UFLP
010	TIM1_KF for K, U-phase voltage for TIM1__UCOP, and W-phase voltage for TIM1__UFLP
011	TIM1__KR for K, U-phase voltage for TIM1__UCOP, and V-phase voltage for TIM1__UFLP
100	TIM1_KR for K, V-phase voltage for TIM1__UCOP, and U-phase voltage for TIM1__UFLP
101	TIM1_KR for K, V-phase voltage for TIM1__UCOP, and W-phase voltage for TIM1__UFLP
110	TIM1_KF for K, W-phase voltage for TIM1__UCOP, and V-phase voltage for TIM1__UFLP
111	Reserved

When TIM1__URES has a negative step or a positive step, ADC position detected event is generated and TIM1_SR[T1ADIF] (ADC position detection interrupt flag) is set to 1. The position at which ADC Position Detection Interrupt is generated is controlled by setting the coefficient K. In this case, the phase commutation degree can be controlled flexibly.

14.1.2.4 Sampling

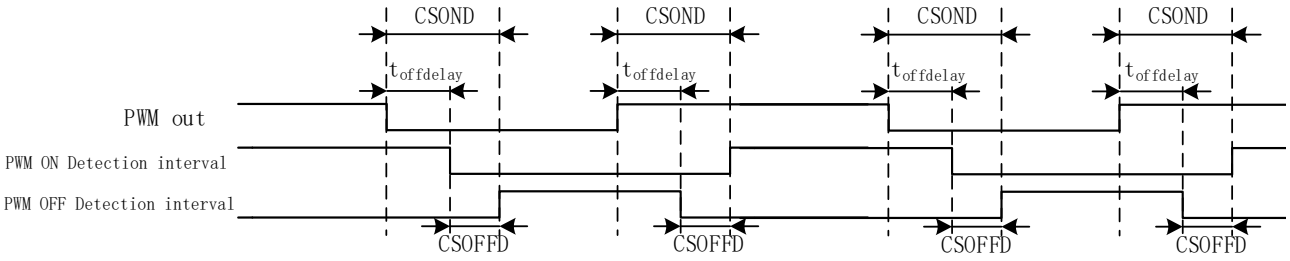


Figure 14-6 Timing Diagram of Sampling

Affected by switching rate of the power device, BEMF signal lags behind PWM output. CMP_SAMR[CSOFFD], CMP_SAMR[CSOND] and CMP_CR4[FAEN] shall be set reasonably to adjust the sampling interval and obtain the valid position detection signal. When TIM1_CR3[T1TIS] = 01 or 10, Timer1 enables CMP0/1/2 to output the compare results between phase BEMF and neutral point, or starts ADC module to sample floating voltage.

See section 28.1.4 for more details.

14.1.2.5 Filtering

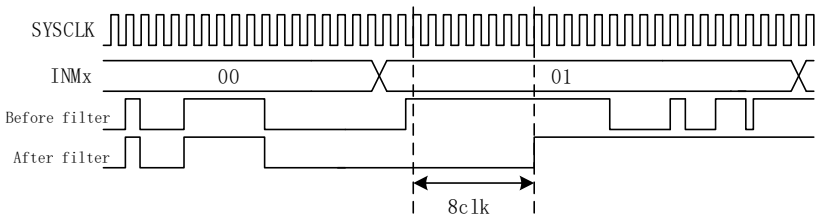


Figure 14-7 Timing Diagram of Filtering Module

According to TIM1_CR3[T1INM] and CMP_CR4[FAEN], the filtered pulse width of input noise can be selected as 8/16/24/32/64/96 system clock. After this feature is enabled, the signal is lagged behind about 8/16/24/32/64/96 system clock.

14.1.3 Writing Sequence Event

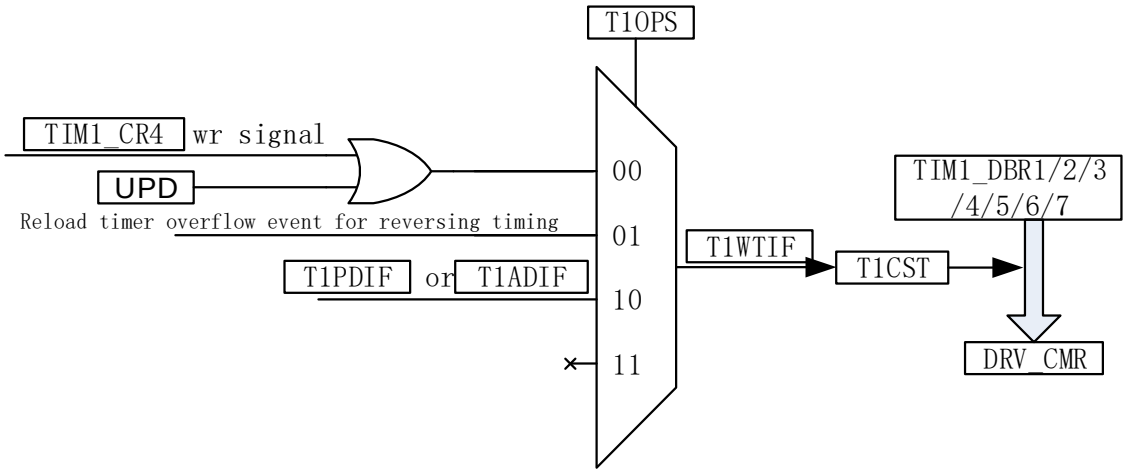


Figure 14-8 Block Diagram of Writing Sequence

When the control logic, predefined in TIM1_DBR1/2/3/4/5/6/7, is sent to driver register DRV_CMR, a writing sequence interrupt is generated. The triggered source is selected by the configuration of TIM1_CR0[T1OPS], and the software, Reload Timer overflow event or position detected event can be selected. When a writing sequence interrupt is generated, writing sequence interrupt flag is set to 1. If TIM1_CR4[T1CST] is in 001 ~ 110, TIM1_CR4[T1CST] adds 1 automatically.

14.1.4 Timer1 Interrupt

Timer1 has 6 interrupt sources:

- Base Timer overflow interrupt
- Reload Timer overflow interrupt
- Writing sequence interrupt
- Diode freewheeling end interrupt

- CMP/GPIO position detection interrupt
- ADC position detected interrupt

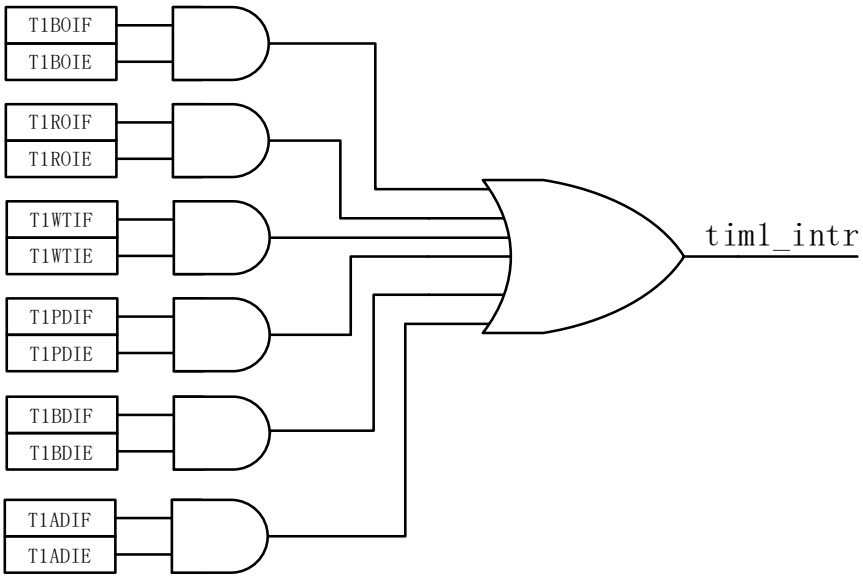


Figure 14-9 Timer1 Interrupt Sources

14.2 Square Wave Control Based BLDC Motor

For BLDC motor square wave control application, Timer1 works with CMP0/1/2 and Driver module to achieve the following features:

- Automatic record of 60 degree time, filtered as 60 degree reference time
- Automatic forced phase commutation when position signal is not detected
- Automatic diode freewheeling masking, i.e., stopping position detection during diode freewheeling
- Automatic control of the time from position detected to phase commutation to achieve automatic commutation
- Take over CMP_CR2[CMP0SEL] to control CMP0/1/2 automatically
- Comparator signal can be set to avoid power device switching oscillation for sampling, and the signal can be configured to be filtered after sampling
- Take over DRV_CMR register to control 6 PWM outputs automatically

14.2.1 Six-step Phase Commutation of Square Wave Control

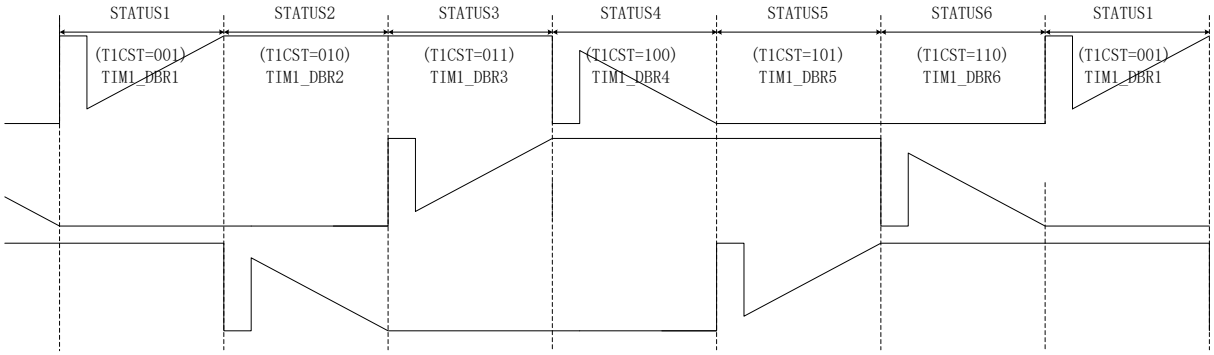


Figure 14-10 Diagram of Six-step Phase Commutation of Square Wave Control

TIM1_CR4[T1CST] is the commutation state machine. Among them, state 0 is used to output off state, and state 7 is customizable for braking, pre-charging, pre-positioning, startup, etc. States 1 ~ 6 are used for six-step automatic commutation, and the state machine TIM1_CR4[T1CST] automatically adds 1 after phase commutation.

The states 1~7 maps to the TIM1_DBR1~7. When writing sequence interrupt occurs, TIM1_DBRx corresponding to the current state is automatically transferred to DRV_CMR and CMP_CR2[CMP0SEL] for phase commutation and position detection.

14.2.2 Square Wave Control Working Principle

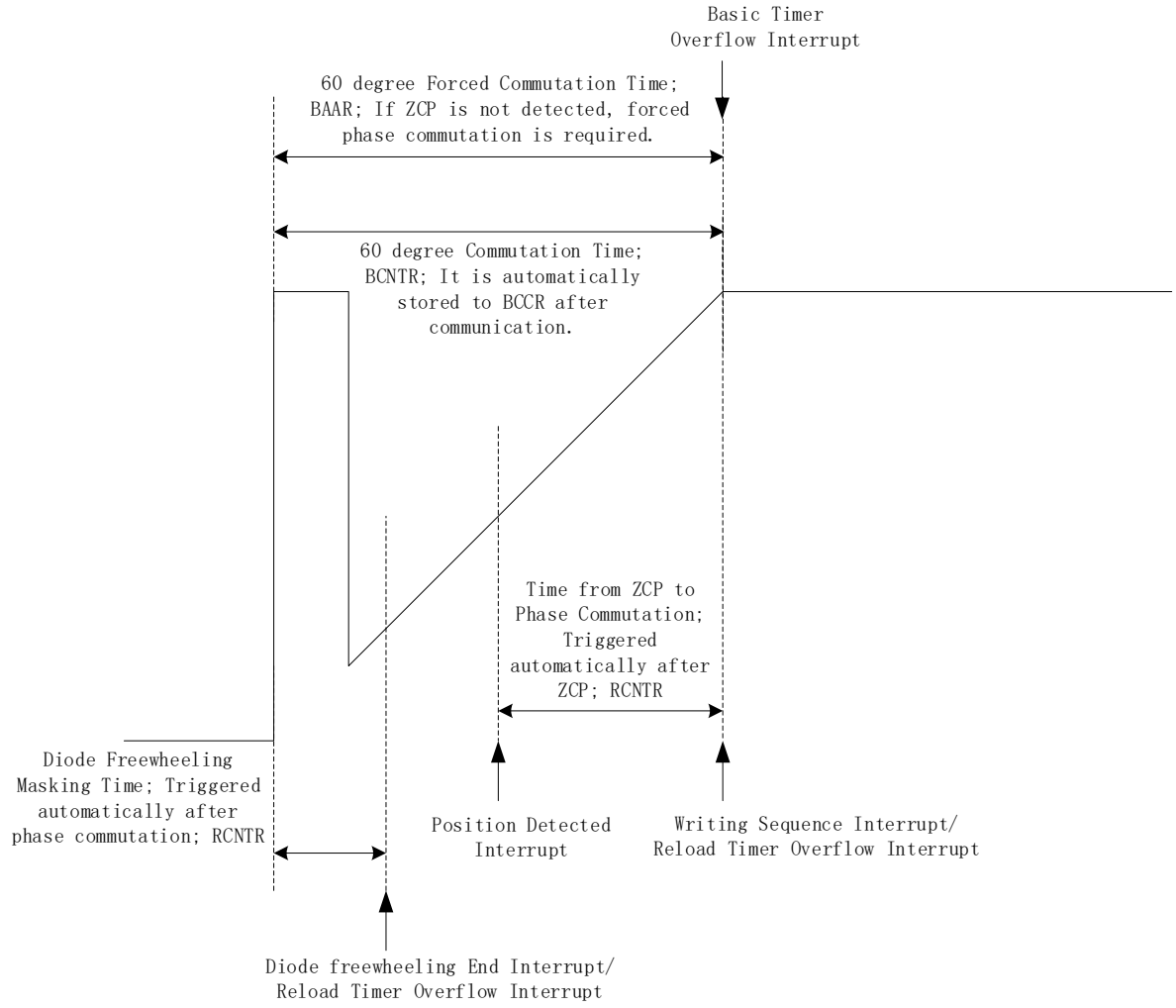


Figure 14-11 Square Wave Control Working Principle

14.2.2.1 60° Commutation Base Time

TIM1__BCCR captures the time of last 60 degree. TIM1_CR2[T1BRS] is set to 0 to capture the time between two writing sequence interrupts and TIM1_CR2[T1BRS] to1 to capture the time between two position detected interrupts.

TIM1_BCOR is the filtered 60 degree time, i.e., 60 degree base time. TIM1_CR0[CFLT] can select the last 1/2/4/8 TIM1__BCCR averaged to obtain TIM1_BCOR.

In square wave control mode, the diode freewheeling masking time, the time from position detected to commutation, and the time to forced commutation are determined by the 60 degree base time TIM1_BCOR. When base timer is auto-load enabled (TIM1_CR1[T1BAPE] = 1) and is reset due to a position detection interrupt or a write timing interrupt, TIM1_BCOR is transferred to TIM1__BARR to control the forced phase commutation.

14.2.2.2 Forced Commutation at 60°

When the motor rotates smoothly, ZCP is generally detected after 30 degrees of rotation after a phase commutation and a position detection interrupt is generated. If ZCP is not detected in 60 degree after the phase commutation, position detection fails and a forced commutation is required. In this case, TIM1_CR0[FORC] is set to 1 to enable the forced commutation feature. During previous commutation, the counter TIM1_BCNTR is cleared to 0 by timing interrupt and restarts counting, while TIM1_BCCR captures the count value held in TIM1_BCNTR, which is filtered and stored in TIM1_BCOR as the 60 degree base time. When auto-load feature is enabled (TIM1_CR1[T1BAPE] = 1), the value held in TIM1_BCOR is loaded into TIM1_BARR after the Basic Timer is cleared. If no ZCP is detected in 60 degree after commutation (TIM1_BCNTR matches TIM1_BARR), TIM1_SR[T1BOIF] (overflow interrupt flag of the Basic Timer) is set to 1 for forced phase commutation, and the counter TIM1_BCNTR is cleared to 0. But if an ZCP is detected within 60 degrees after phase commutation, even when TIM1__BCNTR > TIM1__BARR, the forced commutation will not be triggered and TIM1_SR[T1BOIF] will not be set to 1. When forced commutation feature is disabled (TIM1_CR0[FORC] = 0) and TIM1__BCNTR > TIM1__BARR, the interrupt flag TIM1_SR[T1BOIF] is set to 1 and no forced phase commutation is automatically performed. Phase commutation can be performed manually by Basic Timer overflow interrupt flag and the position detected interrupt flag.

14.2.2.3 Diode Freewheeling Masking

After the commutation, inductance energy of the phase is released to the power supply or ground through the diode since the original active phase becomes a floating phase. During diode freewheeling, the floating phase BEMF signal cannot be measured. By masking comparator signal or ADC sampling value during diode freewheeling, wrong commutation caused by wrong signal generated by the freewheeling is avoided. After freewheeling masking, the freewheeling masking end interrupt flag TIM1_SR[T1BDIF] is generated.

Freewheeling masking time is set by TIM1_CR1[BSEL] with the formula: Masking angle = TIM1_CR1[BSEL]/128*60°.

14.2.2.4 Angle of Position Detected to Commutation

After commutation, a ZCP is detected (generating a position detected interrupt) and the hardware starts counting according to the software-set time between ZCP and the commutation. After the counting ends, the hardware automatically implements phase commutation and generates the write sequence interrupt flag TIM1_SR[T1WTIF].

The time between ZCP and commutation is set by TIM1_CR2[CSEL] with the formula: Commutation angle = TIM1_CR2[CSEL]/128*60°.

14.2.2.5 Cycle-by-Cycle Current Limiting

See section 28.1.1.2 for cycle-by-cycle current limiting.

14.3 Timer1 Registers

14.3.1 TIM1_CR0 (0x4068)

Bit	7	6	5	4	3	2	1	0
Name	T1RWEN	T1CFLT		T1FORC	T1OPS		T1BCEN	T1RCEN
Type	W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	T1RWEN	TIM1_CR0[TIRCEN] Write Enable 0: No effect 1: When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] and TIM1_CR0[T1RCEN] shall be configured simultaneously to enable or disable TIM1_CR0[T1RCEN]. A write of "0x81" to TIM1_CR0 enables TIM1_CR0[T1RCEN] and "0x80" to disable TIM1_CR0[T1RCEN].
[6:5]	T1CFLT	60 Degree Base Time Filtering Selection The average of previous x times 60 degree is used as the base time 00: 1 times 60 degree 01: 2 times 60 degree 10: 4 times 60 degree 11: 8 times 60 degree
[4]	T1FORC	Forced Phase Commutation at 60° Enable 0: Disable 1: Enable Note: If a ZCP is detected, forced phase commutation will not be implemented even if this bit is enabled.
[3:2]	T1OPS	Phase Commutation Trigger Signal Selection Select the trigger signal for TIM1_DBRx to transfer data to DRV_CMR. 00: The transfer is triggered upon a write of "1" to TIM1_IER[T1UPD] in software or on a write to TIM1_CR4[T1CST]. 01: The transfer is triggered upon an overflow interrupt of reload timer commutation counter 10: The transfer is triggered upon a Position Detected Interrupt 11: Reserved
[1]	T1BCEN	Base Timer Enable 0: Disable 1: Enable
[0]	T1RCEN	Reload Timer Enable When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] and TIM1_CR0[T1RCEN] must be configured simultaneously to enable or disable TIM1_CR0[T1RCEN]. 0x81 is written to TIM1_CR0 to enable TIM1_CR0[T1RCEN] and 0x80 to disable TIM1_CR0[T1RCEN]. TIM1_CR0[T1RCEN] can be enabled by position detected interrupt and writing sequence interrupt. TIM1_CR0[T1RCEN] is cleared to 0 by hardware when the Reload Timer overflow interrupt is generated. In manual mode, TIM1_CR0[T1RCEN] cannot be automatically enabled or disabled by hardware. 0: Disable 1: Enable

14.3.2 TIM1_CR1 (0x4069)

Bit	7	6	5	4	3	2	1	0
Name	T1BAPE	BSEL						
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	T1BAPE	TIM1_BARR Auto-load Enable With this bit enabled, TIM1__BCOR is written to TIM1_BARR when Basic Timer is cleared due to a position detected event or a writing sequence event. It is used for forced phase commutation at 60°when no ZCP is detected. Setting the device in Manual mode has no effect on TIM1__BARR Register auto-load feature. 0: Disable 1: Enable
[6:0]	BSEL	Diode Freewheeling Masking Angle Selection Angle of diode freewheeling masking after phase commutation. During diode freewheeling masking, the position is not detected Diode freewheeling angle = TIM1_CR1[BSEL]/128*60° Note: In manual mode, this bit is invalid

14.3.3 TIM1_CR2 (0x406A)

Bit	7	6	5	4	3	2	1	0
Name	T1BRS	CSEL						
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	T1BRS	Base Timer Reset Source Selection In manual mode, this bit is invalid, and TIM1__BCNTR can only be cleared by a BCNTR overflow interrupt. 0: Writing sequence interrupt 1: Position detected interrupt
[6:0]	CSEL	Phase Commutation Angle Selection After a position detected event, phase commutation is implemented after the degree configured by TIM1_CR2[CSEL]. Commutation angle = TIM1_CR2[CSEL]/128*60°

14.3.4 TIM1_CR3 (0x406B)

Bit	7		6	5	4	3	2	1	0
Name	RSV		T1PSC			T1TIS		T1INM	
Type	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	—	0	0	0	0	1	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6:4]	T1PSC	Timer Clock Source Frequency Selection This bit is used to divide the system clock as the clock source for Basic Timer and Reload Timer. The clock source frequency of the two timers: 000: 24MHz 001: 12MHz 010: 6MHz 011: 3MHz 100: 1.5MHz 101: 750kHz 110: 375kHz 111: 187.5kHz
[3:2]	T1TIS	Position Detection Signal Selection 00: GPIO (select P1.4, P1.6, P2.1 or P0.2, P3.7, P3.6 by CMP_CR1[HALLSEL])

		01: Output signal of CMP0/1/2 10: Output signal of ADC 11: Reserved
[1:0]	T1INM	Filtering Pulse Width of the Position Detection Signal Selection. When pulse width of the input signal is less than the set value, it is filtered as noise. The filtering time is multiplied by 4 times according to CMP_CR4[FAEN]. When CMP_CR4[FAEN] = 0: 00: No filtering 01: 8 system clock cycles 10: 16 system clock cycles 11: 24 system clock cycles When CMP_CR4[FAEN]= 1: 00: No filtering 01: 32 system clock cycles 10: 64 system clock cycles 11: 96 system clock cycles

14.3.5 TIM1_CR4 (0x406C)

Bit	7	6	5	4	3	2	1	0
Name	RSV					T1CST		
Type	—	—	—	—	—	R/W	R/W	R/W
Reset	—	—	—	—	—	0	0	0

Bit	Name	Description
[7:3]	RSV	Reserved
[2:0]	T1CST	Phase Commutation State Machine The state machine corresponds to different TIM1_DBRx at different states When TIM1_CR4[T1CST] reads 001 ~ 111, Timer1 automatically enables or disables CMP0/1/2 according to TIM1_DBRx[T1CPE]. When TIM1_CR4[T1CST] reads 001 ~ 110, Timer1 automatically adds 1 each cycle upon a Write Timing Interrupt.

Table 14-4 Mapping between TIM1_CR4[T1CST] and TIM1_DBRx

TIM1_CR4[T1CST]	TIM1_DBRx	TIM1_CR4[T1CST]	TIM1_DBRx
000	0	100	TIM1_DBR4
001	TIM1_DBR1	101	TIM1_DBR5
010	TIM1_DBR2	110	TIM1_DBR6
011	TIM1_DBR3	111	TIM1_DBR7

14.3.6 TIM1_IER (0x406D)

Bit	7	6	5	4	3	2	1	0
Name	T1UPD	T1MAME	T1ADIE	T1BOIE	T1ROIE	T1WTIE	T1PDIE	T1BDIE
Type	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	T1UPD	When TIM1_CR0[T1OPS] = 00, a write of "1" to this bit enables data transfer. This bit is cleared to "0" by hardware after 1 is written.
[6]	T1MAME	Manual Mode Enable With this bit enabled, Basic Timer and Reload Timer acts as separate counters. The details are: TIM1__BCNTR of the Basic Timer is cleared by counter overflow interrupt instead of TIM1_CR2[T1BRS] TIM1_CR0[T1RCEN] of the Reload Timer cannot be cleared to 0 or set to 1 automatically, and is operated by software only. TIM1__RCNTR of the Reload Timer can be cleared to 0 upon a Reload Timer overflow Interrupt only.

		TIM1__RARR of the Reload Timer cannot be updated automatically, and is operated by software only. 0: Disable 1: Enable
[5]	T1ADIE	ADC Position Detected Interrupt Enable 0: Disable 1: Enable
[4]	T1BOIE	Base Timer Overflow Interrupt Enable 0: Disable 1: Enable
[3]	T1ROIE	Reload Timer Overflow Interrupt Enable 0: Disable 1: Enable
[2]	T1WTIE	Writing Sequence Interrupt Enable 0: Disable 1: Enable
[1]	T1PDIE	CMP/GPIO Position Detected Interrupt Enable 0: Disable 1: Enable
[0]	T1BDIE	Diode Freewheeling End Interrupt Enable 0: Disable 1: Enable

14.3.7 TIM1_SR (0x406E)

Bit	7	6	5	4	3	2	1	0
Name	RSV		T1ADIF	T1BOIF	T1ROIF	T1WTIF	T1PDIF	T1BDIF
Type	—	—	R/W0	R/W0	R/W0	R/W	R/W0	R/W0
Reset	—	—	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	T1ADIF	ADC Position Detected Interrupt Event Flag A position detected interrupt is generated when ADC position detection signal is matched with TIM1_DBRx[T1CPE]. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	T1BOIF	Base Timer Overflow Interrupt Flag An overflow event occurs when Basic Timer counts up and TIM1_BCNTR matches with TIM1_BARR. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[3]	T1ROIF	Reload Timer Overflow Interrupt Flag An overflow event occurs and TIM1__RCNTR is cleared to 0 when TIM1__RCNTR matches TIM1__RARR. Read: 0: No interrupt pending 1: Interrupt pending

		Write: 0: This bit is cleared to “0” 1: No effect
[2]	T1WTIF	Writing Sequence Interrupt Flag Writing sequence interrupt is generated when TIM1_DBRx is transferred to DRV_CMR. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: Write sequence interrupt is generated when IM1_CR0[T1OPS] = 00. Otherwise, it has No effect
[1]	T1PDIF	CMP/GPIO Position Detected Interrupt Flag A position detected interrupt is generated when CMP/GPIO position detection signal is matched with TIM1_DBRx[T1CPE]. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	T1BDIF	Diode Freewheeling End Interrupt Flag Diode freewheeling starts after phase commutation and an interrupt is generated at end. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect

14.3.8 TIM1__BCOR (0x4070, 0x4071)

TIM1__BCORH(0x4070)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__BCOR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__BCORL(0x4071)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__BCOR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIM1__BCOR	The bit is configured to capture filtered count values held in the Base Timer. TIM1_BCCR holds the filtering value (i.e., 60 degree base time).

14.3.9 TIM1_DBR1 (0x4074, 0x4075)

TIM1_DBR1H(0x4074)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
TIM1_DBR1L(0x4075)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Signal Input Edge Polarity and Comparator Enable Selection These bits are used to define the edge of the position detection input signal and enable or disable the corresponding comparator. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See CMP/GPIO Position Detection Event and Table 14-2.
[11]	T1WHP	High-side Output Polarity of W-phase 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of W-phase 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of V-phase 0: Active High 1: Active Low
[8]	T1VLP	Low-side output polarity of V-phase 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of U-phase 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of U-phase 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of W-phase 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable

Note: When TIM1_DBR1[T1WLE] and TIM1_DBR1[T1WHE], TIM1_DBR1[T1VLE] and TIM1_DBR1[T1VHE] or TIM1_DBR1[T1ULE] and TIM1_DBR1[T1UHE] are set to "1" at the same time, the three-phase high side and low side of bridges are complementary output and the deadtime is automatically inserted (same for registers TIM1_DBR2 ~ TIM1_DBR7).

14.3.10 TIM1_DBR2 (0x4076, 0x4077)

TIM1_DBR2H(0x4076)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
TIM1_DBR2L(0x4077)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Signal Input Edge Polarity and Comparator Enable Selection These bits are used to define the edge of the position detection input signal and enable or disable the corresponding comparator. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See CMP/GPIO Position Detection Event and Table 14-2.
[11]	T1WHP	High-side Output Polarity of W-phase 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of W-phase 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of V-phase 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of V-phase 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of U-phase 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of U-phase 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of W-phase 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable

[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable

14.3.11 TIM1_DBR3 (0x4078, 0x4079)

TIM1_DBR3H(0x4078)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
TIM1_DBR3L(0x4079)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Signal Input Edge Polarity and Comparator Enable Selection These bits are used to define active edge of the position detection input signal and enable or disable the corresponding comparator. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See CMP/GPIO Position Detection Event and Table 14-2.
[11]	T1WHP	High-side Output Polarity of W-phase 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of W-phase 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of V-phase 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of V-phase 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of U-phase 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of U-phase 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of W-phase 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable

[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable

14.3.12 TIM1_DBR4 (0x4080, 0x4081)

TIM1_DBR4H(0x4080)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
TIM1_DBR4L(0x4081)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Signal Input Edge Polarity and Comparator Enable Selection These bits are used to define active edge of the position detection input signal and enable or disable the corresponding comparator. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See CMP/GPIO Position Detection Event and Table 14-2.
[11]	T1WHP	High-side Output Polarity of W-phase 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of W-phase 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of V-phase 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of V-phase 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of U-phase 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of U-phase 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of W-phase 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable

[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable

14.3.13 TIM1_DBR5 (0x4082, 0x4083)

TIM1_DBR5H(0x4082)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
TIM1_DBR5L(0x4083)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Signal Input Edge Polarity and Comparator Enable Selection These bits are used to define active edge of the position detection input signal and enable or disable the corresponding comparator. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See CMP/GPIO Position Detection Event and Table 14-2.
[11]	T1WHP	High-side Output Polarity of W-phase 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of W-phase 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of V-phase 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of V-phase 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of U-phase 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of U-phase 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of W-phase 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable

[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable

14.3.14 TIM1_DBR6 (0x4084, 0x4085)

TIM1_DBR6H(0x4084)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
TIM1_DBR6L(0x4085)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Signal Input Edge Polarity and Comparator Enable Selection These bits are used to define active edge of the position detection input signal and enable or disable the corresponding comparator. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See CMP/GPIO Position Detection Event and Table 14-2.
[11]	T1WHP	High-side Output Polarity of W-phase 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of W-phase 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of V-phase 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of V-phase 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of U-phase 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of U-phase 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of W-phase 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable

[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable

14.3.15 TIM1_DBR7 (0x4086, 0x4087)

TIM1_DBR7H(0x4086)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
TIM1_DBR7L(0x4087)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Signal Input Edge Polarity and Comparator Enable Selection These bits are used to define active edge of the position detection input signal and enable or disable the corresponding comparator. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See CMP/GPIO Position Detection Event and Table 14-2.
[11]	T1WHP	High-side Output Polarity of W-phase 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of W-phase 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of V-phase 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of V-phase 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of U-phase 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of U-phase 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of W-phase 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable

[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable

14.3.16 TIM1__BCNTR (0x4082, 0x4083)

TIM1__BCNTRH(0x4082)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__BCNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0
TIM1__BCNTRL(0x4083)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__BCNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__BCNTR	Count value of Basic Timer, which is used to count 60 degree commutation time Auto mode: TIM1__BCNTR selects the reset source according to TIM1_CR2[T1BRS]. TIM1__BCNTR cannot restart when TIM1__BCNTR overflow interrupt is generated. Manual mode: TIM1__BCNTR restarts when TIM1__BCNTR overflow interrupt is generated.

14.3.17 TIM1__BCCR (0x4084, 0x4085)

TIM1__BCCRH(0x4084)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__BCCR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__BCCRL(0x4085)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__BCCR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__BCCR	Capture Count Values Held in Base Timer Auto mode: When Basic Timer is reset due to position detected interrupt or sequence writing interrupt, the count value before the reset is stored to TIM1__BCCR. Manual mode: When Basic Timer is reset due to overflow interrupt, the count value before the reset is stored to TIM1__BCCR.

14.3.18 TIM1__BARR (0x4086, 0x4087)

TIM1__BARRH(0x4086)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__BARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__BARRL(0x4087)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__BARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__BARR	Reload Value of Basic Timer When count value of the base timer is equal to TIM1__BARR, an overflow interrupt is generated and the counter is cleared to 0.

14.3.19 TIM1__RARR (0x4088, 0x4089)

TIM1__RARRH(0x4088)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__RARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__RARRL(0x4089)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__RARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__RARR	Auto-reload Value of Reload Timer When count of the Reload Timer is equal to TIM1__RARR, an overflow interrupt event is generated and the value of counter is cleared to 0. Auto mode: The value corresponding to diode freewheeling angle (set by TIM1_CR1[BSEL]) is updated to TIM1__RARR when a writing sequence interrupt is generated. The value corresponding to commutation angle (set by TIM1_CR2[CSEL]) is updated to TIM1__RARR when a position detected interrupt is generated. Manual mode: TIM1__RARR is written by software

14.3.20 TIM1__RCNTR (0x408A, 0x408B)

TIM1__RCNTRH(0x408A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__RCNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
TIM1__RCNTRL(0x408B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__RCNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	Name	Description
[15:0]	TIM1__RCNTR	Count value of the Reload Timer for counting time of diode freewheeling masking and ZCP to phase commutation Note: In manual mode, TIM1__RCNTR can be cleared to 0 only by a Reload Timer overflow interrupt.

14.3.21 TIM1__UCOP (0x408C, 0x408D)

TIM1__UCOPH(0x408C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__UCOP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__UCOPL(0x408D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__UCOP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__UCOP	ADC Sample Value of Conducting Phase Voltage (second-highest bit alignment)

14.3.22 TIM1__UFLP (0x408E, 0x408F)

TIM1__UFLPH(0x408E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__UCOP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__UFLPL(0x408F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__UCOP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__UFLP	ADC Sample Value of Floating Phase Voltage (second-highest bit alignment)

14.3.23 TIM1__URES (0x4090, 0x4091)

TIM1__URESH(0x4090)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__URES[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__URESL(0x4091)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__URES[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__URES	Calculation Result of ADC Position Detection Formula; Q15 format

14.3.24 TIM1__UIGN (0x4092, 0x4093)

TIM1__UIGNH(0x4092)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__UIGN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__UFLPL(0x4093)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__UIGN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__UIGN	No calculation is performed when sampled ADC voltage on conducting phase is less than this value.

14.3.25 TIM1__KF (0x4094, 0x4095)

TIM1__KFH(0x4094)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__KF[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__KFL(0x4095)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__KF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__KF	ADC position detection coefficient when floating phase voltage drops Range [0,32767]

14.3.26 TIM1__KR (0x4096, 0x4097)

TIM1__KRH(0x4096)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__KR[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__KRL(0x4097)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__KR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__KR	ADC position detection coefficient when floating phase voltage rises Range [0,32767]

14.3.27 TIM1__ITRIP (0x4098, 0x4099)

TIM1__ITRIPH(0x4098)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__ITRIP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
TIM1__ITRIPL(0x4099)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__ITRIP[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__ITRIP	Filtered Bus Current When DRV__CNTR = 0, the hardware module automatically samples the bus current and filters it for software using. The default channel is ADC channel 4 Range [0,32767] Note: This value is obtained by averaging the instantaneous current values of 8 samples

15 Timer2

15.1 Timer2 Instructions

Timer2 has four working modes:

- Output mode: PWM generation
- Input capture mode: Detect the duration of high and low level of input PWM
- Input counter mode: Detect the time of set PWM wave numbers
- RSD mode: Rotating State Detection (tailwind/headwind detection) mode

Timer2 features:

- 3-bit programmable prescaler divides the system clock
- 16-bit up counting Basic Timer; Counting clock source serves as the output of prescaler
- 16-bit up/down-counting special counter for input counter mode and RSD mode, with external input signal selected as clock source.
- Input filter module
- Edge detection module
- PWM generation module
- Interrupt event

15.1.1 Prescaler

Frequency prescaler is used to divide the system clock and generate clock source for Basic Timer. 8 frequency division coefficients of prescaler are available and can be selected by TIM2_CR0[T2PSC]. Since this register has no buffer, the clock source frequency is updated immediately after TIM2_CR0[T2PSC] is written. Therefore, the frequency division coefficients shall be configured when Basic Timer is not working. The clock source frequency formula is: clk_psc2 = SYSCLK/(2^TIM2_CR0[T2PSC]). The clock rate corresponding to different TIM2_CR0[T2PSC] value as shown in Table 15-1.

Table 15-1 Mapping between Clock rate and TIM2_CR0[T2PSC]

TIM2_CR0[T2PSC]	Coefficient	clk_psc2 (Hz)	TIM2_CR0[T2PSC]	Coefficient	clk_psc2 (Hz)
000	1	24M	100	16	1.5M
001	2	12M	101	32	750K
010	4	6M	110	64	375K
011	8	3M	111	128	187.5K

15.1.2 Reading, Writing and Counting of TIM2__CNTR

When TIM2_CR1[T2CEN] = 1, TIM2__CNTR starts to count. The write operation to TIM2__CNTR directly changes the value of the register, so Basic Timer shall be disabled before the write operation. When reading

TIM2__CNTR, software reads the 8 high-order bits first, and the hardware synchronously caches the 8 low-order bits. When reading the low byte, software reads the cached data.

15.1.3 Output Mode

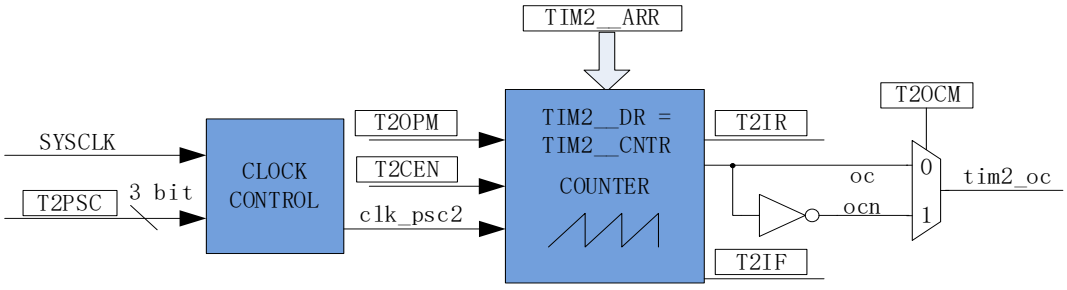


Figure 15-1 Schematic Block Diagram of Output Mode

The output mode generates output signals according to TIM2_CR0[T2OCM], and the comparison results between TIM2__CNTR and registers TIM2__DR, TIM2__ARR. Meanwhile, corresponding interrupts are generated.

15.1.3.1 Reading and Writing of TIM2__ARR/TIM2__DR

In output mode, TIM2__ARR/TIM2__DR contains preload registers and shadow registers. When the software writes TIM2__ARR/TIM2__DR register, the data is saved in the preload register. When the overflow event TIM2_CR1[T2IF] is generated or the Basic Timer stops working (TIM2_CR1[T2CEN] = 0), the set value is transferred to the shadow register.

TIM2__ARR/TIM2__DR is a 16-bit register, which requires to write the high byte first and then the low byte. The hardware ensures that the data in preload register is not transferred to shadow register after the high byte is written and before the low byte is written.

For example, TIM2__DR is a preload register and DR_SH is a shadow register. PWM is generated by comparing TIM2__CNTR with DR_SH. When software writes TIM2__DR, TIM2__DR is not updated to DR_SH immediately, and is updated to TIM2__DR at the end of a PWM (TIM2__CNTR overflow event).

15.1.3.2 High/Low Level Output

When TIM2_CR0[T2OCM] = 0, if TIM2__DR > TIM2__ARR, the output signal is always low. When TIM2_CR0[T2OCM] = 1, if TIM2__DR > TIM2__ARR, the output signal is always high.

15.1.3.3 PWM Generation

In PWM generation mode, TIM2__ARR determines PWM cycle, TIM2__DR determines duty cycle, and duty cycle = TIM2__DR/TIM2__ARR*100%. If TIM2_CR0[T2OCM] = 0, the low level is output when TIM2__CNTR < TIM2__DR, and the high level is output when TIM2__CNTR ≥ TIM2__DR. If

TIM2_CR0[T2OCM] = 1, the high level is output when TIM2__CNTR < TIM2__DR, and the low level is output when TIM2__CNTR ≥ TIM2__DR. When TIM2__CNTR is increased to TIM2__ARR, the output signal is reversed.

15.1.3.4 Interrupt Event

- When TIM2__CNTR = TIM2__DR, a compare match event is generated and the interrupt flag bit TIM2_CR1[T2IR] is set to 1. The timer continues.
- When TIM2__CNTR = TIM2__ARR, an overflow event is generated, and the interrupt flag bit TIM2_CR1[T2IF] is set to 1. The timer is cleared to 0 and then restarts.

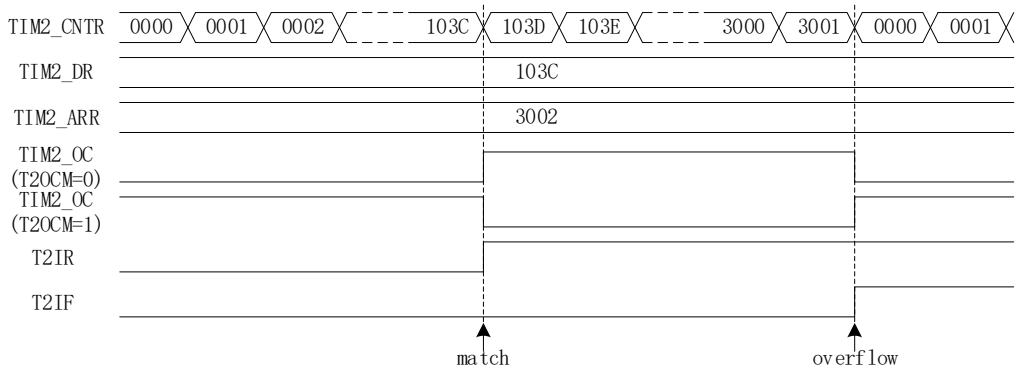


Figure 15-2 Output Mode Waveform

15.1.4 Input Signal Filtering and Edge Detection

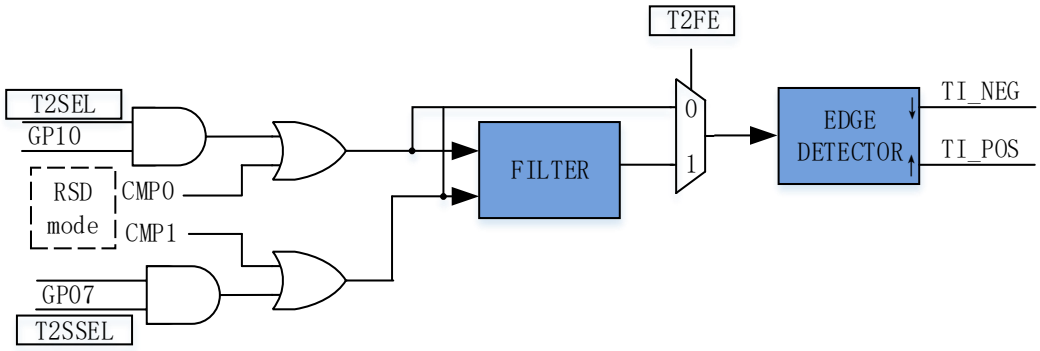


Figure 15-3 Block Diagram of Input Signal Filtering and Edge Detection

The input signal of Timer2 comes from P0.7 or P1.0, set by PH_SEL[T2SEL] and PH_SEL [T2SSEL] (see section 21.3.14). The filter of input signal is optional.

The filtering feature is enabled when TIM2_CR1[T2FE] is set to 1, and filtering circuit filters out the input noise below 4 system clock cycles. The filtered signal is 4 clock cycles later than the signal before filtering.

TIM2_CR0[T2CES] determines the active edge to count.

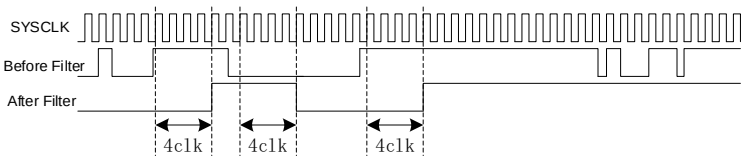


Figure 15-4 Timing Diagram of Filter Module

The edge detection module detects filtered input signals and records rising edge and falling edge for the input capture mode or input counting mode.

15.1.5 Input Capture Mode

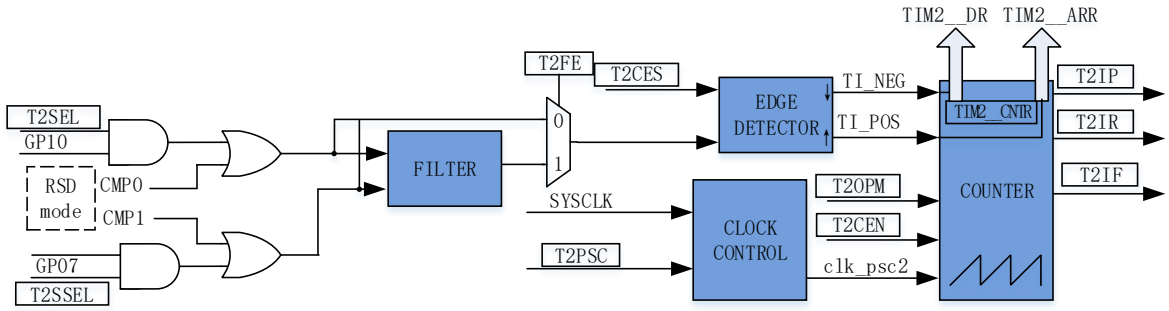


Figure 15-5 Schematic Block Diagram of Input Capture Mode

The input capture mode detects duty cycle and period of the PWM signal. When TIM_CR0[T2CES] = 0, the time between two adjacent rising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). When TIM_CR0[T2CES] = 1, the time between two adjacent falling edges forms one cycle, and the time from falling edge to rising edge forms the pulse width (LOW). When the predefined edge arrives, the count value TIM2__CNTR is stored in TIM2__DR and TIM2__ARR respectively to calculate the period and duty cycle of PWM waveform. The input signal can be filtered or not.

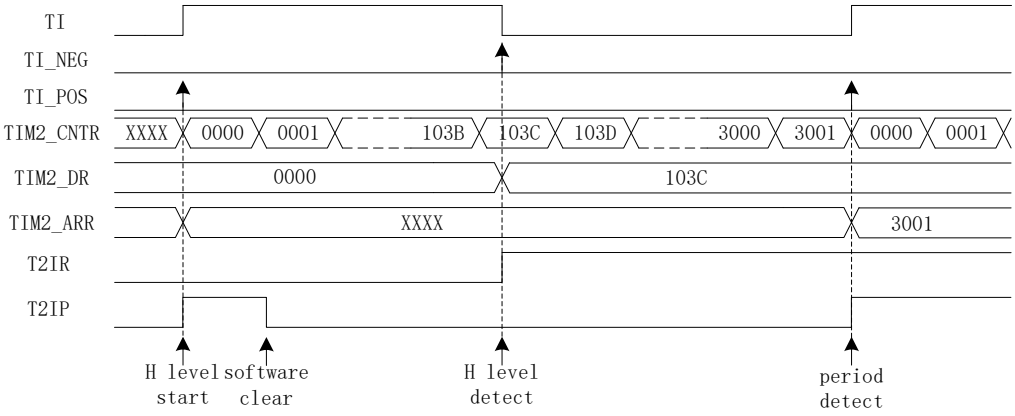


Figure 15-6 Timing Diagram of Input Capture Mode (T2CES = 0)

For example, when TIM2_CR0[T2CES] = 0, TIM2_CR1[T2CEN] is set to 1 to enable the counter. When the first rising edge of the input (falling edge is invalid) is detected, TIM2__CNTR is cleared and restarts. When falling edge of the input is detected, the value of TIM2__CNTR is stored in TIM2__DR, while the interrupt flag TIM2_CR1[T2IR] is set to 1, and TIM2__CNTR continues to count. When the second rising edge of input is detected, the value of TIM2__CNTR is stored in TIM2__ARR. Meanwhile, the interrupt flag TIM2_CR1[T2IP] is set to 1, and TIM2__CNTR is cleared to 0 and restarts.

An overflow event occurs if Timer2 does not detect the second rising edge of the input and TIM2__CNTR reaches 0xFFFF. In this case, the interrupt flag bit TIM2_CR1[T2IF] is set to 1, and TIM2__CNTR is cleared to 0 and restarts counting.

15.1.6 Input Counter Mode

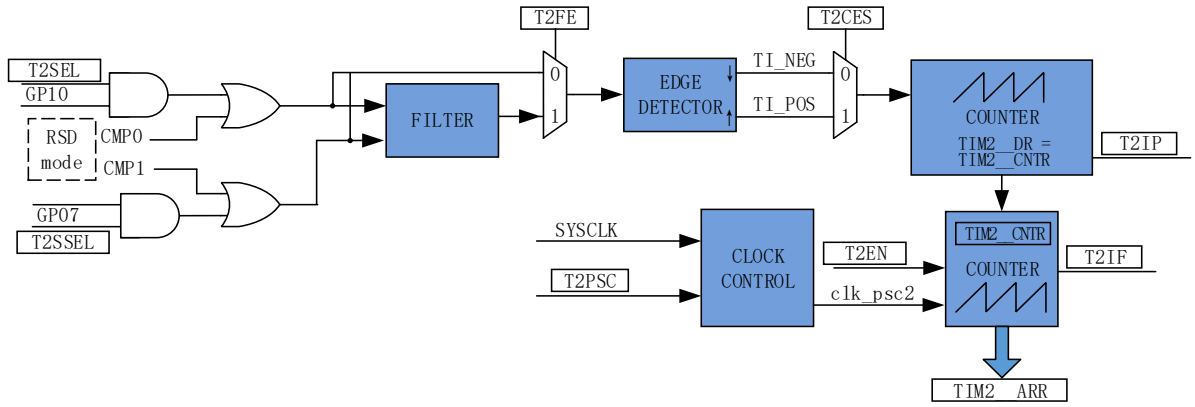


Figure 15-7 Schematic Block Diagram of Input Counter Mode

In input counter mode, TIM2__DR includes preload register and shadow register. When the software writes TIM2__DR register, the data is saved in the preload register first and sent to the shadow register in case of compare match event (TIM2_CR1[T2IP] = 1), overflow event (TIM2_CR1[T2IF] = 1) or counter disable (TIM2_CR1[T2CEN] = 0). TIM2__DR is a 16-bit register, which requires the software writes the high byte first

and then the low byte. The hardware ensures that the data in the preload register is not updated to the shadow register after the high byte is written and before the low byte is written.

The input counter mode is used to detect the time to input the set PWM wave. When the number of input PWM counted by the special counter CCNTR reaches the set value (TIM2_DR), TIM2__CNTR of the Basic Timer is stored in TIM2__ARR. The input signal can be filtered or not. When TIM2_CR0[T2CES] is set to 1, rising edge of the input PWM signal serves as the active counting edge of the special counter; when TIM2_CR0[T2CES] is set to 0, falling edge of the input PWM signal as the active counting edge.

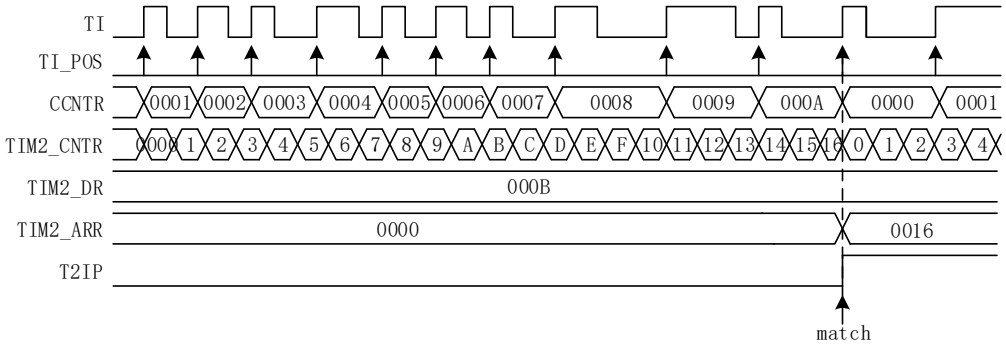


Figure 15-8 Timing Diagram of Input Counter Mode

The Basic Timer is enabled when TIM2_CR1[T2CEN] is set to 1. If the first active edge of the input signal is detected, TIM2__CNTR is cleared to 0 and restarts counting. Whenever active edge of the input signal arrives, one is added to the count value of the special counter CCNTR. When the count value reaches TIM2__DR, TIM2__CNTR is stored in TIM2__ARR. Meanwhile, TIM2_CR1[T2IP] is set to 1, TIM2__CNTR and CCNTR are cleared to 0 and restart counting.

When the number of input PWM does not reached the set value and TIM2__CNTR reaches 0xFFFF, an overflow event generates, and the interrupt flag TIM2_CR1[T2IF] is set to 1. TIM2__CNTR is cleared to 0 with CCNTR uncleared. TIM2__CNTR starts counting from 0, and CCNTR continues counting with the previous value.

15.1.7 RSD Mode

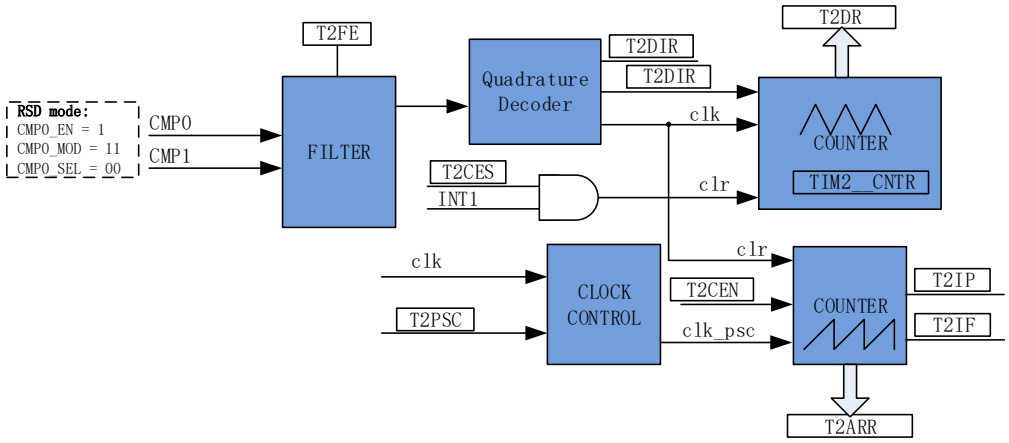


Figure 15-9 Schematic Block Diagram of RSD Mode

RSD mode obtains relative position, direction and speed of the motor by detecting orthogonal signals on the two channels. CMP0 or CMP1 are the input signal sources, which are sent to the quadrature decoding module from the filtering module to obtain active edge and direction (TIM2_CR1[T2DIR]). Direction change generates TIM2_CR1[T2IR] interrupt flag.

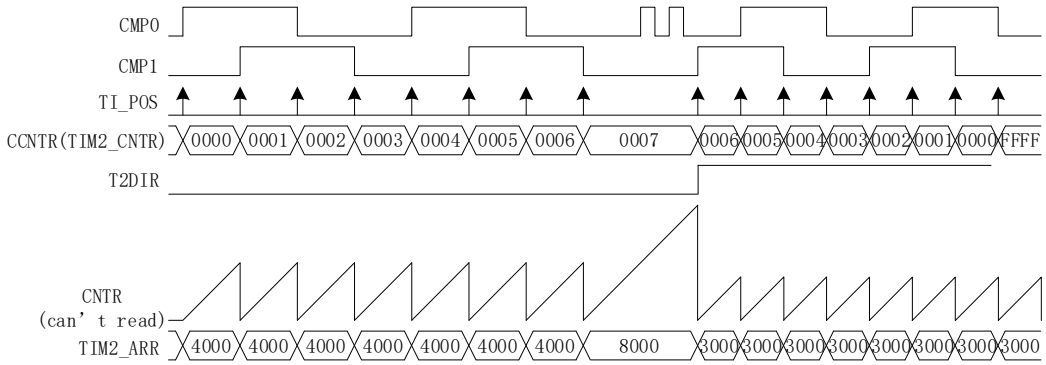


Figure 15-10 Timing Diagram of RSD Mode

The special counter is an up/down counter, and the signal source is the active edge from orthogonal decoding module. If TIM2_CR1[T2DIR] = 0, the direction is positive, and the special counter counts upward. When the active edge arrives, the counter increases by one. If TIM2_CR1[T2DIR] = 1, the direction is reverse and the special timer counts downward. When the active edge arrives, the counter decreases by one. The special counter can be cleared to 0 by INT1. INT1 is enabled after mechanical zero point of the encoder is connected to any port of INT1. When TIM2_CR0[T2CES] is set to 1 and INT1 is generated, current count value of the special counter is stored in TIM2__DR and cleared to 0. If count value of the special counter reaches 65535 from 0, it is automatically cleared to 0. If it decreases from 65535 to 0, it is automatically set to 65535. TIM2__CNTR is read to obtain the value of special counter.

The Basic Timer is an up-counter used to record the time of two active counting edges. The clock source frequency can be divided. When counting edge arrives, the value of Basic Timer is stored in TIM2__ARR and cleared to 0, and TIM2_CR1[T2IP] interrupt flag bit is set to 1.When Basic Timer counts to 0xFFFF, the count overflows and TIM2_CR1[T2IF] interrupt flag is generated.

15.1.7.1 RSD Comparator Sampling

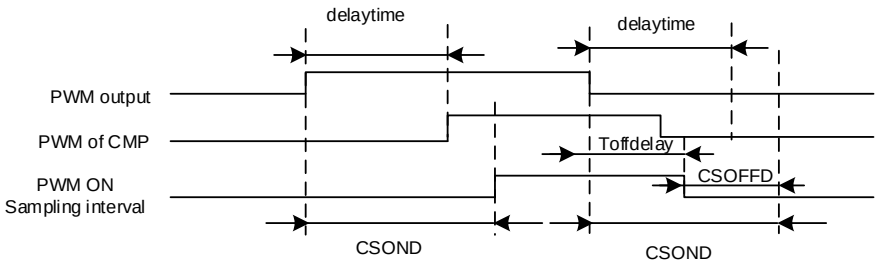


Figure 15-11 PWM ON Sampling mode

During RSD sampling, in order to obtain the accurate BEMF comparison signal, it is necessary to set the ON-delayed sampling time and Off-lead sampling time.

For details, see section 28.1.4.

15.2 TIM2 Registers

15.2.1 TIM2_CR0(0xA1)

Bit	7	6	5	4	3	2	1	0
Name	T2PSC			T2OCM	T2IRE	T2CES	T2MOD	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:5]	T2PSC	Base Timer Clock Source Frequency Division Selection This bit divides system clock as the clock source for Basic Timer The clock source frequency after frequency division is as follows: 000: 24MHz 001: 12MHz 010: 6MHz 011: 3MHz 100: 1.5MHz 101: 750kHz 110: 375kHz 111: 187.5kHz
[4]	T2OCM	Output mode: Output mode selection 0: Output 0 when TIM2__CNTR < TIM2__DR; and output 1 when TIM2__CNTR ≥ TIM2__DR. 1: Output 1 when TIM2__CNTR < TIM2__DR; and output 0 when TIM2__CNTR ≥ TIM2__DR. Input capture mode: No effect Input counter mode: No effect RSD mode selection 0: RSD mode
[3]	T2IRE	Output mode: Compare match interrupt enable Input capture mode: Pulse width detection interrupt enable Input counter mode: No effect RSD mode: Direction change interrupt enable 0: Disable 1: Enable
[2]	T2CES	Output mode: No effect Input capture mode: Counting edge selection 0: the time between two adjacent rising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). 1: the time between two adjacent falling edges forms one cycle, and the time from falling edge to rising edge forms the pulse width (LOW). Input counter mode: Counting edge selection 0: Falling edge count 1: Rising edge count RSD mode: INT1 (zero point) clear special counter enable 0: Disable 1: Enable
[1:0]	T2MOD	Mode selection 00: Input capture mode 01: Output mode 10: Input counter mode 11: RSD mode

15.2.2 TIM2_CR1(0xA9)

Bit	7	6	5	4	3	2	1	0
Name	T2IR	T2IP	T2IF	T2IPE	T2IFE	T2FE	T2DIR	T2CEN
Type	R/W0	R/W0	R/W0	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	—	0

Bit	Name	Description
[7]	T2IR	Output mode: Compare match interrupt flag Input capture mode: Pulse width detection interrupt flag Input counter mode: No effect RSD Mode: Direction Change Interrupt Enable Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[6]	T2IP	Output mode: No effect Input capture mode: PWM cycle detected interrupt flag Input counter mode: Input PWM count match interrupt flag RSD mode: active edge detected interrupt flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[5]	T2IF	Output mode: Basic Timer overflow interrupt flag. This bit is set to 1 when the counter value TIM2__CNTR matches with the comparison value TIM2__ARR Input capture mode: Base Timer overflow interrupt flag. This bit is set to 1 when the counter value TIM2__CNTR reaches 0xFFFF and timer does not detect the input of a PWM cycle. Input counter mode: Counter overflow interrupt flag bit. This bit is set to 1 when the number of the input PWM does not reach the value of TIM2__DR and the counter value TIM2__CNTR reaches 0xFFFF. RSD mode: Base Timer overflow interrupt flag bit. This bit is set to 1 and Basic Timer is cleared to 0 when Basic Timer reaches to 0xFFFF. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[4]	T2IPE	Output mode: No effect Input capture mode: PWM cycle detected interrupt enable Input counter mode: Input PWM count match interrupt enable 0: Disable 1: Enable
[3]	T2IFE	Output mode: Base Timer overflow interrupt enable Input capture mode: Base Timer overflow interrupt enable Input counter mode: Base Timer overflow interrupt enable 0: Disable 1: Enable
[2]	T2FE	Input signal filtering enable 0: Disable 1: Enable
[1]	T2DIR	RSD: Motor rotation direction Rotation direction of the motor is determined according to the phase relationship of the two input signals.
[0]	T2CEN	Basic Timer Enable 0: Disable 1: Enable

15.2.3 TIM2__CNTR(0xAA,0xAB)

TIM2__CNTRH(0xAB)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM2__CNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2__CNTRL(0xAA)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM2__CNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIM2__CNTR	Output mode/Input capture mode/Input counter mode: Count value of the Basic Timer RSD mode: Count value of the special counter

15.2.4 TIM2__DR (0xAC,0xAD)

TIM2__DRH(0xAD)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM2__DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2__DRL(0xAC)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM2__DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIM2__DR	Output mode: Compare match values (written by software) Input capture mode: Count value of the detected input pulse width (written by hardware) Input counter mode: The number of PWM needs to be counted (written by software) RSD mode: The value of the special counter (written by hardware) when TIM2_CR0[T2CES] = 1 and INT1 (zero point) arrives.

15.2.5 TIM2__ARR(0xAE,0xAF)

TIM2__ARRH(0xAF)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM2__ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2__ARRL(0xAE)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM2__ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**

[15:0]	TIM2__ARR	Output mode: PWM waveform cycle (written by software) Input capture mode: Count value of the Basic Timer of a detected PWM cycle (written by hardware) Input counter mode: Count value of the Basic Timer when the input PWM count matches (written by hardware) RSD mode: Count value of the Basic Timer when the input edge is detected as an active edge (written by hardware)

16 Timer3/Timer4

16.1 Timer3/Timer4 Instructions

Timer3/Timer4 support output and input modes:

- Output mode: Generate PWM
- Input capture mode: Detect the duration of high and low level of input PWM, which can be used to calculate PWM duty cycle

Timer3/Timer4 Features:

- 3-bit programmable prescaler divides system clock as the clock source for Basic Timer (clock source of Timer3 can be doubled to 48MHz in input capture mode)
- 16-bit up counting Basic Timer; the output of the prescaler serves as the counting clock source
- Input signal filtering
- Input signal edge detection
- Output PWM signal, single compare output
- Interrupt event

16.1.1 Prescaler

Frequency prescaler is used to divide the system clock and generate clock source for Basic Timer. 8 frequency division coefficients of prescaler are available and can be selected by TIMx_CR0[TxPSC]. Since this register has no buffer, the clock source frequency is updated immediately after TIMx_CR0[TxPSC] is written. Therefore, the frequency division coefficients shall be configured when Basic Timer is not working. The clock source frequency formula is: clk_psc2 = SYSCLK/(2^TIMx_CR0[TxPSC]). The clock rate corresponding to different TIMx_CR0[TxPSC] value as shown in Table 16-1.

Table 16-1 Mapping between Clock rate and TIMx_CR0[TxPSC]

TIMx_CR0[TxPSC]	Coefficient	clk_pscx (Hz)	TIMx_CR0[TxPSC]	Coefficient	clk_pscx (Hz)
000	1	24M	100	16	1.5M
001	2	12M	101	32	750K
010	4	6M	110	64	375K
011	8	3M	111	128	187.5K

16.1.2 Reading, Writing and Counting of TIMx__CNTR

When TIMx_CR1[TxEN] = 1, TIMx__CNTR starts count. The write operation to TIMx__CNTR directly changes the value of the register, so the counter shall be disabled before the write operation. When reading TIMx__CNTR, the software reads the high-order byte first and then the low-order byte, and the hardware caches the low-order byte simultaneously. When reading low-order byte, the software reads the cached data.

16.1.3 Output Mode

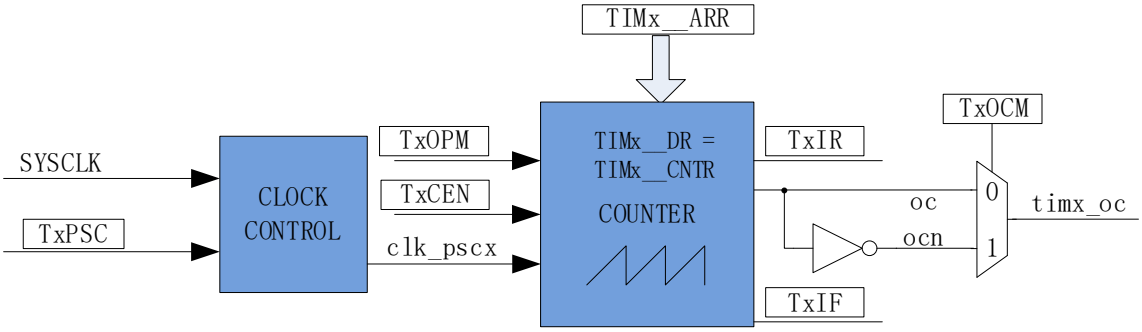


Figure 16-1 Schematic Block Diagram of Output Mode

The output mode generates output signals according to TIMx_ CR0[TxOCM], and the comparison results between TIMx__CNTR and registers TIMx__DR, TIMx__ARR. Meanwhile, corresponding interrupts are generated.

16.1.3.1 High/Low Level Output

When TIMx_CR0[TxOCM] = 0 and TIMx__DR > TIMx__ARR, the output signal is always low. When TIMx_CR0[TxOCM] = 1 and TIMx__DR > TIMx__ARR, the output signal is always high..

16.1.3.2 PWM Generation

In PWM generation mode, TIMx__ARR determines PWM cycle, TIMx__DR determines the duty cycle, and duty cycle = TIMx__DR/TIMx__ARR*100%. If TIMx_CR0[Tx_OCM] = 0, the low level is output when TIMx__CNTR < TIMx__DR, and the high level is output when TIMx__CNTR ≥ TIMx__DR. If TIMx_CR0[Tx_OCM] = 1, the high level is output when TIMx__CNTR < TIMx__DR, and the low level is output when TIMx__CNTR ≥ TIMx__DR. When TIMx__CNTR > TIMx_ARR, the output signal is reversed.

16.1.3.3 Interrupt Event

- When TIMx__CNTR = TIMx__DR, a compare match interrupt is generated. The interrupt flag TIMx_CR1[TxIR] is set to 1, and the timer continues.
- When TIMx__CNTR = TIMx__ARR, an overflow event is generated. The interrupt flag TIMx_CR1[TxIF] is set to 1, and the timer is cleared to 0. TIMx_CR0[TxOPM] determines whether the timer recounts. The timer stops when TIMx_CR0[TxOPM]= 1, and restarts when TIMx_CR0[TxOPM]= 0.

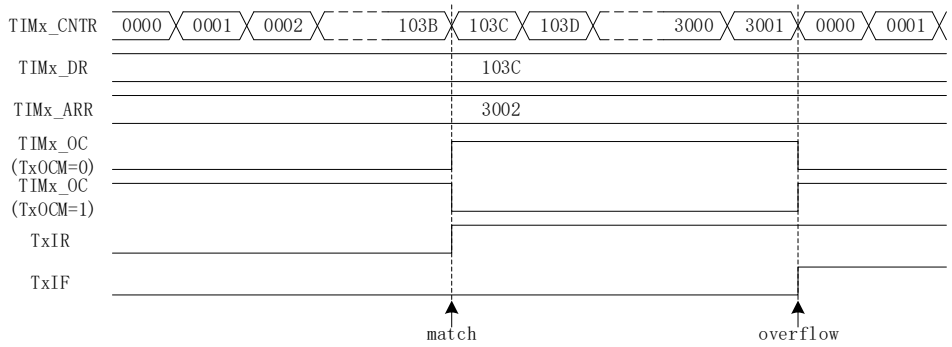


Figure 16-2 Output Waveform of Output Mode

16.1.4 Input Signal Filtering and Edge Detection

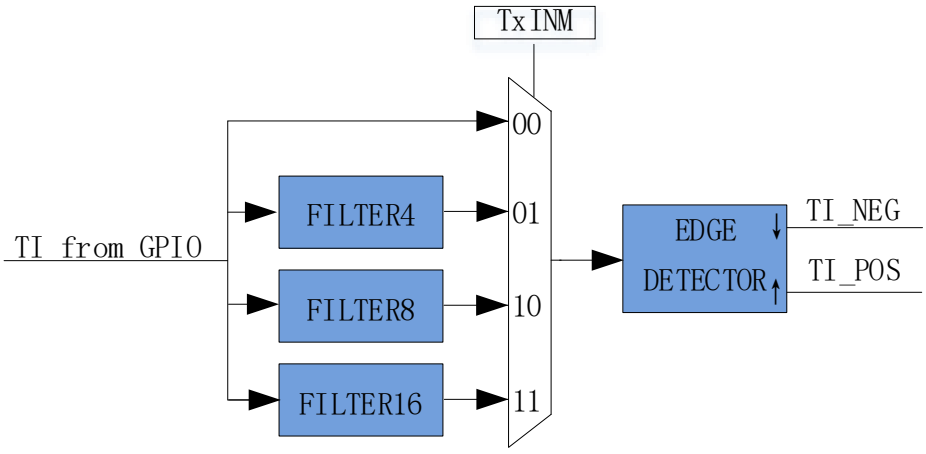


Figure 16-3 Block Diagram of Input Signal Filtering and Edge Detection

The input signals of Timer3/Timer4 come from GPIO. TIMx_CR1[TxINM] is configured to disable the filtering circuit or filter out the input noise below 4/8/16 system clock cycles. The filtered signal is 4/8/16 system clock cycles delayed than the signal before filtering.

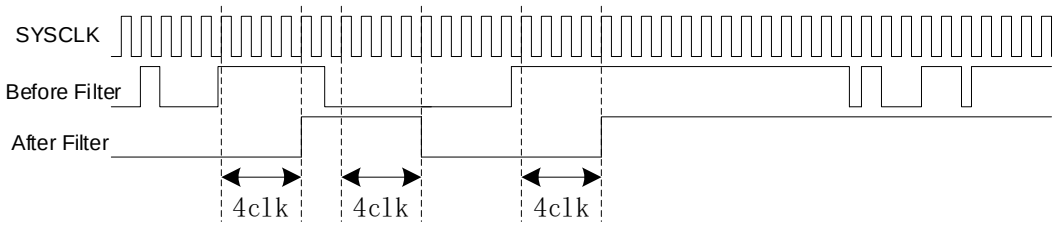


Figure 16-4 Timing Diagram of Filter Module

The edge detection module detects the filtered input signal from filtering module, and records the rising edge

and falling edge for input capture mode.

16.1.5 Input Capture Mode

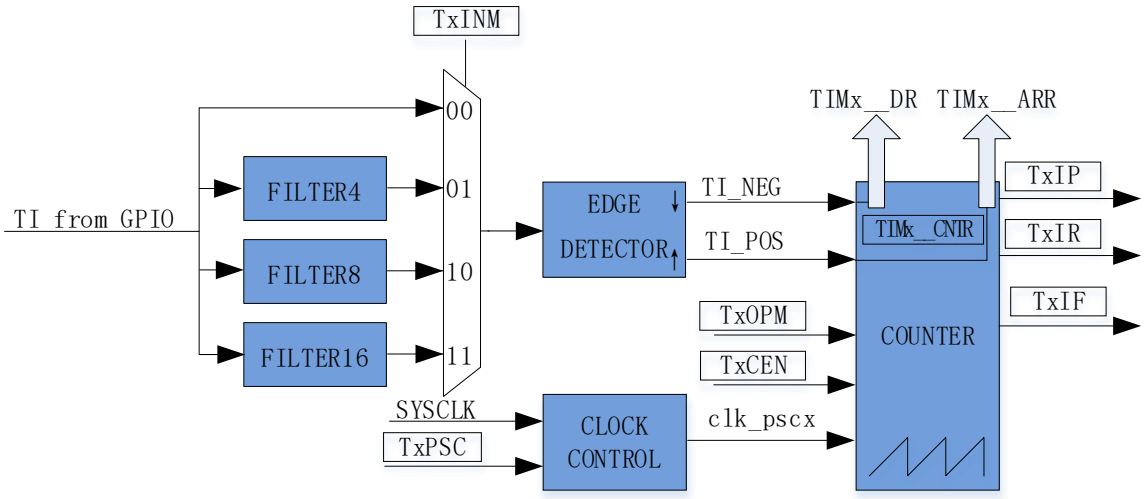


Figure 16-5 Schematic Block Diagram of Input Capture Mode

The input capture mode detects pulse width and period of the input PWM signals. When TIMx_CR0[TxOCM] = 0, the time between two adjacent rising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). When TIMx_CR0[TxOCM] = 1, the time between two adjacent falling edges forms one cycle, and the time from falling edge to rising edge forms the pulse width (LOW). The pulse width and the period obtained by TIMx__CNTR are stored in TIMx__DR and TIMx__ARR respectively.

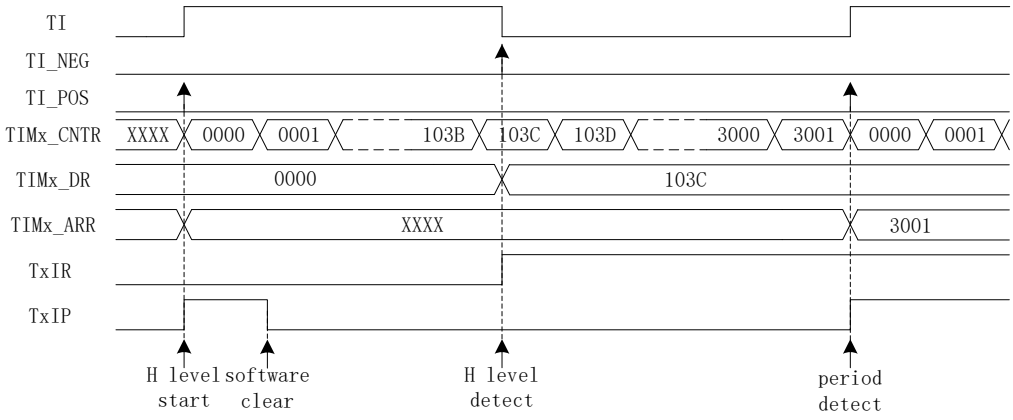


Figure 16-6 Timing Diagram of Input Capture Mode (TIMx_CR0[TxOCM] = 0)

For example, when TIMx_CR0[TxOCM]= 0, TIMx_CR1[TxEN] is set to “1” to enable the timer. When the first rising edge is detected, the counter is cleared to “0” and restarts counting. When the falling edge is detected, the value of TIMx__CNTR is stored in TIMx__DR. Meanwhile, the interrupt flag bit TIMx_CR1[TxIR] is set to “1”, and TIMx__CNTR continues to count. When the second rising edge is detected, the value of TIMx__CNTR

is saved into TIMx__ARR. The interrupt flag bit TIMx_CR1[TxIP] is set to "1" and TIMx__CNTR is cleared to "0". TIMx_CR0[TxOPM] determines whether the counter restarts. If TIMx_CR0[TxOPM] = 1, the timer stops; and if TIMx_CR0[TxOPM] = 0, it restarts.

An overflow event occurs if Timer3/4 do not detect the second rising edge of the input and the count value TIMx__CNTR reaches 0xFFFF. In this case, interrupt flag bit TIMx_CR1[TxIF] is set to "1", and TIMx__CNTR is cleared to "0". TIMx_CR0[TxOPM] determines whether the counter restarts. If TIMx_CR0[TxOPM] = 1, the timer stops; and if TIMx_CR0[TxOPM] = 0, it restarts.

16.1.6 FG Output Mode of Timer4

See FG Generation.

16.2 Timer3/Timer4 Registers

16.2.1 TIMx_CR0 (0x9C/0x9E) (x = 3/4)

Bit	7	6	5	4	3	2	1	0
Name	TxPSC			TxOCM	TxIRE	RSV	TxOPM	TxMOD
Type	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W
Reset	0	0	0	0	0	—	0	0

Bit	Name	Description
[7:5]	TxPSC	Base Timer Clock Source Frequency Division Selection This bit divides the system clock as the clock source for Basic Timer The clock source frequency after frequency division is as follows: 000: 24MHz 001: 12MHz 010: 6MHz 011: 3MHz 100: 1.5MHz 101: 750kHz 110: 375kHz 111: 187.5kHz Note: In the input capture mode of Timer3, 111 corresponds to 48MHz
[4]	TxOCM	Output mode: Output mode selection 0: Output 0 when TIM2__CNTR < TIM2__DR; and output 1 when TIM2__CNTR ≥ TIM2__DR. 1: Output 1 when TIM2__CNTR < TIM2__DR; Output 0 when TIM2__CNTR ≥ TIM2__DR. Input capture mode: Active edge selection 0: The time between two adjacent raising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). 1: The time between two adjacent falling edges forms one cycle, and the time from falling edge to raising edge forms the pulse width (LOW).
[3]	TxIRE	Output mode: Compare match interrupt enable Input capture mode: Pulse width detected interrupt enable 0: Disable 1: Enable
[2]	RSV	Reserved
[1]	TxOPM	Single Mode The counter stops when the following events occur Output mode: Base Timer overflows Input capture mode: PWM cycle is detected or Basic Timer overflows 0: Base Timer continues 1: Base Timer stops (TIMx_CR1[TxEN] is cleared to 0)

[0]	TxMOD	Working Mode Selection 0: Input capture mode 1: Output mode

16.2.2 TIMx_CR1 (0x9D/0x9F) (x = 3/4)

Bit	7	6	5	4	3	2	1	0
Name	TxIR	TxIP	TxIF	TxIPE	TxIFE	TxINM		TxEN
Type	R/W0	R/W0	R/W0	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	TxIR	Output mode: Compare match interrupt flag Input capture mode: Pulse width detected interrupt flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[6]	TxIP	Output mode: No effect Input capture mode: PWM cycle detected interrupt flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[5]	TxIF	Output mode: Base Timer overflow interrupt flag. This bit is set to 1 when TIMx__CNTR matches with TIMx__ARR. Input capture mode: Base Timer overflow interrupt flag. This bit is set to 1 when the Timer does not detect the input PWM cycle and TIMx__CNTR reaches 0xFFFF. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[4]	TxIPE	Output mode: No effect Input capture mode: PWM cycle detected interrupt enable. 0: Disable 1: Enable
[3]	TxIFE	Output mode: Base Timer overflow interrupt enable Input capture mode: Base Timer overflow interrupt enable 0: Disable 1: Enable
[2:1]	TxINM	Input Signal Filtering Pulse Width Selection When pulse width of the input signal is less than the set time, it is filtered as noise 00: No filtering 01: 4 system clock cycles, 4*41.67ns 10: 8 system clock cycles, 8*41.67ns 11: 16 system clock cycles, 16*41.67ns
[0]	TxEN	Basic Timer Enable 0: Disable 1: Enable

16.2.3 TIMx__CNTR (0xA2, 0xA3/0x92, 0x93) (x = 3/4)

TIMx__CNTRH(0xA3/0x93)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIMx__CNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIMx__CNTRL(0xA2/0x92)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIMx__CNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIMx__CNTR	Count Value of Basic Timer

16.2.4 TIMx__DR (0xA4, 0xA5/0x94, 0x95) (x = 3/4)

TIMx__DRH(0xA5/0x95)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIMx__DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIMx__DRL(0xA4/0x94)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIMx__DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIMx__DR	Output mode: Compare match values (written by software) Input capture mode: Count value of the detected input pulse width (written by hardware)

16.2.5 TIMx__ARR (0xA6, 0xA7/0x96, 0x97) (x = 3/4)

TIMx__ARRH(0xA7/0x97)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIMx__ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIMx__ARRL(0xA6/0x96)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIMx__ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIMx__ARR	Output mode: Reload value (written by software). See section FG Generation for FG mode. Input capture mode: Count value of a detected PWM cycle (written by hardware)

17 Systick

17.1 Systick Instructions

The chip can generate Systick interrupts with a fixed frequency, and the interrupt cycle is controlled by SYST_ARR. Systick interrupt is enabled when DRV_SR[SYSTIE] is set to “1”, and the interrupts are accessed by P10.

17.2 Systick Registers

17.2.1 DRV_SR (0x4061)

Bit	7	6	5	4	3	2	1	0
Name	SYSTIF	SYSTIE	FGIF	DCIF	FGIE	DCIP	DCIM	
Type	R/W0	R/W	R/W0	R/W0	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	SYSTIF	Systick Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[6]	SYSTIE	Systick Interrupt Enable 0: Disable 1: Enable
[5]	FGIF	FG Interrupt Flag When FOC Drive/ Square Wave Drive is enabled, an FGIF interrupt is generated in each rotation cycle (electrical cycle) Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[4]	DCIF	Driver Compare Match Interrupt Flag When the Driver counter value is equal to DRV_COMR, the system decides whether to generate an interrupt according to the counting direction set by DRV_SR[DCIM] Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[3]	FGIE	FG Interrupt Enable After the interrupt is enabled, one FG interrupt is generated in each electric cycle during FOC drive/square wave control 0: Disable 1: Enable
[2]	DCIP	Number of PWM Cycles Generating Compare Match Interrupt 0: 1 interrupt in 1 PWM cycle 1: 1 interrupt in 2 PWM cycles

[1:0]	DCIM	Compare Match Interrupt Mode Selection When the count value is equal to DRV_COMR, whether to generate an interrupt request is determined by DRV_SR[DCIM]. 00: No interrupt is generated. 01: Interrupt is generated when the counter counts up. 10: Interrupt is generated when the counter counts down. 11: Interrupt is generated when the counter counts up/down.

17.2.2 SYST_ARR (0x4064, 0x4065)

SYST_ARRH(0x4064)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SYST_ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	1	1	0	1
SYST_ARRL(0x4065)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SYST_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	1	1	1	1	1

Bit	**Name**	**Description**
[15:0]	SYST_ARR	Systick Reload Value This value determines the cycle at which Systick interrupts are generated. The default time is 1ms. Calculation formula: Systick interrupt frequency = SYSCLK/(SYST_ARR[15:0] + 1) Range [0,65535]

18 Driver

18.1 Driver Instructions

18.1.1 Driver Introduction

The chip has built-in pre-driver output.

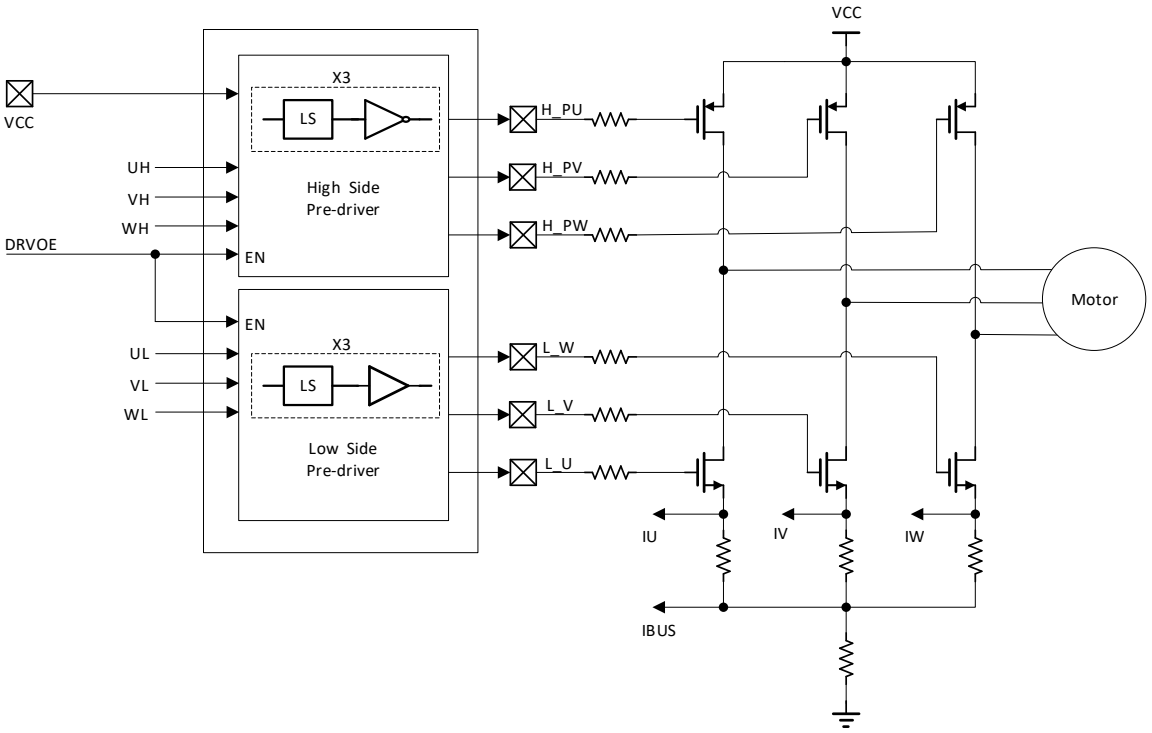


Figure 18-1 Block Diagram of Pre-driver Module

3P3N Pre-driver module is shown in Figure 18-1. UH/VH/WH and UL/VL/WL, 3-phase PWM signal, are the input signals. H_PU/H_PV/H_PW and L_U/L_V/L_W are the output signals. H_PU/H_PV/H_PW are reversely related to UH/VH/WH, and DRV_CR[DRVOE] is the enable bit.

Pre-driver module is enabled when DRV_CR[DRVOE] is set to 1. UH/VH/WH are reversely and sent to H_PU/H_PV/H_PW for driving gate of PMOS. UL/VL/WL are sent to L_U/L_V/L_W for driving gate of NMOS. PMOS and NMOS output voltages to drive motors.

18.1.2 Output Control Module

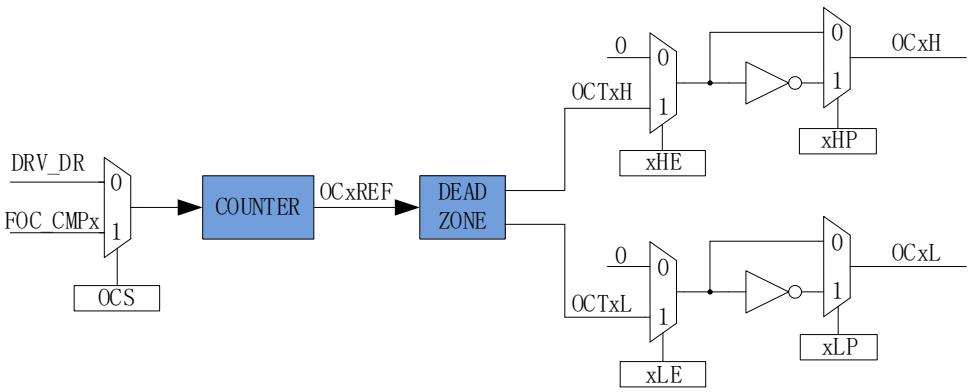


Figure 18-2 Block Diagram of Output Control Module Pre-stage

Before Driver module works, DRV_CR[MESEL] is set to "1" to select FOC mode or to "0" to select square-wave control mode.

If DRV_CR[OCS] = 0, comparison value of PWM comes from DRV_DR, and reference source of the output PWM signal is OCTxH. When OCxH and OCxL are output at the same time, OCTxL outputs reverse signal. If DRV_CR[OCS] = 1, comparison value of PWM comes from FOC module, and reference source of the output PWM signal is OCTxL. When OCxH and OCxL are output at the same time, OCTxH outputs reverse signal.

18.1.2.1 Count and Compare Module

DRV_CR[OCS] is configured to select the comparison value of PWM from FOC_CMPU/V/W of FOC module or DRV_DR set by software. The comparison value is sent to the counter for comparison to obtain the three-phase original PWM signal OCxREF, and DRV_DR is used for motor pre-charging, braking and square-wave control. If DRV__CNTR is smaller than the comparison value, OCxREF outputs high-level signal, and if DRV__CNTR is larger than DRV_DR, OCxREF outputs low-level signal.

When DRV_CR[OCS] = 1, FOC_CMPU/V/W is compared with the count value to generate the duty cycle OC1REF/OC2REF/OC3REF.

When DRV_CR[OCS] = 0, DRV_DR set by software is compared with the count value to generate OC1REF/OC2REF/OC3REF with the same duty cycle. Duty cycle = DRV_DR/DRV_ARR*100%.

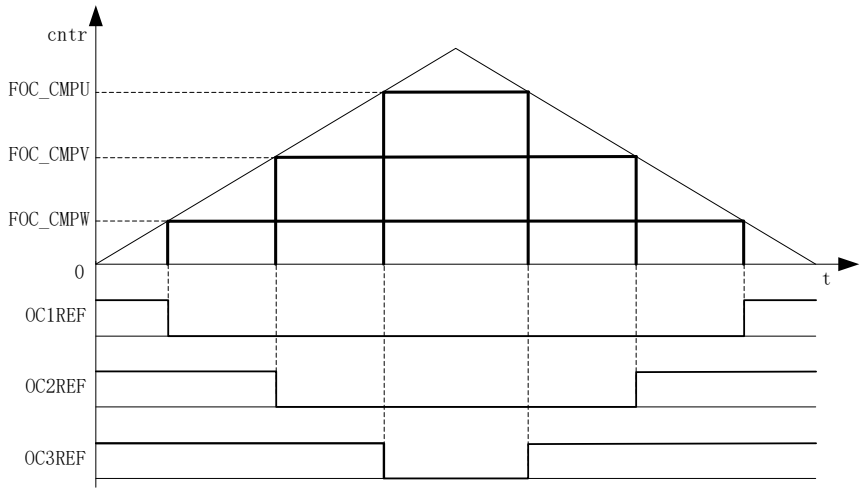


Figure 18-3 PWM Generation Diagram

18.1.2.2 Deadtime Module

OCxREF has hardware deadtime insertion. Each channel has an 8-bit deadtime generator, and three channels have the same dead time, which is set by DRV_DTR. In the rising edge of OCxREF, the delay time for OCxL to generate high-level output is the one set by DRV_DTR. In the falling edge of OCxREF, the delay time for OCxH to generate high-level output is the one set by DRV_DTR.

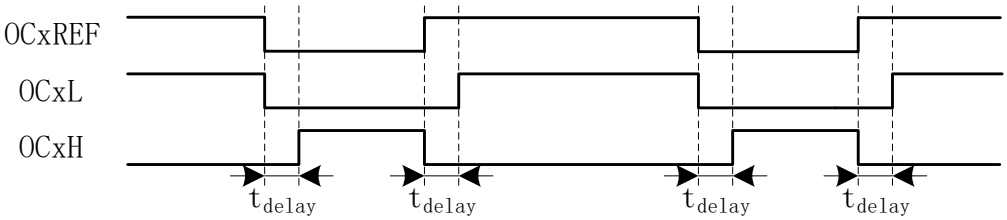


Figure 18-4 Complementary Output with Deadtime Insertion

18.1.2.3 Enable and Polarity of Output Signals

DRV_CMR[xHE] and [xLE] are configured to select the output mode as idle-level output or complementary output with deadtime inserted. DRV_CMR[xHP] and [xLP] are configured to select the polarity of output. In the application of square wave control, Timer1 automatically controls DRV_CMR to implement phase commutation feature. When DRV_CR[MESEL] is set to 0, square-wave control mode is enabled. After Timer1 generates a writing sequence event, the data stored in the corresponding TIM1_DBRx is transferred to DRV_CMR.

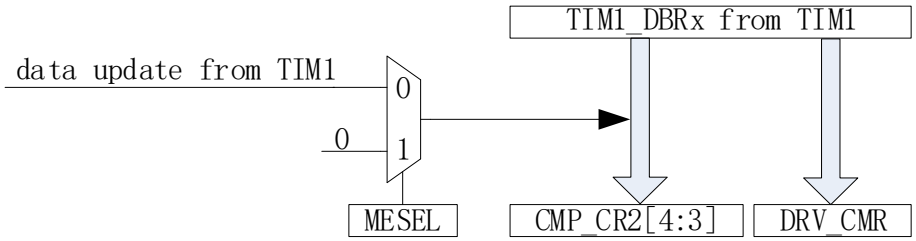


Figure 18-5 Timer1 Automatically Controls DRV_CMR and CMP_CR2[4:3]

DRV_DR, DRV_ARR and DRV_CMR can be configured to implement pre-charging, brake, etc. DRV_DR and DRV_ARR control the duty cycle and frequency of PWM. DRV_CMR[xHE] and DRV_CMR[xLE] control the output modes.

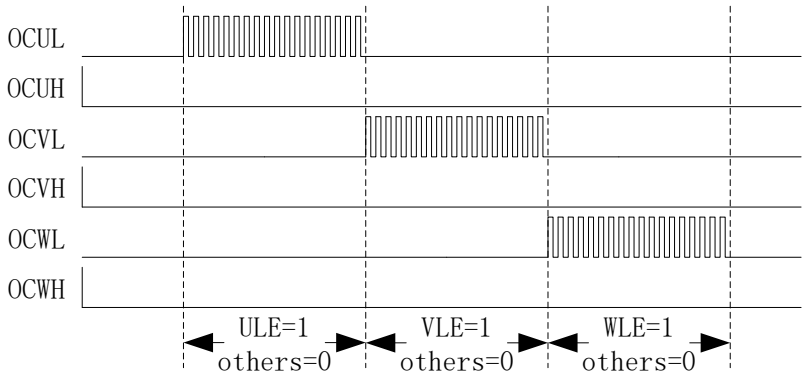


Figure 18-6 Waveform Diagram of Pre-charging

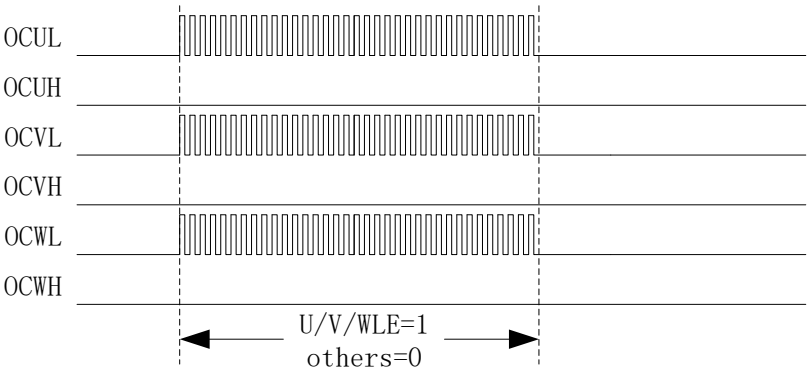


Figure 18-7 Waveform Diagram of Brake

18.1.2.4 MOE

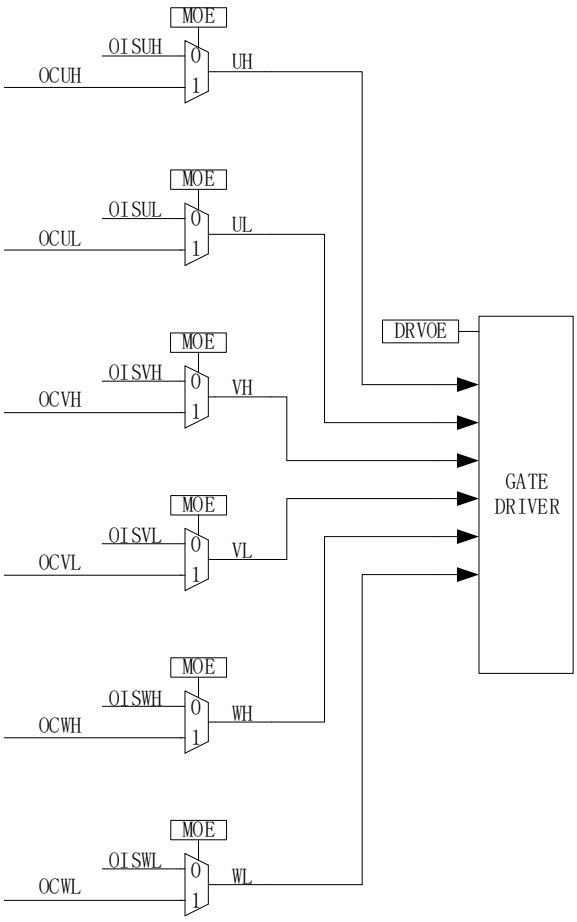


Figure 18-8 Block Diagram of Rear Stage of Output Control Module

When DRV_OUT[MOE] is enabled, PWM waveform is generated for motor driving. When DRV_OUT[MOE] is disabled, the idle level set by software is output to stop motor driving.

18.1.2.5 Interrupt

18.1.2.5.1 Compare Match Interrupt

The generation conditions and time for compare match interrupt are configured by DRV_SR[DCIM] and DRV_COMR respectively. When the counter reaches to the value set in DRV_COMR and the conditions set by DRV_SR[DCIM] are met, a compare match interrupt is generated and the interrupt flag DRV_SR[DCIF] is set to 1 by hardware.

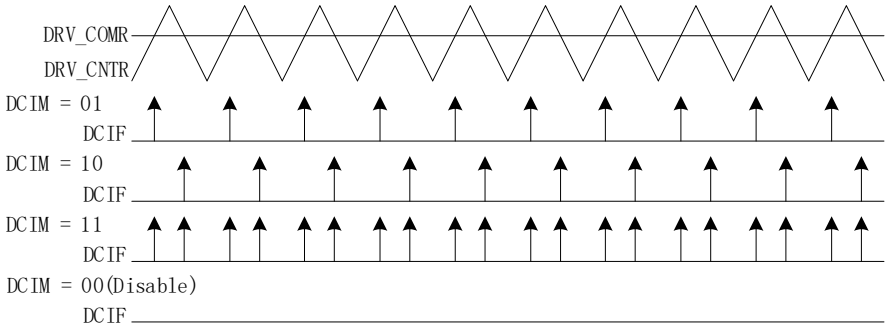


Figure 18-9 Compare Match Interrupt of Driver

18.1.2.5.2 FG Interrupt

FG interrupt is enabled when DRV_SR[FGIE] is set to "1". The motor generates an interrupt for every electrical cycle.

18.2 Driver Registers

18.2.1 DRV_CR (0x4062)

Bit	7	6	5	4	3	2	1	0
Name	DRVEN	DDIR	FOCEN	DRPE	OCS	MESEL	RSV	DRVOE
Type	R/W	R/W	R/W	R/W	R/W	R/W	—	R/W
Reset	0	0	0	0	0	0	—	0

Bit	Name	Description
[7]	DRVEN	Counter Enable 0: Disable 1: Enable
[6]	DDIR	Output Direction (Forward or Reverse) This bit is valid for both square-wave control and FOC. In sensorless FOC mode, setting this bit changes motor rotation. In Hall-sensored FOC mode, it is also required to modify the angle by the software. In square-wave control mode, parameters related to Timer1 shall be configured. 0: Forward 1: Reverse
[5]	FOCEN	FOC Module Enable 0: Disable 1: Enable
[4]	DRPE	DRV_DR Preload Enable When preload is enabled, the data written to DRV_DR is updated after a timer underflow event occurs. When preload is disabled, the data written to DRV_DR is updated immediately. 0: Disable 1: Enable
[3]	OCS	Comparison Source Selection 0: DRV_DR 1: FOC module
[2]	MESEL	ME Operating Mode Selection 0: Square wave control mode 1: FOC mode

[1]	RSV	Reserved
[0]	DRVOE	Driver Enable 0: Disable 1: Enable

18.2.2 DRV_SR (0x4061)

Bit	7	6	5	4	3	2	1	0
Name	SYSTIF	SYSTIE	FGIF	DCIF	FGIE	DCIP	DCIM1	DCIM0
Type	R/W0	R/W	R/W0	R/W0	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	SYSTIF	Systick Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	SYSTIE	Systick Interrupt Enable 0: Disable 1: Enable
[5]	FGIF	FG Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	DCIF	Driver Compare Match Interrupt Flag When the driver counter value is equal to DRV_COMR, the system decides whether to generate an interrupt according to DRV_SR[DCIM] Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[3]	FGIE	FG Interrupt Enable After interrupt feature is enabled, one FG interrupt is generated in each electric cycle under FOC/square-wave control mode. 0: Disable 1: Enable
[2]	DCIP	Number of PWM Cycles Generating Compare Match Interrupt 0: 1 PWM cycle 1: 2 PWM cycles
[1:0]	DCIM	Compare Match Interrupt Mode Selection When the count value is equal to DRV_COMR, the system decides whether to generate an interrupt request according to DRV_SR[DCIM]. 00: No interrupt is generated. 01: Interrupt is generated when the counter counts up. 10: Interrupt is generated when the counter counts down. 11: Interrupt is generated when the counter counts up/down.

18.2.3 DRV_OUT (0xF8)

Bit	7	6	5	4	3	2	1	0
Name	MOE	RSV	OISWL	OISWH	OISVL	OISVH	OISUL	OISUH
Type	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	—	0	0	0	0	0	0

Bit	Name	Description
[7]	MOE	Main Output Enable This bit is used to select the source for high and low sides of the bridge of 3-phase output signals. It can be set to 1 and cleared to 0 by software. When bus current protection is generated (see section 28.1.1.1), this bit is automatically cleared to 0 by the hardware to turn off the output. 0: Disable. The output is from the idle levels set by DRV_OUT[OISUH/OISVH/OISWH] and DRV_OUT[OISUL/OISVL/OISWL] 1: Enable. The output is from the comparison value of the timer
[6]	RSV	Reserved
[5]	OISWL	Output Idle Level of WL See descriptions on DRV_OUT[OISUH].
[4]	OISWH	Output Idle Level of WH See descriptions on DRV_OUT[OISUH].
[3]	OISVL	Output Idle Level of VL See descriptions on DRV_OUT[OISUH].
[2]	OISVH	Output Idle Level of VH See descriptions on DRV_OUT[OISUH].
[1]	OISUL	Output Idle Level of UL See descriptions on DRV_OUT[OISUH].
[0]	OISUH	Output Idle Level of UH This bit sets the output idle level of UH. When DRV_OUT[MOE] = 0, it outputs idle level and disables MOS. 0: Low level 1: High level

18.2.4 DRV_CMR (0x405C, 0x405D)

DRV_CMRH(0x405C)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				WHP	WLP	VHP	VLP
Type	—	—	—	—	R/W	R/W	R/W	R/W
Reset	—	—	—	—	0	0	0	0
DRV_CMRL(0x405D)								
Bit	7	6	5	4	3	2	1	0
Name	UHP	ULP	WHE	WLE	VHE	VLE	UHE	ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11]	WHP	Output Polarity of WH 0: Active High 1: Active Low
[10]	WLP	Output Polarity of WL 0: Active High 1: Active Low

[9]	VHP	Output Polarity of VH 0: Active High 1: Active Low
[8]	VLP	Output Polarity of VL 0: Active High 1: Active Low
[7]	UHP	Output Polarity of UH 0: Active High 1: Active Low
[6]	ULP	Output Polarity of UL 0: Active High 1: Active Low
[5]	WHE	Output Enable of WH 0: Disable 1: Enable
[4]	WLE	Output Enable of WL 0: Disable 1: Enable
[3]	VHE	Output Enable of VH 0: Disable 1: Enable
[2]	VLE	Output Enable of VL 0: Disable 1: Enable
[1]	UHE	Output Enable of UH 0: Disable 1: Enable
[0]	ULE	Output Enable of UL 0: Disable 1: Enable

Notes:

- When DRV_CMR[W/V/ULE] and DRV_CMR[W/V/UHE] are 1 at the same time, high-side and low-side outputs of W/V/U-phases are complementary to PWM signals with deadtime insertion. Low side output is the reference polarity
- In square-wave control mode, Timer1 automatically controls DRV_CMR register.

18.2.5 DRV_ARR (0x405E, 0x405F)

DRV_ARRH(0x405E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV		DRV_ARR [13:8]					
Type	—	—	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	—	0	0	0	0	0	0
DRV_ARRL(0x405F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DRV_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:14]	RSV	Reserved
[13:0]	DRV_ARR	Counter reload value, which determines PWM frequency (central alignment mode) Driver counter counts from 0 to DRV_ARR/2 - 1 and an overflow event occurs.

		Then it counts down to 0. Calculation formula: $f_{carrier}$ = 48MHz/DRV_ARR Note: LSB is always 0. Writing 1 to LSB is meaningless. The value of DRV_ARR is calculated by clock of 48MHz, and the range is [0,16382]

18.2.6 DRV_COMR (0x405A, 0x405B)

DRV_COMRH(0x405A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DRV_COMR[11:8]			
Type	—	—	—	—	R/W	R/W	R/W	R/W
Reset	—	—	—	—	0	0	0	0
DRV_COMRL(0x405B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DRV_COMR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:12]	RSV	Reserved
[11:0]	DRV_COMR	Counter Compare Match Value The Driver compare match interrupt is generated when the count value is equal to DRV_COMR and the conditions set by DRV_SR[DCIM] are met. The clock rate for the calculation of DRV_COMR is 12MHz Duty cycle of matched point = DRV_COMR*4/DRV_ARR*100% Range: [0,4095]

18.2.7 DRV_DR (0x4058, 0x4059)

DRV_DRH(0x4058)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV		DRV_DR[13:8]					
Type	—	—	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	—	0	0	0	0	0	0
DRV_DRL(0x4059)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DRV_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:14]	RSV	Reserved
[13:0]	DRV_DR	PWM Duty Cycle Setting Duty cycle = DRV_DR/DRV_ARR*100% The clock rate for the calculation of the DRV_DR is 48MHz Range: [0,16382] Note: When this register is used as a comparison source, PWM is referenced to high side of the bridge and a deadtime is inserted in the complementary output of the low side of bridge.

18.2.8 DRV_DTR (0x4060)

Bit	7	6	5	4	3	2	1	0
Name	DRV_DTR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DRV_DTR	Deadtime Setting Deadtime = (DRV_DTR + 1)*41.67ns For example, when DRV_DTR = 11, Deadtime = 12*41.67ns = 500ns Note: When DRV_DTR = 0, no deadtime is inserted.

18.2.9 DRV__CNTR (0x4066, 0x4067)

DRV__CNTRH(0x4066)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DRV__CNTR[11:8]			
Type	—	—	—	—	R/W	R/W	R/W	R/W
Reset	—	—	—	—	0	0	0	0
DRV__CNTRL(0x4067)								
Bit	7	6	5	4	3	2	1	0
Name	DRV__CNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:0]	DRV__CNTR	Counter Value The clock rate for the calculation of the DRV__CNTR is 12MHz Duty cycle of driver = DRV__CNTR*4/DRV_ARR*100% Range: [0,4095] Note: DRV__CNTR can only be written when DRV_CR[DRVEN] = 1

19 WDT

The watchdog timer (WDT) is a timer that works on the internal slow clock to monitor the master program operation and prevent the MCU running out. Watchdog works as follows: After watchdog operates, WDT starts counting. When WDT overflows, watchdog sends a signal to reset the MCU and the program restarts running from address 0. During the operation of master program, WDT has to be initialized at regular intervals to prevent WDT overflowing.

After being enabled, WDT starts counting from 0. When it reaches 0xFFFC, watchdog outputs a signal that is 4 internal slow clock cycles wide to reset the MCU, and the program starts running from address 0. WDT has to be initialized at regular intervals during operation, and the WDT rolls over to WDT_ARR setting and restarts counting.

19.1 WDT Notes

- When MCU enters standby or sleep mode, WDT stops counting, but the count values are retained.
- WDT is automatically disabled during emulation.
- RST_SR[RSTWDT] is set to 1 when MCU is reset by WDT counter overflow.

19.2 WDT Operations

1. Set CCFG1[WDT_EN] = 1 to start the WDT which then starts counting from 0;
2. Set WDT_ARR (this operation can also be performed before starting WDT);
3. Set WDT_CR[WDTRF] = 1 in the running of program. The WDT rolls over to WDT_ARR setting.

19.3 WDT Registers

19.3.1 WDT_CR (0x4026)

Bit	7	6	5	4	3	2	1	0
Name	RSV							WDTRF
Type	—	—	—	—	—	—	—	R/W
Reset	—	—	—	—	—	—	—	0

Bit	Name	Description
[7:1]	RSV	Reserved
[0]	WDTRF	WDT Initialization 0: No effect 1: WDT rolls over to the WDT_ARR setting and restarts counting.

19.3.2 WDT_ARR (0x4027)

Bit	7	6	5	4	3	2	1	0
Name	WDT_ARR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	WDT_ARR	WDT Reload Timer Set 8 high-order bits of the initialized value of the watchdog.

19.3.3 CCFG1 (0x401E)

Bit	7	6	5	4	3	2	1	0
Name	RSV	LVWIE	WDT_EN	RSV				
Type	—	R/W	R/W	—	—	—	—	—
Reset	—	0	0	—	—	—	—	—

Bit	Name	Description
[7]	RSV	Reserved
[6]	LVWIE	LVW Interrupt Enable 0: Disable 1: Enable
[5]	WDT_EN	WDT Enable 0: Disable 1: Enable
[4:0]	RSV	Reserved

20 RTC and Clock Calibration

20.1 RTC Functional Block Diagram

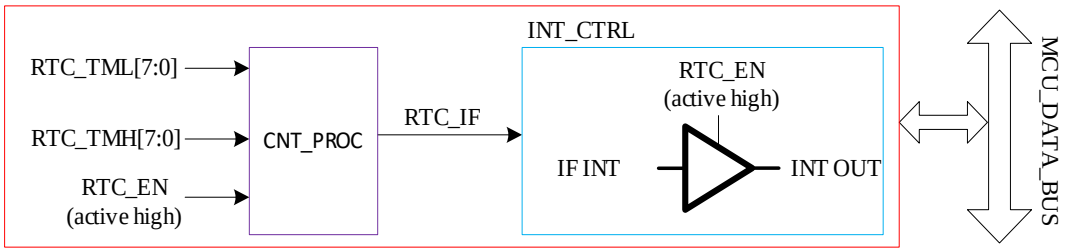


Figure 20-1 RTC Functional Block Diagram

20.2 RTC Operations

A write to RTC_TM sets the RTC reload value. RTC is enabled when RTC_STA[RTC_EN] is set to 1.

20.3 RTC Registers

20.3.1 RTC_TM (0x402C, 0x402D)

RTC_TMH(0x402C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RTC_TM[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
RTC_TML(0x402D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RTC_TM[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	**Name**	**Description**
[15:0]	RTC_TM	RTC Register Read: Instantaneous value of counter Write: RTC up-counts at a rate of 32768Hz from 0 to the written value and becomes overflowed. Then, an interrupt request is generated, causing the timer to be cleared and restart counting.

20.3.2 RTC_STA (0x402E)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RTC_EN	RTC_IF	RSV	ISOSCEN	RSV			
Type	R/W	R/W0	—	R/W	—	—	—	—
Reset	0	0	—	0	—	—	—	—

Bit	**Name**	**Description**
[7]	RTC_EN	RTC Enable 0: Disable 1: Enable

[6]	RTC_IF	RTC Interrupt Flag This bit is set to 1 when count value matches RTC_TM setting. Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect
[5]	RSV	Reserved
[4]	ISOSCEN	Internal Slow Clock Enable 0: Disable 1: Enable
[3:0]	RSV	Reserved

20.4 Clock Calibration

20.4.1 Introduction

Clock calibration is a feature that uses the internal slow clock to calibrate the internal fast clock. Calibration principle: A 12-bit counter is used to count the length of 4 slow clock cycles with the fast clock as the clock source.

Calibration operations: Set CAL_CR0[CAL_STA] = 1 to start the calibration. Read CAL_CR0[CAL_BUSY] in software to check whether the calibration is finished. When the calibration is completed (CAL_CR0[CAL_BUSY] = 0), the readout of CAL_CR0[CAL_ARR] is the value of the length of counting 4 slow clock cycles.

20.4.2 Clock Calibration Registers

20.4.2.1 CAL_CR0 (0x4044) CAL_CR1 (0x4045)

CAL_CR0(0x4044)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	CAL_STA/ CAL_BUSY	RSV			CAL_ARR[11:8]			
Type	R/W1	—	—	—	R/W	R/W	R/W	R/W
Reset	1	—	—	—	0	0	0	0
CAL_CR1(0x4045)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CAL_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	CAL_STA /CAL_BUSY	Clock Calibration Enable Read: 0: Calibration is completed. 1: Calibration is in progress. Write: 0: No effect. 1: Clock calibration starts.
[14:12]	RSV	Reserved
[11:0]	CAL_ARR	Calibrate Count Values The value of fast clock to count 4 slow clock cycles Note: When this value is 0, it indicates that there is no corresponding slow clock input; and when this value is 0xFFF, it indicates a count overflows (slow clock is too slow or fast clock is too fast)

21 IO

21.1 IO Introduction

The chip has 22 GPIO pins: P0.0 ~ P0.1, P0.5 ~ P0.7, P1.1 ~ P1.7, P2.0 ~ P2.4, P2.7, P3.0 ~ P3.2 and P3.4.

21.2 IO Instructions

Each GPIO port pin has relevant registers to meet different application requirements. For example, P0.0 is mapped to register P0, and P1.0 to register P1. P0_OE and P1_OE registers are configured for digital input and output.

- The enable bits of pull-up resistors and pull-down resistors are configured to "1". See 21.3.9 P0_PU (0x4053) ~ 21.3.13 P4_PU (0x4057) for port pins and registers;
- See 5.3 GPIO Electrical Characteristics for the values of pull-up resistors and pull-down resistors;
- The relevant bits of P1_AN, P2_AN and P3_AN registers are configured to "1" to activate analog signal mode. See 21.3.6 P1_AN (0x4050) ~ 21.3.8 P3_AN (0x4052) for port pins and registers. After the port pins are configured to analog mode, all their digital features are disabled and the port state is 0 by reading relevant bits in P1, P2 and P3 registers;
- Pull-up resistors of P1.6 ~ P1.7, P2.0 ~ P2.4, P2.7, P3.0 ~ P3.2 and P3.4 are automatically disabled when the port pins are configured as analog mode.

21.3 IO Registers

21.3.1 P0_OE (0xFC)

Bit	7	6	5	4	3	2	1	0
Name	P0_OE							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P0_OE	P0.0 ~ P0.7 Digital Input/Output Selection 0: Input 1: Output

21.3.2 P1_OE (0xFD)

Bit	7	6	5	4	3	2	1	0
Name	P1_OE							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P1_OE	P1.0 ~ P1.7 Digital Input/Output Selection 0: Input 1: Output

21.3.3 P2_OE (0xFE)

Bit	7	6	5	4	3	2	1	0
Name	P2_OE							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P2_OE	P2.0 ~ P2.7 Digital Input/Output Selection 0: Input 1: Output

21.3.4 P3_OE (0xFF)

Bit	7	6	5	4	3	2	1	0
Name	P3_OE							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P3_OE	P3.0 ~ P3.7 Digital Input/Output Selection 0: Input 1: Output

21.3.5 P4_OE (0xE9)

Bit	7	6	5	4	3	2	1	0
Name	RSV		P4_OE[5:4]		RSV	P4_OE[2]	RSV	
Type	—	—	R/W	R/W	—	R/W	—	—
Reset	—	—	0	0	—	0	—	—

Bit	Name	Description
[7:6]	RSV	Reserved
[5:4]	P4_OE[5:4]	P4.4 ~ P4.5 Digital Input/Output Selection 0: Input 1: Output
[3]	RSV	Reserved
[2]	P4_OE[2]	P4.2 Digital Input/Output Selection 0: Input 1: Output
[1:0]	RSV	Reserved

21.3.6 P1_AN (0x4050)

Bit	7	6	5	4	3	2	1	0
Name	P1_AN				HBMOD	RSV	ODE1	ODE0
Type	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W
Reset	0	0	0	0	0	—	0	0

Bit	Name	Description
[7:4]	P1_AN	P1.4 ~ P1.7 Analog Mode Enable 0: Disable 1: Enable

<table>
<tr><td>[3]</td><td>HBMOD</td><td>P1.3 mode configuration, which determines the functional mode of P1.3 pin in combination with P1_OE[3], as shown in Table 21-1.
Table 21-1 P1.3 Mode Configuration
<table>
<tr><th>HBMOD</th><th>P1_OE[3]</th><th>P1.3 pin mode</th></tr>
<tr><td>0</td><td>0</td><td>Digital input</td></tr>
<tr><td>0</td><td>1</td><td>Digital output</td></tr>
<tr><td>1</td><td>0</td><td>Analog mode</td></tr>
<tr><td>1</td><td>1</td><td>Digital enhanced drive output mode. The maximum output current of high level output can be up to 20mA for Hall bias power supply. The drive mode of low level output is the same as that of the digital output mode.</td></tr>
</table></td></tr>
<tr><td>[2]</td><td>RSV</td><td>Reserved</td></tr>
<tr><td>[1]</td><td>ODE1</td><td>P0.1 Collector Open-drain Enable
0: Disable
1: Enable</td></tr>
<tr><td>[0]</td><td>ODE0</td><td>P0.0 Collector Open-drain Enable
0: Disable
1: Enable</td></tr>
</table>

21.3.7 P2_AN (0x4051)

Bit	7	6	5	4	3	2	1	0
Name	P2_AN							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P2_AN	P2.0 ~ P2.7 Analog Mode Enable 0: Disable 1: Enable

21.3.8 P3_AN (0x4052)

Bit	7	6	5	4	3	2	1	0
Name	P11_PL	P01_PL	P3_AN					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	P11_PL	P1.1 Pull-down Resistor Enable 0: Disable 1: Enable Note: Pull-up and pull-down resistors of P1.1 cannot be enabled at the same time.
[6]	P01_PL	P0.1 Pull-down Resistor Enable 0: Disable 1: Enable Note: Pull-up and pull-down resistors of P0.1 cannot be enabled at the same time.
[5:0]	P3_AN	P3.0 ~ P3.5 Analog Mode Enable 0: Disable 1: Enable

21.3.9 P0_PU (0x4053)

Bit	7	6	5	4	3	2	1	0
Name	P0_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P0_PU	P0.0 ~ P0.7 Pull-up Resistor Enable 0: Disable 1: Enable

21.3.10 P1_PU (0x4054)

Bit	7	6	5	4	3	2	1	0
Name	P1_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P1_PU	P1.0 ~ P1.7 Pull-up Resistor Enable 0: Disable 1: Enable

21.3.11 P2_PU (0x4055)

Bit	7	6	5	4	3	2	1	0
Name	P2_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P2_PU	P2.0 ~ P2.7 Pull-up Resistor Enable 0: Disable 1: Enable

21.3.12 P3_PU (0x4056)

Bit	7	6	5	4	3	2	1	0
Name	P3_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	P3_PU	P3.0 ~ P3.7 Pull-up Resistor Enable 0: Disable 1: Enable

21.3.13 P4_PU (0x4057)

Bit	7	6	5	4	3	2	1	0
Name	RSV		P4_PU[5]	P4_PU[4]	RSV	P4_PU[2]	RSV	
Type	—	—	R/W	R/W	—	R/W	—	—
Reset	—	—	0	0	—	0	—	—

Bit	Name	Description
[7:6]	RSV	Reserved
[5:4]	P4_PU[5:4]	P4.4 ~ P4.5 Pull-up Resistor Enable 0: Disable 1: Enable
[3]	RSV	Reserved
[2]	P4_PU[2]	P4.2 Pull-up Resistor Enable 0: Disable 1: Enable
[1:0]	RSV	Reserved

21.3.14 PH_SEL (0x404C)

Bit	7	6	5	4	3	2	1	0
Name	SPITMOD	UART1EN	UART2EN	T4SEL	T3SEL	T2SEL	T2SSEL	RSV
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—
Reset	0	0	0	0	0	0	0	—

Bit	Name	Description
[7]	SPITMOD	MISO port status after SPI slave device completes transmission 0: Output state 1: High-impendence state
[6]	UART1EN	Port multiplexed as RXD, TXD and UART1 enabled 0: Disable 1: P0.5, P0.6 multiplexed as RXD, TXD and UART1 enabled
[5]	UART2EN	Port multiplexed as RXD2, TXD2 and UART2 enabled 0: Disable 1: P3.6, P3.7 multiplexed as RXD2, TXD2 and UART2 enabled
[4]	T4SEL	Port multiplexed as Timer4 or Timer4S 0: Not multiplexed 1: P0.1 or P0.0 (PH_SEL1[T4CT] = 1) multiplexed as input and output of Timer4
[3]	T3SEL	Port multiplexed as Timer3 or Timer3S 0: Not multiplexed 1: P1.1 or P0.1 (PH_SEL1[T3CT] = 1) multiplexed as input and output of Timer3
[2]	T2SEL	Port multiplexed as Timer2 0: Not multiplexed 1: P1.0 multiplexed as input and output of Timer2
[1]	T2SSEL	Port multiplexed as Timer2S 0: Not multiplexed 1: P0.7 multiplexed as input and output of Timer2
[0]	RSV	Reserved

21.3.15 PH_SEL1 (0x404D)

Bit	7	6	5	4	3	2	1	0
Name	RSV					SPICT	T4CT	T3CT
Type	—	—	—	—	—	R/W	R/W	R/W
Reset	—	—	—	—	—	0	0	0

Bit	Name	Description
[7:3]	RSV	Reserved
[2]	SPICT	SPI Function Switching Enable 0: No function switching, with P0.5 serving as SCLK pin and P0.6 as MOSI pin 1: Function switching, with P0.0 serving as MOSI pin and P0.1 as SCLK pin Note: FU6832N1 has no NSS pin. When SPI feature is used, SPI_CR1[NSSMOD] is configured to 00 so that NSS signal is not routed to the pin.
[1]	T4CT	Timer4 Function Switching Enable 0: No function switching, with P0.1 serving as input and output of Timer4 1: Function switching, with P0.0 serving as input and output of Timer4
[0]	T3CT	Timer3 Function Switching Enable 0: No function switching, with P1.1 serving as input and output of Timer3 1: Function switching, with P0.1 serving as input and output of Timer3

21.3.16 P0 (0x80)

Port output registers P0/1/2/3/4 support read/write access. The RMW commands are used to access the value of the registers (see Table 21-2 for RMW commands), and other commands are used to access the PORT pin.

Bit	7	6	5	4	3	2	1	0
Name	GP07	GP06	GP05	GP04	GP03	GP02	GP01	GP00
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP07	Port GP07
[6]	GP06	Port GP06
[5]	GP05	Port GP05
[4]	GP04	Port GP04
[3]	GP03	Port GP03
[2]	GP02	Port GP02
[1]	GP01	Port GP01
[0]	GP00	Port GP00

21.3.17 P1 (0x90)

Bit	7	6	5	4	3	2	1	0
Name	GP17	GP16	GP15	GP14	GP13	GP12	GP11	GP10
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP17	Port GP17
[6]	GP16	Port GP16
[5]	GP15	Port GP15
[4]	GP14	Port GP14

[3]	GP13	Port GP13
[2]	GP12	Port GP12
[1]	GP11	Port GP11
[0]	GP10	Port GP10

21.3.18 P2 (0xA0)

Bit	7	6	5	4	3	2	1	0
Name	GP27	GP26	GP25	GP24	GP23	GP22	GP21	GP20
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP27	Port GP27
[6]	GP26	Port GP26
[5]	GP25	Port GP25
[4]	GP24	Port GP24
[3]	GP23	Port GP23
[2]	GP22	Port GP22
[1]	GP21	Port GP21
[0]	GP20	Port GP20

21.3.19 P3 (0xB0)

Bit	7	6	5	4	3	2	1	0
Name	GP37	GP36	GP35	GP34	GP33	GP32	GP31	GP30
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP37	Port GP37
[6]	GP36	Port GP36
[5]	GP35	Port GP35
[4]	GP34	Port GP34
[3]	GP33	Port GP33
[2]	GP32	Port GP32
[1]	GP31	Port GP31
[0]	GP30	Port GP30

21.3.20 P4 (0xE8)

Bit	7	6	5	4	3	2	1	0
Name	RSV		GP45	GP44	RSV	GP42	RSV	
Type	—	—	R/W	R/W	—	R/W	—	—
Reset	—	—	0	0	—	0	—	—

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	GP45	Port GP45
[4]	GP44	Port GP44
[3]	RSV	Reserved
[2]	GP42	Port GP42
[1:0]	RSV	Reserved

Table 21-2 RMW Commands

Command	Description
ANL	Bitwise logical AND operation
ORL	Bitwise logical OR operation
XRL	Bitwise logical XOR operation
JBC	Jump if the bit is set to“1” and then cleared to “0”
CPL	Bitwise logical converse operation
INC, DEC	+1, -1 logical operation
DJNZ	Jump if the bit is not “0”
MOV Px,y,C	Assign carry bit C to Px,y
CLR Px,y	Px,y is cleared to “0”
SETB Px,y	Px,y is set to “1”

22 ADC

22.1 ADC Introduction

The ADC module is a 12-bit successive approximation register ADC with 15 channels, where channel 0 ~ 13 are external pin channels and channel 14 is internal channel. VCC pin is divided by built-in resistor, and the result is sampled by channel 14. The ratio of voltage division is configured by ADC_CR[URATIO]. The sampling mode supports sequential sampling (i.e., from ADC channel 0 to ADC channel 14 in sequence) and triggered sampling (including FOC triggered sampling mode and Timer1 triggered sampling mode). The result of sequential sampling is stored in ADCx_DR (x = 0 ~ 14) in right-aligned or left-second-highest-bit-aligned format. The result of triggered sampling is sent to FOC module or Timer1 module instead of ADCx_DR for motor drive. The relevant registers of FOC module or Timer1 module are always left-second-highest-bit-aligned to store the triggered sample results. Triggered sampling is done automatically by hardware and sequential sampling is controlled by software. The priority of triggered sampling is higher than that of sequential sampling. If both triggered sampling and sequential sampling are applied at the same time, triggered sampling is performed first, and ADC automatically restores sequential sampling mode upon completion of triggered sampling.

The clock source of ADC is 12MHz, and the sampling time is set by DAC_CR[5:2] and ADC_SCYC. See ADC Electrical Characteristics for the sampling time and conversion time.

22.2 ADC Block Diagram

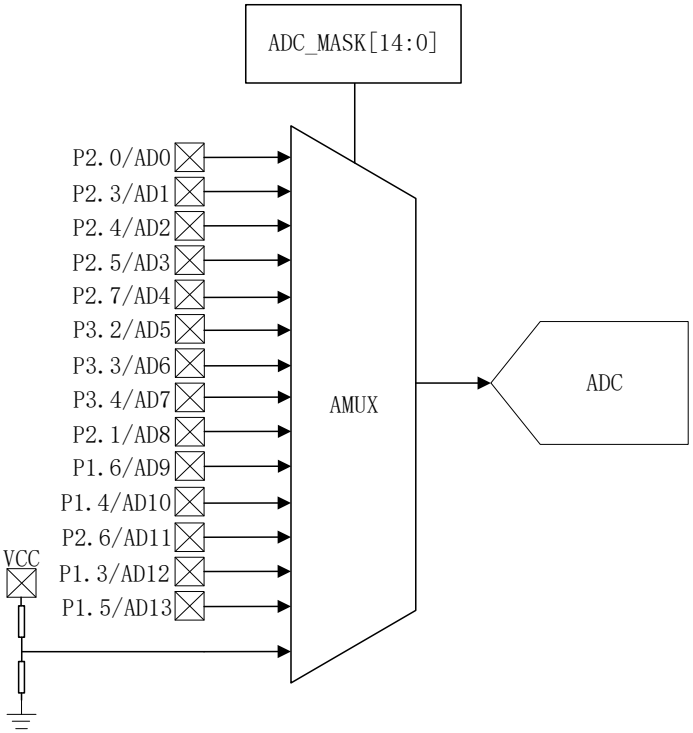


Figure 22-1 ADC Multiplexer Block Diagram

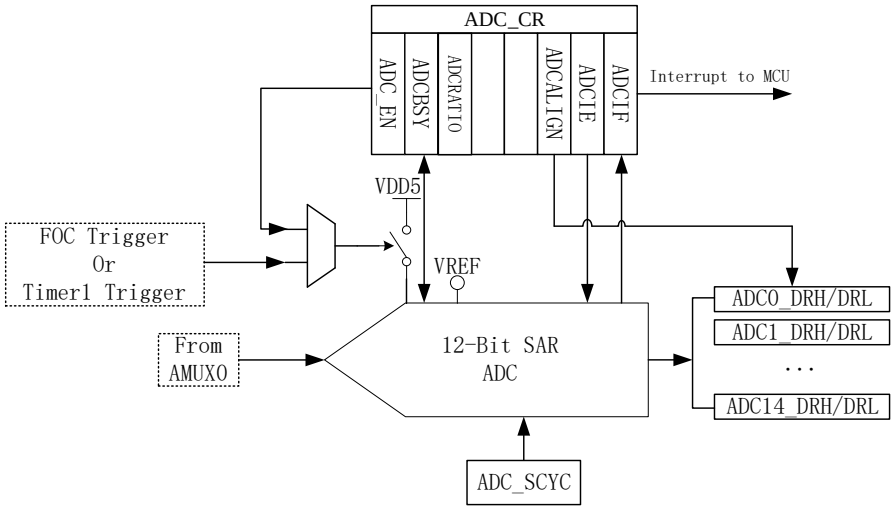


Figure 22-2 ADC Functional Block Diagram

22.3 ADC Operations

22.3.1 Sequential Sampling Mode

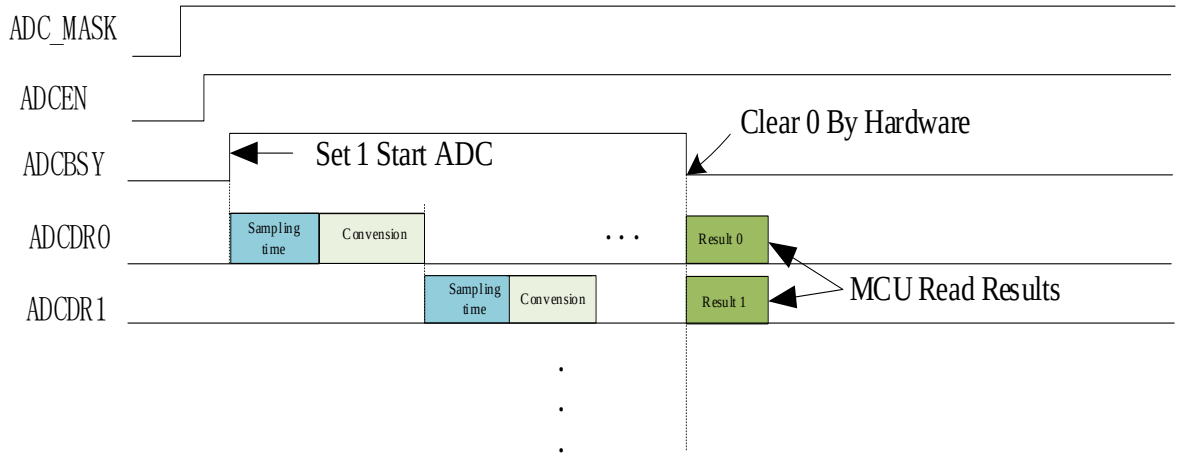


Figure 22-3 Timing Diagram of ADC Sequential Sampling Mode

ADC Operations:

1. Configure VREF_CR[VREFVSEL] to select appropriate ADC reference voltage (VREF);
2. Configure ADC_MASK to enable the corresponding channel required to sample;
3. Configure ADC_SCYC (minimum value is 3) to select the sampling period of each channel;
4. Set ADC_CR[ADCEN] to 1 to enable ADC;
5. Set ADC_CR[ADCBSY] to 1 to start ADC;
6. When ADC_CR[ADCBSY] = 0, ADC results are read by ADCx_DR.

Note: The ADC conversion sequence is from low to high according to the enabled channel (i.e., when channel

2/3/4 is enabled, the signal is sampled in order of 2/3/4, and then a single conversion result is read after confirming ADC_CR[ADCBSY] = 0).

22.3.2 Triggered Sampling Mode

When FOC module is enabled, ADC channel 0/1/2/4/14 can be used for FOC triggered sampling. FOC_CR0[UCSEL] is configured to select ADC channel 2 or channel 14 as voltage sampling channel. In single-shunt current sampling mode, ADC channel 4 is used for itrip sampling. In dual-shunt current sampling mode, ADC channel 0 is used for IA sampling and channel 1 for IB sampling. In triple-shunt current sampling mode, ADC channel 0 is used for IA sampling, channel 1 for IB sampling and channel 4 for IC sampling.

When Timer1 is enabled, ADC channel 4 is used for bus current sampling. TIM1_CR3[T1TIS] is configured to select the input source of position detection as ADC. When CMP0_CR4[CMP0FS] = 0, ADC channel 10 is used for U-phase voltage sampling, channel 9 for V-phase voltage sampling and channel 8 for W-phase voltage sampling. When CMP0_CR4[CMP0FS] = 1, ADC channel 10 is used for U-phase voltage sampling, channel 12 for V-phase voltage sampling and channel 13 for W-phase voltage sampling.

22.3.3 Output Data Format

Registers ADCx_DRH and ADCx_DRL hold the high-order bits and the low-order bits of ADC sampling results. Data can be right-aligned or left-second-highest-bit-aligned by configuring ADC_CR[ADCALIGN]. The relation between the input voltage and result data is shown as in Table 22-1. The bits, which are not used in ADCx_DRH and ADCx_DRL, are set to 0.

Table 22-1 Relation between the Input Voltage and Result Data

Input Voltage	Right-aligned	Left-second-highest-aligned
0	0x0000	0x0000
VREF/2	0x0800	0x4000
VREF	0x0FFF	0x7FF8

22.4 ADC Registers

22.4.1 ADC_CR(0x4039)

Bit	7	6	5	4	3	2	1	0
Name	ADCEN	ADCBSY	ADCRATIO	RSV		ADCALIGN	ADCIE	ADCIF
Type	R/W	R/W1	R/W	—	—	R/W	R/W	R/W0
Reset	0	0	0	—	—	0	0	0

Bit	Name	Description
[7]	ADCEN	ADC Enable 0: Disable 1: Enable
[6]	ADCBSY	ADC Start & ADC Busy Flag Read: 0: ADC idle 1: ADC busy Write: 0: No effect 1: ADC conversion starts Note: When ADC_MASK = 0, writing 1 to this bit has no effect
[5]	ADCRATIO	Division Ratio of VCC Sampling by ADC Channel 14 0: 1/12 1: 1/6.5
[4:3]	RSV	Reserved
[2]	ADCALIGN	ADC Data Format Selection 0: ADC output is right-aligned; ADC result is ADCx_DR[11:0] 1: ADC output is left-second-highest-bit-aligned; ADC result is ADCx_DR[14:3] Note: The results of triggered sampling mode are always left-second-highest-bit-aligned.
[1]	ADCIE	ADC Interrupt Enable (except triggered sampling mode) 0: Disable 1: Enable
[0]	ADCIF	ADC Interrupt Flag This bit is set to “1” by hardware when ADC conversion is completed Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect

22.4.2 ADC_MASK (0x4036, 0x4037)

ADC_MASKH(0x4036)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	CH14EN	CH13EN	CH12EN	CH11EN	CH10EN	CH9EN	CH8EN
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0
ADC_MASKL(0x4037)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CH7EN	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15]	RSV	Reserved
[14]	CH14EN	ADC channel 14 enable
[13]	CH13EN	ADC channel 13 enable
[12]	CH12EN	ADC channel 12 enable
[11]	CH11EN	ADC channel 11 enable
[10]	CH10EN	ADC channel 10 enable
[9]	CH9EN	ADC channel 9 enable
[8]	CH8EN	ADC channel 8 enable
[7]	CH7EN	ADC channel 7 enable
[6]	CH6EN	ADC channel 6 enable
[5]	CH5EN	ADC channel 5 enable
[4]	CH4EN	ADC channel 4 enable
[3]	CH3EN	ADC channel 3 enable
[2]	CH2EN	ADC channel 2 enable
[1]	CH1EN	ADC channel 1 enable
[0]	CH0EN	ADC channel 0 enable

Note: In triggered sampling mode, it is not required to configure ADC_MASK.

22.4.3 DAC_CR(0x4035)

DAC_CR (0x4035)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DACEN	DACMOD	ADC_SCYCH[3:0]				RSV	
Type	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	1	1	-	-

Bit	Name	Description
[7]	DACEN	See section DAC_CR (0x4035)in DAC chapter for details.
[6]	DACMOD	See section DAC_CR (0x4035)in DAC chapter for details.
[5:2]	ADC_SCYCH [3:0]	ADC sampling cycle for ADC channel 8 ~ 13 ADC_SCYCH[3] = 0: The sampling cycle is ADC_SCYCH [2:0] ADC clock cycles. ADC_SCYCH[3] = 1: The sampling cycle is (ADC_SCYC [2:0]*8 + 7) ADC clock cycles.
[1:0]	RSV	Reserved

22.4.4 ADC_SCYC (0x4038)

ADC_SCYC (0x4038)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC_SCYC[7:4]				ADC_SCYC[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	1

Bit	Name	Description
[7:4]	ADC_SCYC[7:4]	ADC sampling cycle for ADC channel 5 ~ 7 and 14 ADC_SCYC[7] = 0: The sampling cycle is ADC_SCYC [6:4] ADC clock cycles. ADC_SCYC[7] = 1: The sampling cycle is (ADC_SCYC [6:4]*8 + 7) ADC clock cycles.

[3:0]	ADC_SCYC[3:0]	ADC sampling cycle for ADC channel 0, 1, 3 and 4 ADC_SCYC[3] = 0: The sampling cycle is ADC_SCYC [2:0] ADC clock cycles. ADC_SCYC[3] = 1: The sampling cycle is (ADC_SCYC[2:0]*8 + 7) ADC clock cycles.

22.4.5 ADC0_DR (0x0300, 0x0301)

ADC0_DRH(0x0300)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC0_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC0_DRL(0x0301)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC0_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC0_DR	ADC channel 0 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.6 ADC1_DR (0x0302, 0x0303)

ADC1_DRH(0x0302)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC1_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC1_DRL(0x0303)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC1_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC1_DR	ADC channel 1 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.7 ADC2_DR (0x0304, 0x0305)

ADC2_DRH(0x0304)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC2_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC2_DRL(0x0305)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Name	ADC2_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC2_DR	ADC channel 2 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.8 ADC3_DR (0x0306, 0x0307)

ADC3_DRH(0x0306)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC3_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC3_DRL(0x0307)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC3_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC3_DR	ADC channel 3 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.9 ADC4_DR (0x0308, 0x0309)

ADC4_DRH(0x0308)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC4_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC4_DRL(0x0309)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC4_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC4_DR	ADC channel 4 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.10 ADC5_DR (0x030A, 0x030B)

ADC5_DRH(0x030A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC5_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC5_DRL(0x030B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC5_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC5_DR	ADC channel 5 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.11 ADC6_DR (0x030C, 0x030D)

ADC6_DRH(0x030C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC6_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC6_DRL(0x030D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC6_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC6_DR	ADC channel 6 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.12 ADC7_DR (0x030E, 0x030F)

ADC7_DRH(0x030E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC7_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC7_DRL(0x030F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC7_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC7_DR	ADC channel 7 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.13 ADC8_DR (0x0310, 0x0311)

ADC8_DRH(0x0310)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC8_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC8_DRL(0x0311)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC8_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC8_DR	ADC channel 8 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.14 ADC9_DR (0x0312, 0x0313)

ADC9_DRH(0x0312)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC9_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC9_DRL(0x0313)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC9_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC9_DR	ADC channel 9 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.15 ADC10_DR (0x0314, 0x0315)

ADC10_DRH(0x0314)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC10_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC10_DRL(0x0315)								

Bit	7	6	5	4	3	2	1	0
Name	ADC10_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC10_DR	ADC channel 10 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.16 ADC11_DR (0x0316, 0x0317)

ADC11_DRH(0x0316)								
Bit	15	14	13	12	11	10	9	8
Name	ADC11_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC11_DRL(0x0317)								
Bit	7	6	5	4	3	2	1	0
Name	ADC11_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC11_DR	ADC channel 11 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.17 ADC12_DR (0x0318, 0x0319)

ADC12_DRH(0x0318)								
Bit	15	14	13	12	11	10	9	8
Name	ADC12_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC12_DRL(0x0319)								
Bit	7	6	5	4	3	2	1	0
Name	ADC12_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC12_DR	ADC channel 12 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.18 ADC13_DR (0x031A, 0x031B)

ADC13_DRH(0x031A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC13_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC13_DRL(0x031B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC13_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	ADC13_DR	ADC channel 13 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting. Note: ADC results in triggered sampling mode are not updated to this register.

22.4.19 ADC14_DR (0x031C, 0x031D)

ADC14_DRH(0x031C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC14_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC14_DRL(0x031D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC14_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	ADC14_DR	ADC channel 14 conversion result upon completion of ADC conversion in the Sequential Sampling Mode Data alignment is selected according to ADC_CR[ADCALIGN] setting Note: ADC results in triggered sampling mode are not updated to this register.

23 DAC

23.1 DAC Introduction

The chip integrates two DAC modules, where DAC0 is a 9-bit digital-to-analog converter, and DAC1 is a 6-bit digital-to-analog converter.

23.2 DAC0 Functional Block Diagram

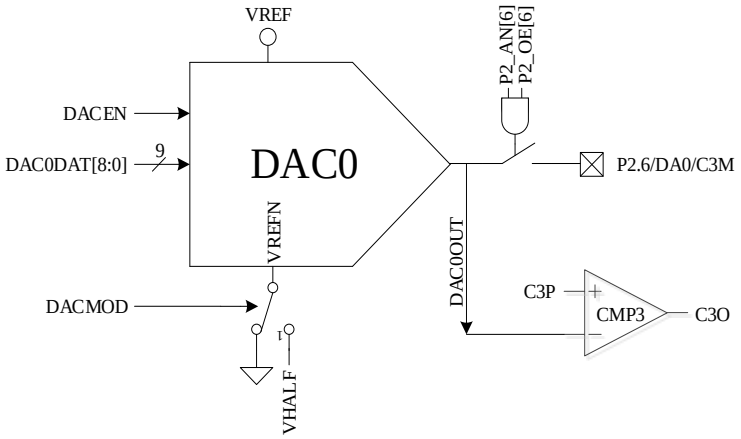


Figure 23-1 Functional Block Diagram of DAC0

As shown in Figure 23-1, DAC0 converts the 9-bit digital data to analog voltage. The voltage is sent to CMP3 negative input for bus overcurrent protection, while P2.6 can be configured as the analog output.

Notes:

1. DAC0 output has no current drive capability and can only carry capacitive load. To carry resistive load, operational amplifiers are used to follow the voltage output.
2. Configure P2_AN[6] = 1 and P2_OE[6] = 1. DAC0 outputs to P2.6/DA0;
3. Configure VREF_CR[VREFEN] to 1 to select VREF as the reference voltage for DAC0, and set DAC_CR[DAC0_1EN] to 1 to enable DAC0;
4. The range of output voltage is selected by DAC_CR[DACMOD]. When DAC_CR[DACMOD] = 0, full-voltage output mode is active, and the range of output voltage is 0 ~ VREF. When DAC_CR[DACMOD] = 1, half-voltage output mode is active, and the range of output voltage is VHALF ~ VREF. The output voltage of DAC0 in different modes are shown in Table 23-1.

Table 23-1 Voltage Output of DAC0 in Different Configurations

DAC0DAT[8:0]	DAC Output Voltage (DAC_CR[DACMOD] = 0)	DAC Output Voltage (DAC_CR[DACMOD] = 1)
0x000	0	VHALF
0x100	VREF/2	(VREF - VHALF)/2 + VHALF
0x1FF	VREF*511/512	(VREF -V HALF)*511/512 + VHALF

23.3 DAC1 Functional Block Diagram

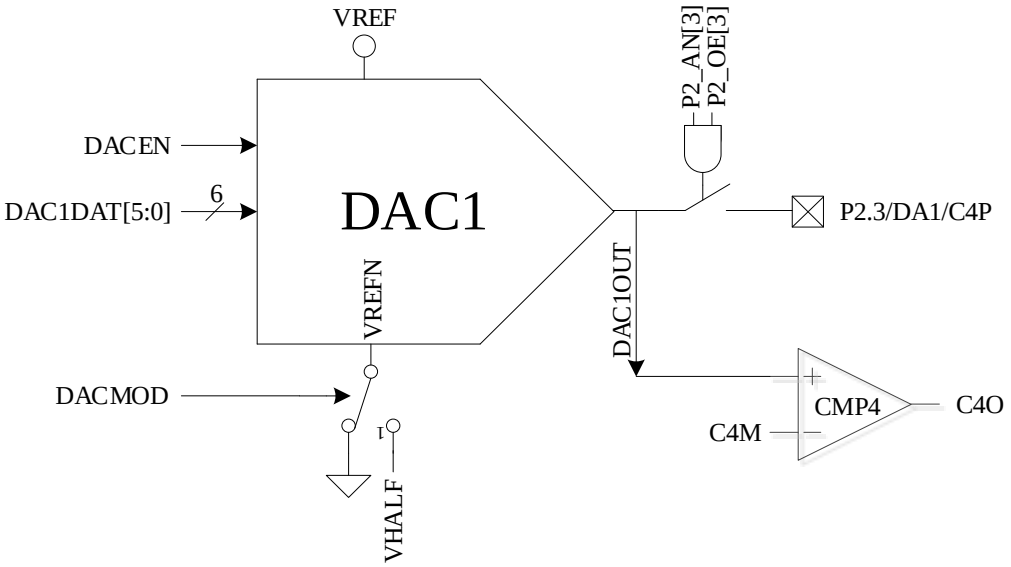


Figure 23-2 Functional Block Diagram of DAC1

As shown in Figure 23-2, DAC1 converts the 6-bit digital data to analog voltage and sends the voltage to the CMP4 positive input for cycle-by-cycle current limiting, while P2.3 can be configured as the analog output.

Notes:

1. DAC1 output no few current drive capability and can only carry capacitive load. To carry resistive load, operational amplifiers are used to follow the voltage output.
2. Configure P2_AN[3] = 1 and P2_OE[3] = 1. DAC1 outputs to P2.3/DA1;
3. Configure VREF_CR[VREFEN] to 1 to select VREF as the reference voltage for DAC1, and set DAC_CR[DAC0_1EN] to 1 to enable DAC1;
4. The range of output voltage is set by DAC_CR[DACMOD]. When DAC_CR[DACMOD] = 0, full-voltage output mode is active, and the output voltage range is 0 ~ VREF. When DAC_CR[DACMOD] = 1, half-voltage output mode is active, and the output voltage range is VHALF ~ VREF. The output voltage of DAC1 under different configurations is shown in Table 23-2.

Table 23-2 Voltage DAC1 output in Different Configurations

DAC1DAT[5:0]	DAC Output Voltage (DAC_CR[DACMOD] = 0)	DAC Output Voltage (DAC_CR[DACMOD] = 1)
0x00	0	VHALF
0x20	VREF/2	(VREF-VHALF)/2 + VHALF
0x3F	VREF*63/64	(VREF - VHALF)*63/64 + VHALF

23.4 DAC Registers

23.4.1 DAC_CR (0x4035)

Bit	7	6	5	4	3	2	1	0
Name	DAC0_1EN	DACMOD	ADC_SCYCH[3:0]				RSV	
Type	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	1	1	-	-

Bit	Name	Description
[7]	DAC0_1EN	DAC0 & 1 Enable 0: Disable 1: Enable
[6]	DACMOD	DAC Mode Selection 0: Full-voltage output mode, where DAC output voltage ranges from 0 to VREF. 1: Half-voltage output mode, where DAC output voltage ranges from VHALF to VREF.
[5:2]	ADC_SCYCH [3:0]	See DAC_CR(0x4035) in ADC chapter for details.
[1:0]	RSV	Reserved

23.4.2 DAC0_DR (0x404B)

Bit	7	6	5	4	3	2	1	0
Name	DAC0DAT[8:1]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DAC0DAT[8:1]	8 High-order Bits Input of DAC0 Controller

23.4.3 DAC1_DR (0x404A)

Bit	7	6	5	4	3	2	1	0
Name	DAC0_DR_0	RSV	DAC1DAT					
Type	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	—	0	0	0	0	0	0

Bit	Name	Description
[7]	DAC0_DR_0	LSB of DAC0 Controller
[6]	RSV	Reserved
[5:0]	DAC1DAT	6 High-order Bits Input of DAC1 Controller

24 DMA

24.1 DMA Instructions

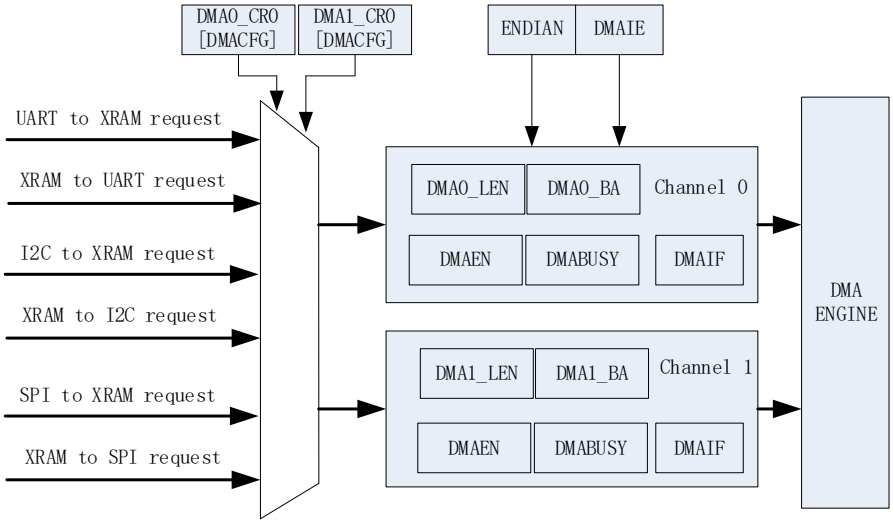


Figure 24-1 Functional Block Diagram of DMA

The DMA module is a dual-channel DMA controller, which performs direct data transfer between peripherals (SPI, UART, I^2C) and XRAM (IRAM data invalid). DMA accessing to XRAM does not interfere with the normal CPU read/write operation to XRAM. The length of the transferred data and the start address of XRAM access is configurable. Data transfer mode is configurable and interrupt can be enabled.

DMA instructions are as follows:

1. Configure the peripheral and enable the peripheral, and set input and output channels taken over by DMA by DMAx_CR0[DMACFG];
2. Configure DMA interrupt enable, transfer order, transfer length and XRAM start address. Write DMAx_CR0[DMAEN] and DMAx_CR0[DMABSY] to 1 to start DMA;
3. After data transfer, the interrupt flag bit DMAx_CR0[DMAIF] is set to 1 by hardware and it is cleared to 0 by software;
4. Set DMAx_CR0[DMABSY] to 1 to start DMA again.

24.2 DMA Registers

24.2.1 DMA0_CR0 (0x403A)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DMAIE	ENDIAN	DMAIF
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	DMAEN	DMA Channel 0 Enable 0: Disable 1: Enable
[6]	DMABSY	DMA Channel 0 Busy State/Start Read: 0: Channel 0 is idle 1: Channel 0 is transmitting Write: 0: No effect 1: DMA channel 0 starts for data transfer
[5:3]	DMACFG [2:0]	DMA Channel 0 Peripheral and Transfer Direction Selection 000: From UART1 to XRAM 001: From XRAM to UART1 010: From I^2C to XRAM 011: From XRAM to I^2C 100: From SPI to XRAM 101: From XRAM to SPI 110: From UART2 to XRAM 111: From XRAM to UART2 Note: It cannot be configured when DMA channel 0 is busy
[2]	DMAIE	DMA Interrupt Enable 0: Disable 1: Enable
[1]	ENDIAN	DMA Data Transfer Sequence 0: High byte is received or transmitted first 1: Low byte is received or transmitted first Notes: This bit is set for 16-bit data mode, and shall be configured to “0” for 8-bit data mode. It cannot be configured when channel 0 or 1 is busy
[0]	DMAIF	DMA Channel 0 Transfer Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0”. 1: The interrupt is generated.

24.2.2 DMA1_CR0 (0x403B)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DBGSW	DBGEN	DMAIF
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	DMAEN	DMA Channel 1 Enable 0: Disable 1: Enable

[6]	DMABSY	DMA Channel 1 Busy State/Start Read: 0: Channel 1 is idle 1: Channel 1 is transmitting Write: 0: No effect 1: Channel 1 starts for data transfer
[5:3]	DMACFG [2:0]	DMA Channel 1 Peripheral and Transfer Direction Selection 000: From UART1 to XRAM 001: From XRAM to UART1 010: From I^2C to XRAM 011: From XRAM to I^2C 100: From SPI to XRAM 101: From XRAM to SPI 110: From UART2 to XRAM 111: From XRAM to UART2 Note: It cannot be configured when DMA channel 1 is busy
[2]	DBGSW	Debug Area Selection 0: XSFR as the debug area (exported address space: 0x4020 ~ 0x40FF) 1: XRAM as the debug area (exported address space: 0x0000 ~ 0x0317)
[1]	DBGEN	Debug Mode Enable When DMA1_CR0[DMACFG] is set to 101 and DMA1_CR0[DBGEN] to 1, DMA works in Debug mode. After SPI is enabled, DMA automatically sends the relevant data in the area specified by DMA_CR0[DBGSW] via MOSI, and DMA1_BA/ DMA1_LEN defines the start address and range of the relevant data. 0: Disable 1: Enable Note: In Debug mode, DMA channel 1 interrupt is turned off
[0]	DMAIF	DMA Channel 0 Transfer Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: Generate interrupt

24.2.3 DMA0_LEN (0x403C)

Bit	7	6	5	4	3	2	1	0
Name	RSV		DMA0_LEN					
Type	—	—	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	—	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	DMA0_LEN	DMA Channel 0 Transfer Length Setting Read: The number of the bytes currently transferred by DMA channel 0 (0 means the first byte) Write: DMA channel 0 XRAM transfer data length Note: It cannot be configured when DMA channel 0 is busy. When DMA0_CR0[ENDIAN] = 1 (low byte is received or sent first), it is recommended that DMA0_LEN be set to an odd number.

24.2.4 DMA0_BA (0x403E, 0x403F)

DMA0_BAH(0x403E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV					DMA0_BA[10:8]		
Type	—	—	—	—	—	R/W	R/W	R/W
Reset	—	—	—	—	—	0	0	0
DMA0_BAL(0x403F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA0_BA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:11]	RSV	Reserved
[10:0]	DMA0_BA	DMA Channel 0 Transfer Data Start Address Setting DMA Channel 0 XRAM Start Address It cannot be configured when DMA channel 0 is busy Note: XRAM address space for data transfer by channel 0: DMA0_BA[10:0] ~ (DMA0_BA[10:0] + DMA0_LEN[5:0])

24.2.5 DMA1_LEN (0x403D)

Bit	7	6	5	4	3	2	1	0
Name	RSV		DMA1_LEN					
Type	—	—	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	—	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	DMA1_LEN	DMA Channel 1 Transfer Data Length Setting Read: The number of the bytes currently transferred by DMA channel 1 (0 means the first byte) Write: DMA channel 1 XRAM transfer data length Note: It cannot be configured when DMA channel 1 is busy. When DMA0_CR0[ENDIAN] = 1 (low byte is received or sent first), it is recommended that DMA1_LEN be set to an odd number.

24.2.6 DMA1_BA (0x4040, 0x4041)

DMA1_BAH(0x4040)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV					DMA1_BA[10:8]		
Type	—	—	—	—	—	R/W	R/W	R/W
Reset	—	—	—	—	—	0	0	0
DMA1_BAL(0x4041)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA1_BA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:11]	RSV	Reserved
[10:0]	DMA1_BA	DMA Channel 1 Transfer Data Start Address Setting DMA Channel 1 XRAM Start Address It cannot be configured when DMA channel 1 is busy Notes: XRAM address space for data transfer by channel 1: DMA0_BA[10:0] ~ (DMA0_BA[10:0] + DMA0_LEN[5:0])

Note: When I^2C is selected as DMA channel peripherals (including from I^2C to XRAM and from XRAM to I^2C), START + address interrupt for I^2C communication still requires to be cleared to 0 by software. In I^2C slave mode, if STOP is received, I2C_SR[I2CSTP] = 0 is configured to clear I^2C interrupt and restart the DMA transfer.

25 VREF

25.1 VREF Instructions

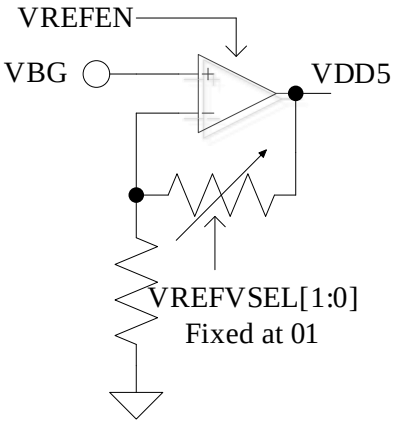


Figure 25-1 VREF Selection

The input and output ports of the VREF module are shown in Figure 25-1. VREF is the voltage reference generation block that provides internal voltage reference to ADC and DAC module. VBG is the voltage provided by the chip internally.

VREF is enabled when VREF_CR[VREFEN] is set to 1. The output voltage is selected by configuring VREF_CR[VREFVSEL]. VREF serves as the internal voltage reference to ADC and DAC module.

25.2 VREF Registers

25.2.1 VREF_CR (0x404F)

Bit	7	6	5	4	3	2	1	0
Name	VREFVSEL		RSV	VREFEN	RSV			VHALFEN
Type	R/W	R/W	—	R/W	—	—	—	R/W
Reset	0	0	—	0	—	—	—	0

Bit	Name	Description
[7:6]	VREFVSEL	VREF Module Output Voltage Selection 00: 4.5V 01: VDD5 10: 3V 11: 4V
[5]	RSV	Reserved
[4]	VREFEN	VREF Module Enable 0: Disable. P3_AN[5] is set to 1, and external voltage reference is input from P3.5 1: Enable. P3_AN[5] is set to 1, and the internal VREF reference is output to P3.5. A 0.1μF ~ 1μF capacitor is added to improve the stability of VREF.
[3:1]	RSV	Reserved
[0]	VHALFEN	VHALF Enable 0: Disable 1: Enable

Note: For FU6832N1, only VDD5 can be used as VREF voltage.

26 VHALF

26.1 VHALF Instructions

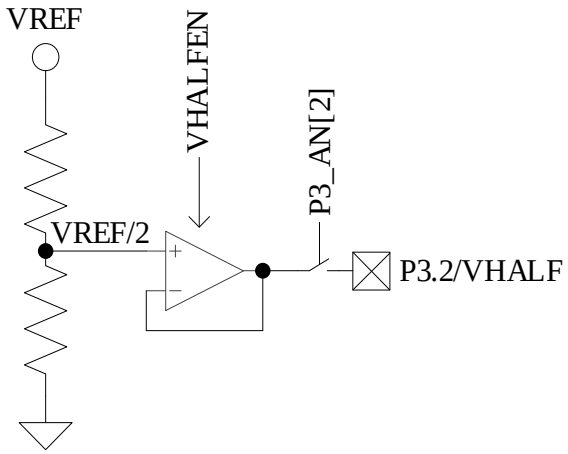


Figure 26-1 Diagram of VHALF Module Input and Output

The VHALF module input and output ports are shown in Figure 26-1. This module generates the voltage reference VREF/2.

VHALF is enabled when VREF_CR[VREFEN] is set to 1, and the voltage is output to P3.2. A 1μF capacitor is added.

26.2 VHALF Registers

See VREF_CR (0x404F).

27 Operational Amplifier

27.1 Operational Amplifier Introduction

The chip integrates three high-speed independent operational amplifiers, AMP0, AMP1, and AMP2. Each operational amplifier has a separate enable bit. AMP0 can be configured as a programmable gain amplifier.

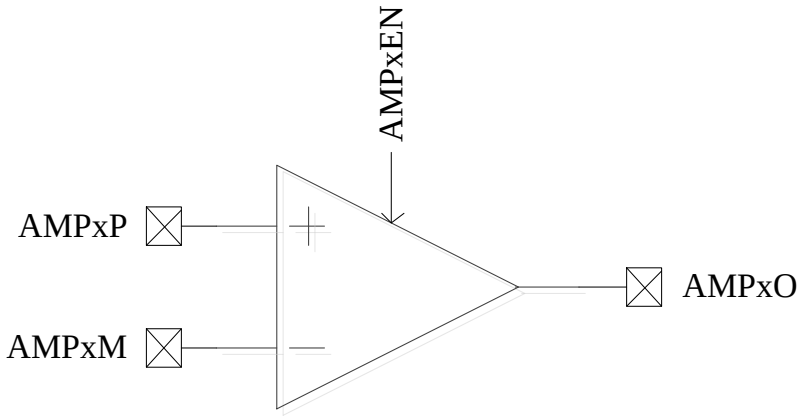


Figure 27-1 Schematic Diagram of Operational Amplifier module

27.1 Operational Amplifier Instructions

27.1.1 Bus Current Sampling Operational Amplifier (AMP0)

AMP0 operates in normal mode or PGA differential input mode.

27.1.1.1 AMP0 Normal Mode

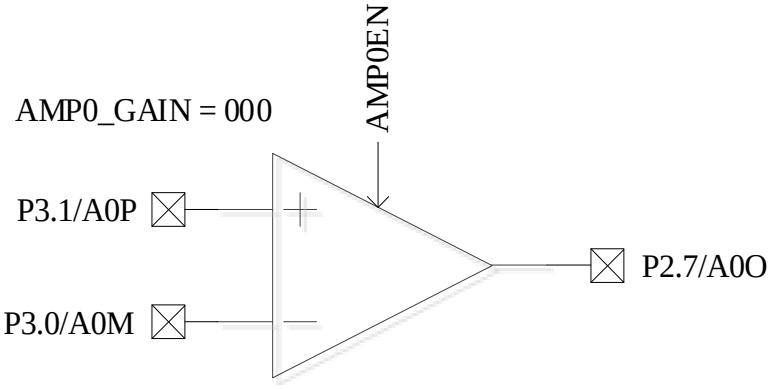


Figure 27-2 Diagram of Bus Current Operational Amplifier AMP0

The inputs and outputs of the bus current operational amplifier are shown in Figure 27-2. AMP0 is enabled when AMP_CR[AMP0EN] = 1, and P2.7, P3.0 and P3.1 are automatically configured to analog signal mode by the hardware.

27.1.1.2 AMP0 PGA Differential Input Mode

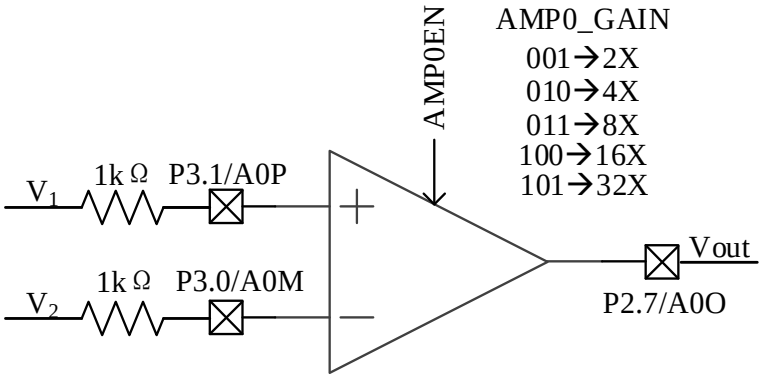


Figure 27-3 Diagram of AMP0 PGA Differential Input Mode

As shown in Figure 27-3, positive and negative inputs of AMP0 are connected with a 1kΩ resistor in the external circuit respectively.

When PGA differential Input Mode is selected for AMP0, the amplification gain is set by AMP0_GAIN and AMP0 is enabled when AMP_CR[AMP0EN] = 1. The relationship between the output and the input of operational amplifier is: Vout = VHALF + (V1 - V2)*AMP0_GAIN.

27.1.2 Phase Current Operational Amplifier (AMP1/AMP2)

27.1.2.1 AMP1

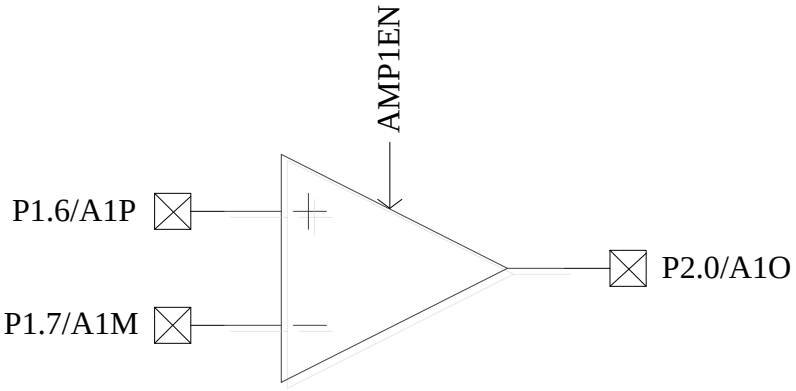


Figure 27-4 Diagram of AMP1 Input and Output

The inputs and outputs of the phase current operational amplifier are shown in Figure 27-4. The phase current operational amplifier AMP1 is enabled when AMP_CR[AMP1EN] = 1, and P1.6, P1.7, and P2.0 are automatically configured to analog signal mode by the hardware.

27.1.2.2 AMP2

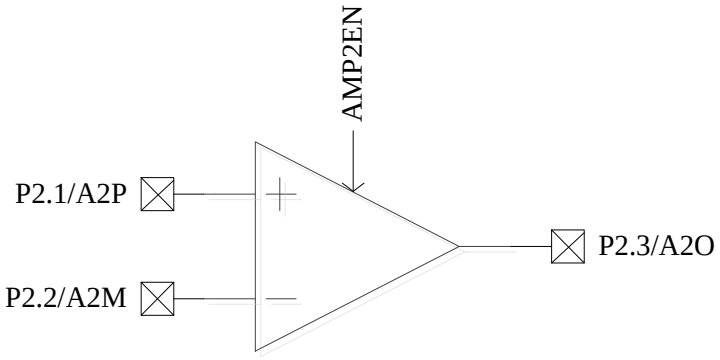


Figure 27-5 Diagram of AMP2 Input and Output

The inputs and outputs of the phase current operational amplifier are shown in Figure 27-5. The phase current operational amplifier AMP2 is enabled when CMP_CR[AMP2EN] = 1, and P2.1, P2.2, and P2.3 are automatically configured to analog signal mode by the hardware.

27.2 Operational Amplifier Registers

27.2.1 AMP_CR (0x404E)

Bit	7	6	5	4	3	2	1	0
Name	RSV					AMP2EN	AMP1EN	AMP0EN
Type	—	—	—	—	—	R/W	R/W	R/W
Reset	—	—	—	—	—	0	0	0

Bit	Name	Description
[7:3]	RSV	Reserved
[2]	AMP2EN	AMP2 Enable 0: Disable 1: Enable
[1]	AMP1EN	AMP1 Enable 0: Disable 1: Enable
[0]	AMP0EN	AMP0 Enable 0: Disable 1: Enable

27.2.2 AMP0_GAIN (0x4034)

Bit	7	6	5	4	3	2	1	0
Name	RSV					AMP0_GAIN		
Type	—	—	—	—	—	R/W	R/W	R/W
Reset	—	—	—	—	—	0	0	0

Bit	Name	Description
[7:3]	RSV	Reserved

[2:0]	AMP0_GAIN	Operational Amplifier Gain Setting 000: The gain is configured by the external circuit 001: 2x 010: 4x 011: 8x 100: 16x 101: 32x 110: Reserved 111: Reserved Note: The built-in amplification is isotropic amplification. When the difference of input voltage is 0, the output voltage is VHALF. For other applications, AMP0_GAIN is set to 000 to select external circuit to configure the gain.

28 Comparator

28.1 Comparator Operations

28.1.1 CMP3

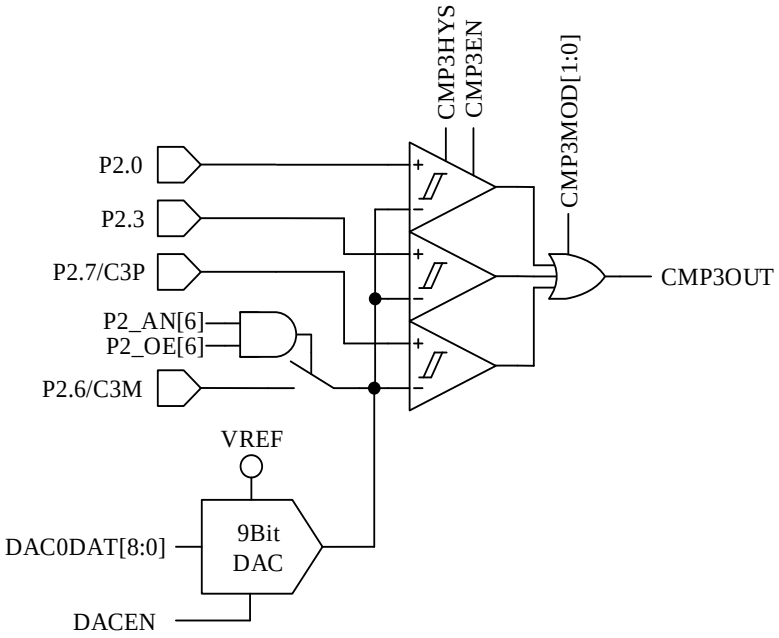


Figure 28-1 CMP3 Input and Output Pins

The input and output pins of CMP3 are shown in Figure 28-1.

The configuration of CMP3 is as follows:

1. Configure P2_AN[6] and P2_OE[6] to 1 to enable CMP3 and VREF on the negative input. The VREF source can be the on-chip DAC0 output voltage or the external circuit input voltage. Select DAC0 output, and place an external capacitance between P2.6 pin and GRND (the recommended capacitance value is 100pF, and the output voltage stabilizes after DAC0 output for a period of time);
2. Configure CMP_CR1[CMP3MOD] to select single comparator input, dual comparator input or triple comparator input mode;
   - When CMP_CR1[CMP3MOD] = 00, CMP3 works in single comparator input mode. The connection of input and output pins are shown in Figure 28-2.
   - When CMP_CR1[CMP3MOD] = 01, CMP3 works in dual comparator input mode. The connection of input and output pins are shown in Figure 28-3.
   - When CMP_CR1[CMP3MOD] = 1X, CMP3 works in triple comparator input mode. The connection of input and output pins are shown in Figure 28-4.
3. Configure CMP_CR1[CMP3HYS] to enable or disable hysteresis;
4. Set CMP_CR1[CMP3EN] = 1 to enable CMP3.

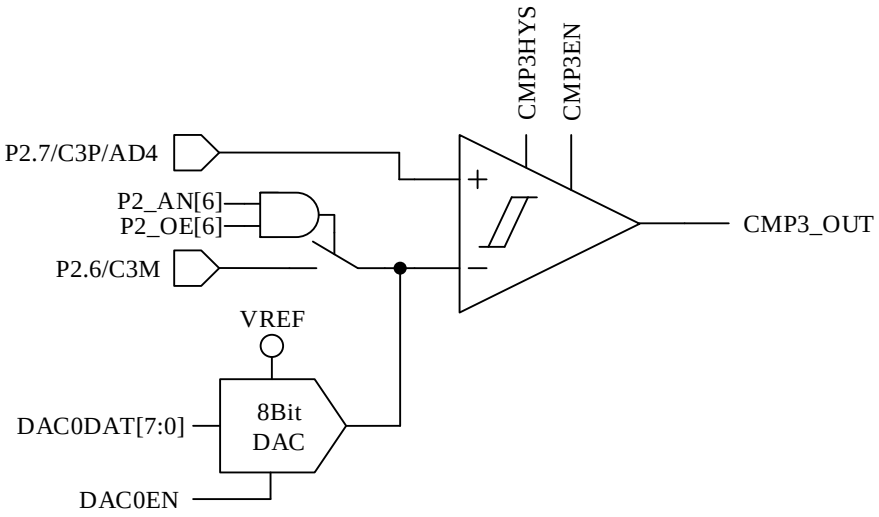


Figure 28-2 Single-comparator Input Mode

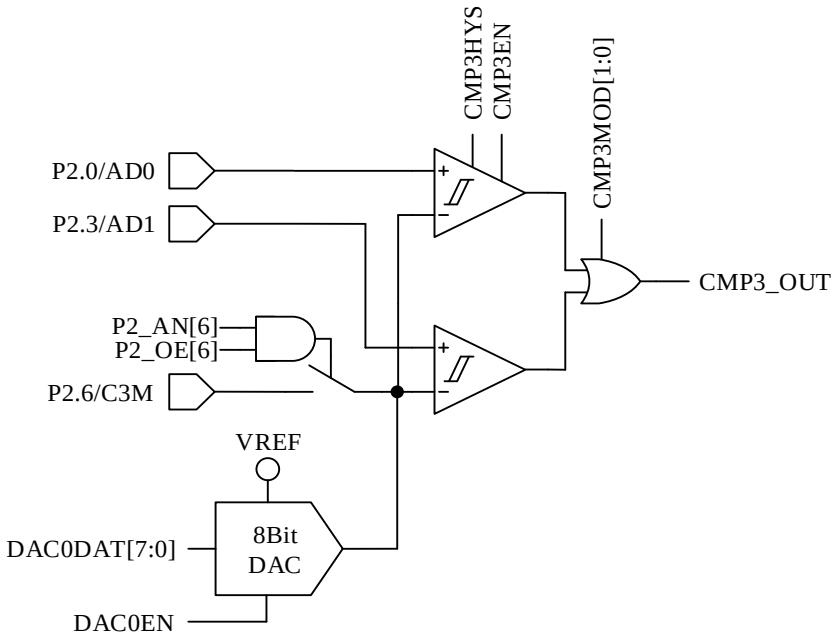


Figure 28-3 Dual-comparator Input Mode

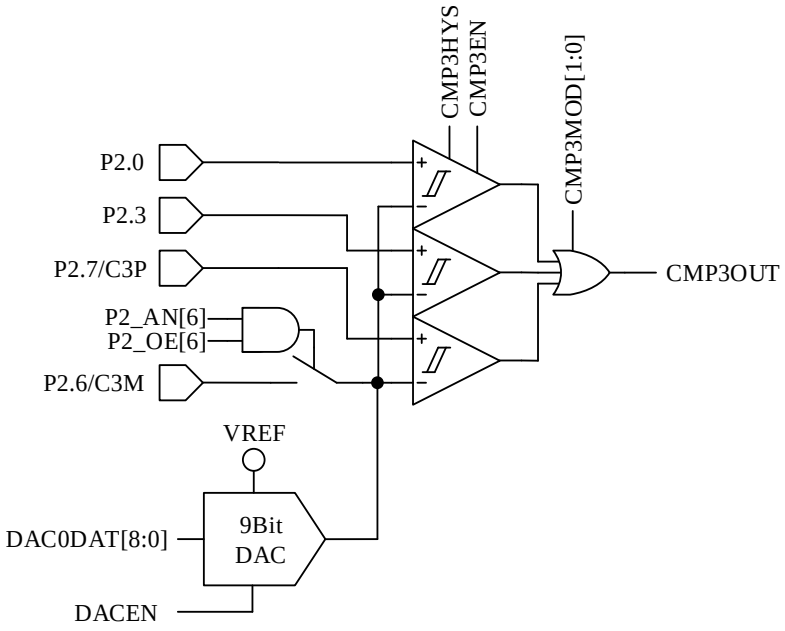


Figure 28-4 Triple-comparator Input Mode

28.1.1.1 Overcurrent Protection (OCP)

When an overcurrent protection signal is generated, DRV_OUT[MOE] is automatically cleared to output idle voltage to stop the motor drive for chip and motor protection. OCP feature is enabled when EVT_FILT[MOEMD] = 01, which automatically turns off the output and generates an OCP interrupt request if the current exceeds the threshold. When EVT_FILT[MOEMD] = 00, the output is not automatically turned off if the current exceeds the threshold. However, an OCP request is generated by the hardware.

The source of OCP interrupt is selected from CMP3 or INT0 by configuring EVT_FILT[EFSRC]. When EVT_FILT[EFSRC] =1, the TCON[IT0] bit is programmed to select the trigger edge of the external interrupt INT0 which generates an OCP output. At this time, the source of OCP interrupt is INT0. When EVT_FILT[EFSRC] = 0 and CMP_CR0[CMP3IM] = 11, the OCP output is generated on the raising edge of CMP3. At this time, the source of OCP interrupt is CMP3. In triple-shunt current sampling mode, CMP_CR1[CMP3MOD] is configured to select triple-comparator input mode. When current of any phase is over the threshold, CMP3 generates an OCP signal. For other sampling modes, CMP_CR1[CMP3MOD] is configured to choose single-comparator input mode. When bus current is over the threshold, CMP3 generates an OCP signal.

Setting EVT_FILT[EFDIV] enables the filtering of interrupt signals for OCP, and programming EVT_FILT[EFDIV] = 01/10/11 selects filter width of 4/8/16 clock cycles. When the filtering feature is enabled, the filtered signal is delayed by 4/8/16 clock cycles compared to the signal before filtering.

28.1.1.2 Cycle-by-Cycle Current Limiting

The Cycle-by-Cycle current limiting feature is applied to square wave-based drive control of BLDC motors.

When an OCP event occurs, DRV_OUT[MOE] is set to 1 after it has been cleared to “0” for a period of time, so that the motor drive automatically restored. When CMP_CR0[CMP3IM] = 11, DRV_OUT[MOE] is cleared to 0 on the rising edge of CMP3OUT to protect motor. When EVT_FILT[MOEMD] = 10, the outputs are automatically turned off upon an OCP interrupt. DRV_OUT[MOE] is enabled automatically upon Driver counter overflow/underflow events or after 10μs to restore motor drive. When EVT_FILT[MOEMD] = 11, the output is turned off upon an OCP interrupt. DRV_OUT[MOE] is automatically enabled after 5μs or upon Driver counter overflow/underflow to restore motor drive.

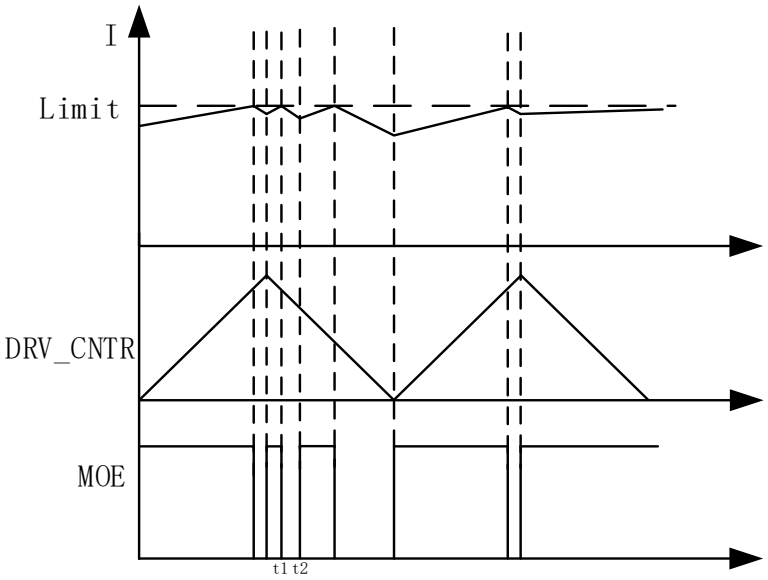


Figure 28-5 Cycle-by-Cycle Current Limiting Waveform (t2 - t1 = 10μs) when EVT_FILT[MOEMD] = 10

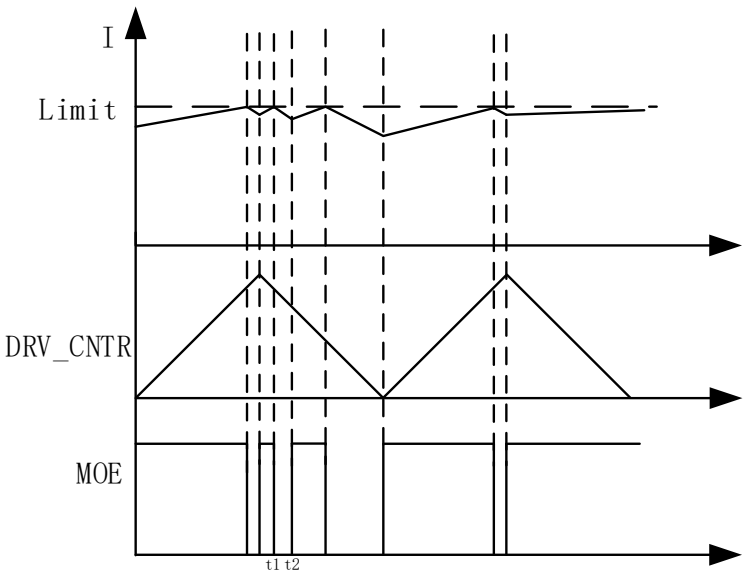


Figure 28-6 Cycle-by-Cycle Current Limiting Waveform (t2 - t1 = 5μs) when EVT_FILT[MOEMD] = 11

28.1.2 CMP4

CMP4 is a hysteresis comparator as shown in Figure 28-7. CMP4OUT can be read by software or reversed on external interrupt INT0. When CMP3 is used for Cycle-by-Cycle current limiting protection, CMP4 is used for bus current protection. Output must be turned off by software when bus current OCP feature of CMP4 is triggered.

The configuration of CMP4 is as follows:

1. Configure P2_AN[3] = 1 and P2_OE[3] = 1 to enable VREF on the positive input of CMP4. The VREF source can be the on-chip DAC1 output voltage or the external circuit input voltage. Select DAC1 output, and place an external capacitance between P2.3 pin and GRND (the recommended capacitance value is 100pF, and the output voltage stabilizes after DAC1 output for a period of time);
2. Configure P2_AN[7] = 1 to assign P2.7 pin to analog signal;
3. Configure CMP_CR2[CMP4EN] = 1 to enable CMP4;
4. Clear INT0 flag bit to enable INT0;
5. Set LVSR[EXT0CFG] = 111 to select CMP4 as the source of INT 0;
6. Configure TCON[IT0] = 01 to select falling edge triggered INT0.

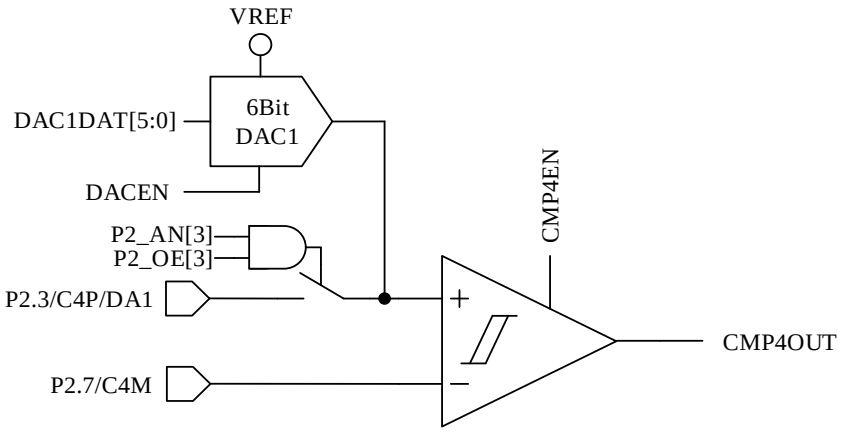


Figure 28-7 Schematic Diagram of CMP4 Module

28.1.3 Comparator Group (CMPG)

The comparator group CMPG is a collection of CMP0, CMP1 and CMP2, with multiple comparison modes for different applications.

When CMP_CR2[CMP0MOD] = 00, CMPG works in the mode of three comparators without built-in resistor. It is used for BEMF detection with the external virtual neutral point resistors. The input and output pins are shown in Figure 28-8. The negative inputs of the three comparators are connected together to P1.5, and the positive inputs are connected to P1.4, P1.6 and P2.1. The outputs are CMP0OUT, CMP1OUT, and CMP2OUT respectively. The number of comparators working in this mode is determined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0, CMP1 and CMP2 work simultaneously, which is the recommended setting. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works.

When CMP_CR2[CMP0SEL] = 11, only CMP2 works.

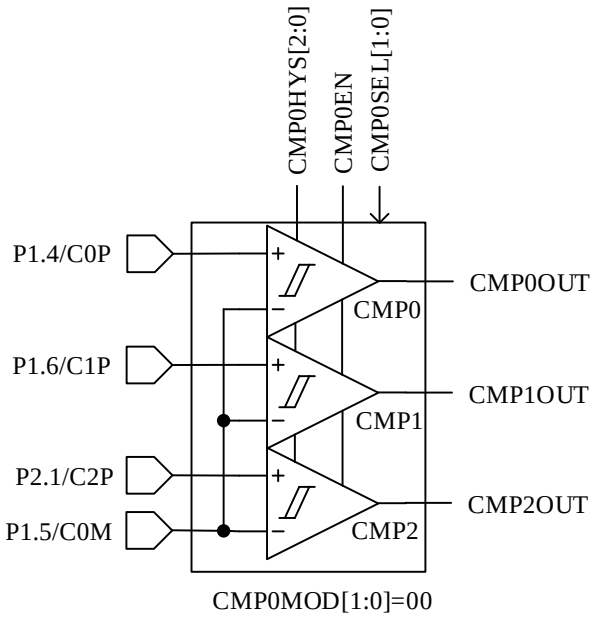


Figure 28-8 CMPG Mode with Built-in Three Comparators

(without Built-in Resistor)

When CMP_CR2[CMP0MOD] = 01, CMPG works in the mode of three comparators with built-in resistors. It is used for BEMF detection with the internal virtual neutral point resistors. The input port is selected by setting the function switching bit CMP_CR4[CMP0FS]. The number of comparators operating in this mode is determined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0, CMP1 and CMP2 comparators work simultaneously, which is the recommended setting. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works. When CMP_CR2[CMP0SEL] = 11, only CMP2 works.

When CMP_CR4[CMP0FS] = 0, the input and output pins are shown in Figure 28-9. The negative inputs of the three comparators are connected together to the neutral point of the built-in resistors. The positive inputs are connected to P1.4, P1.6 and P2.1, and the outputs are CMP0OUT, CMP1OUT, and CMP2OUT, respectively.

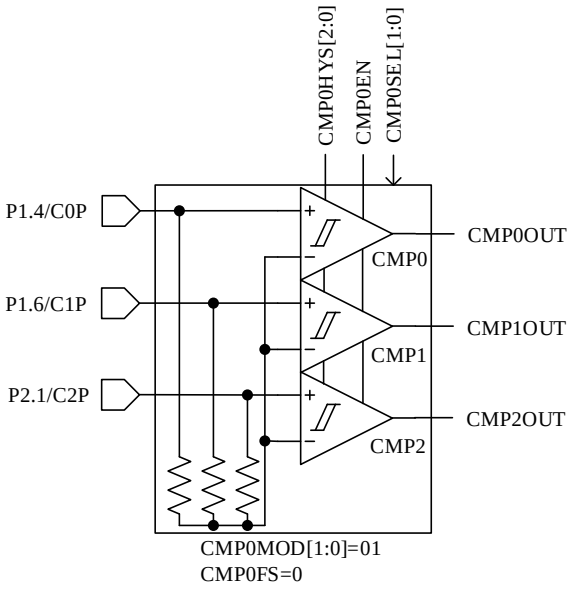


Figure 28-9 CMPG Mode with Built-in Three Comparators and Resistors

(without Function Switching)

When CMP_CR4[CMP0FS] = 1, the input and output pins are shown in Figure 28-10. The negative inputs of the three comparators are connected together to the neutral point of the built-in resistors, the positive inputs are connected to P1.4, P1.3 and P1.5 respectively, and the outputs results are CMP0OUT, CMP1OUT and CMP2OUT respectively.

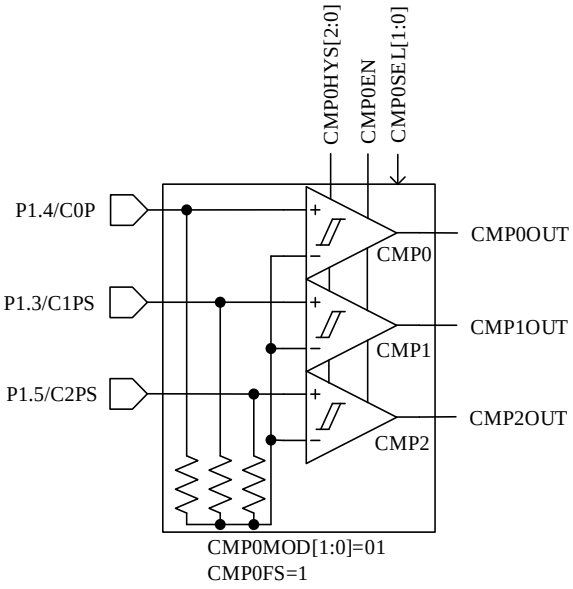


Figure 28-10 CMPG Mode with Built-in Three Comparators (with Function Switching)

When CMP_CR2[CMP0MOD] = 10, CMPG mode with three differential comparators is selected for the differential Hall sensor to detect the motor rotor position. The input and output pins are shown in Figure 28-11. The negative inputs of the three comparators are respectively connected to P1.5, P1.7 and P2.2, the positive inputs

are respectively connected to P1.4, P1.6 and P2.1, and the outputs are CMP0OUT, CMP1OUT and CMP2OUT respectively. In this mode, the number of comparators is determined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0, CMP1 and CMP2 work simultaneously, which is the recommended setting. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works. When CMP_CR2[CMP0SEL] = 11, only CMP2 works.

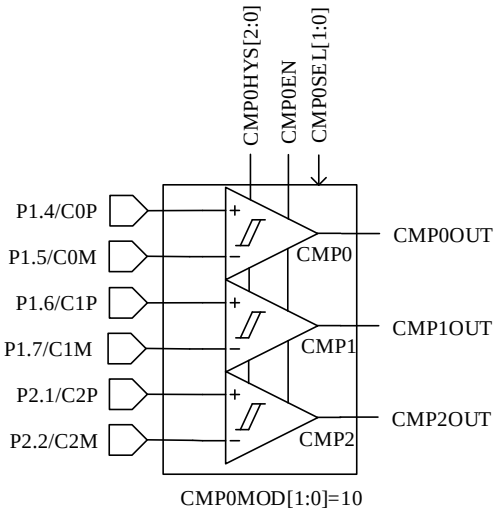


Figure 28-11 CMPG Mode with Three Differential Comparators

When CMP_CR2[CMP0MOD] = 11, CMPG mode with two comparators is selected for motor speed detection. The input and output ports are shown in Figure 28-12. The negative inputs of the comparators are connected together to P1.5, the positive inputs are connected to P1.4 and P1.3, and the outputs are CMP0OUT and CMP1OUT respectively. In this mode, the number of comparators is determined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0 and CMP1 work simultaneously, which is the recommended setting. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works.

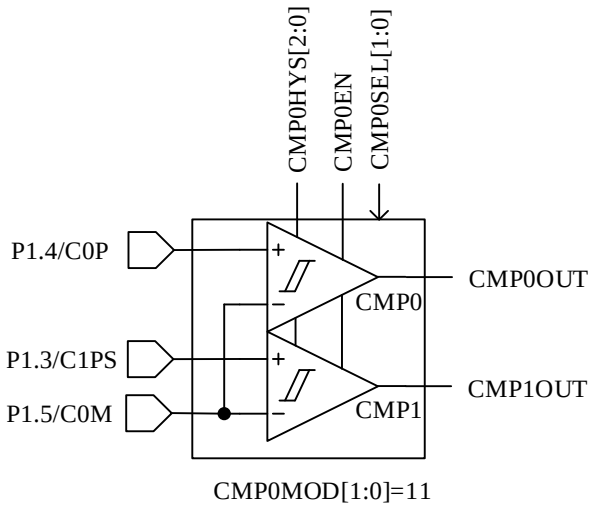


Figure 28-12 CMPG Mode with Two Comparators

The output signals of CMP0/CMP1/CMP2 are sent to Timer1 after filtering and sampling modules.

28.1.4 Comparator Sampling

The comparator sampling feature is mainly used for the square wave control and RSD, which eliminates the switching interference from driving circuit. See CMP/GPIO Position Detection Event for square wave control in 14.1.2.3 and RSD Comparator Sampling for RSD in 15.1.7.1.

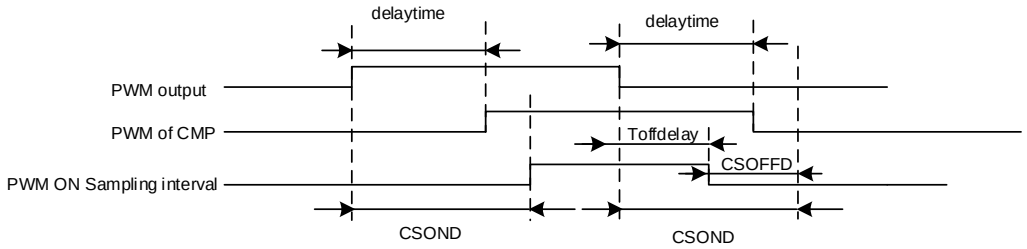


Figure 28-13 PWM ON Sampling Mode

There is a delay from PWM output to the output of the comparator, which is mainly affected by the following factors: resistance value of the drive resistor, switching speed of the power device, and input delay and hysteresis settings of the comparator. As shown in Figure 28-13, the delay time is from the chip output to the comparator output. When high-level sampling is performed, the sampling interval shall be enveloped by the actual high-level output of the comparator. First, the sampling ON-delayed time CMP_SAMR[CSOND] is set to overcome the output delay and the oscillation interval of the power device. At the end of the sampling interval, CMP_SAMR [CSOND] is delayed after the falling edge of PWM, at which time the actual sampling window has exceeded the corresponding high-level interval. The sampling OFF-advanced time CMP_SAMR[CSOFFD] is set to stop

sampling Toffdelay after the PWM output falling edge, where Toffdelay = CMP_SAMR[CSOND] – CMP_SAMR[CSOFFD]. By configuring CMP_SAMR[CSOND] and CMP_SAMR[CSOFDD], the sampling interval can be located in the high-level interval of the actual output of the comparator.

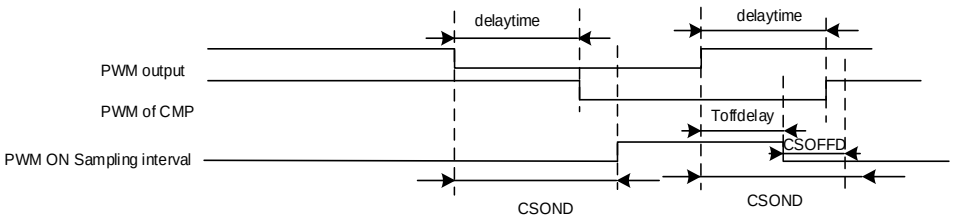


Figure 28-14 PWM OFF Sampling Mode

Similarly, when low-level sampling is performed, the sampling ON-delayed time CMP_SAMR[CSOND] and the sampling OFF-advanced time CMP_SAMR[CSOFFD] are set reasonably to ensure that the actual sampling interval is located in the actually low-level output interval of the comparator.

Method for measuring the delay of PWM output to comparator: Set CMP_CR3[SAMSEL] = 00 to disable the comparator sampling delay feature. Set CMP_CR3[CMPSEL] to select the corresponding comparator output to test pin P0.7. Enable the PWM output and comparator, manually rotate the motor to change the comparator value, and measure the delay between the PWM output and the comparator output.

28.1.5 Comparator Output

CMP_CR3[CMPSEL] is configured to output results of one comparator to P0.7.

28.2 Comparator Registers

28.2.1 CMP_CR0 (0xD5)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IM		CMP2IM		CMP1IM		CMP0IM	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	CMP3IM	CMP3 Interrupt Mode 00: No interrupt is generated. 01: An interrupt is generated upon rising edge. 10: An interrupt is generated upon falling edge. 11: When a rising edge is detected, DRV_OUT[MOE] is cleared to 0, and the interrupt event flag bit CMP_SR[CMP3IF] is set to 1. However, the interrupt is not enabled. (Note: In Cycle-by-Cycle Current Limiting Mode, EVT_FILT[MOEMD] must be set to "10/11".)
[5:4]	CMP2IM	CMP2 Interrupt Mode See descriptions on CMP_CR0[CMP0IM].
[3:2]	CMP1IM	CMP1 Interrupt Mode See descriptions on CMP_CR0[CMP0IM].
[1:0]	CMP0IM	CMP0 Interrupt Mode 00: No interrupt generates 01: Interrupt on rising edge 10: Interrupt on falling edge 11: Interrupt on both rising/falling edges

28.2.2 CMP_CR1 (0xD6)

Bit	7	6	5	4	3	2	1	0
Name	HALLSEL	CMP3MOD		CMP3EN	CMP3HYS	CMP0HYS		
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	HALLSEL	Hall Signal Functional Switching 0: Hall signals are not transferred. P0.2, P3.7 and P3.6 serve as the inputs for Hall signal. 1: Hall signals are transferred. P1.4, P1.6 and P2.1 serve as the inputs for Hall signal.
[6:5]	CMP3MOD	CMP3 Mode Selection Negative input connected to P2.6 or DAC0 output 00: Single-comparator mode; P2.7 is connected to the positive input. See Figure 28-2. 01: Dual-comparator mode; P2.0 and P2.3 are connected to the positive inputs. See Figure 28-3. 1X: Triple-comparator mode; P2.0, P2.3 and P2.7 are connected to the positive inputs. See Figure 28-4.
[4]	CMP3EN	CMP3 Enable 0: Disable 1: Enable
[3]	CMP3HYS	CMP3 Hysteresis Selection 0: No hysteresis 1: Hysteresis voltage is 15mV
[2:0]	CMP0HYS	CMP0/1/2 Hysteresis Voltage Selection: 000: No hysteresis 001: ±2.5mV

		010: –5mV 100: +5mV 011: ±5mV 101: –10mV 110: +10mV 111: ±10mV

28.2.3 CMP_CR2 (0xDA)

Bit	7	6	5	4	3	2	1	0
Name	CMP4EN	CMP0MOD		CMP0SEL		RSV		CMP0EN
Type	R/W	R/W	R/W	R/W	R/W	—	—	R/W
Reset	0	0	0	0	0	—	—	0

Bit	Name	Description
[7]	CMP4EN	CMP4 Enable 0: Disable 1: Enable
[6:5]	CMP0MOD	CMPG Mode Selection 00: CMPG mode with built-in three comparators (without built-in resistor), as shown in Figure 28-8 01: CMPG mode with built-in three comparators and resistors, where function switching is configured by CMP_CR4[CMP0FS]. See Figure 28-9 and Figure 28-10. 10: CMPG mode with three differential comparators, as shown in Figure 28-11. 11: CMPG mode with two comparators, where only CMP0/CMP1 work. See Figure 28-12.
[4:3]	CMP0SEL	Pin Combination Selection of CMPG. This bit works with CMP_CR2[CMP0MOD]. The default value is 00. Under square wave control application, TIM1_DBRx[T1CPE] automatically controls CMP_CR2[CMP0SEL] to enable or disable each comparator.

Table 28-1 Function Description of CMPG Port and CMP_CR2[CMP0MOD] Combination

CMP_CR2 [CMP0MOD]	CMP_CR2 [CMP0SEL]	Description
00	00	CMP0/1/2 work simultaneously, refer to Figure 28-8. The negative inputs of 3 comparators are connected to C0M. The hardware compares the positive inputs C0P, C1P and C2P with C0M, and the output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.
	01	Only CMP0 works. The positive input is connected to C0P, and the negative input to C0M The output results are transferred to CMP0OUT.
	10	Only CMP1 works. The positive input is connected to C1P, and the negative input to C0M. The output results are transferred to CMP0OUT.
	11	Only CMP2 works. The positive input is connected to C2P and the negative input to C0M. The output results are transferred to CMP2OUT.
01	00	CMP0/1/2 work simultaneously, refer to Figure 28-9 and Figure 28-10. The negative inputs of the 3 comparators are connected to the neutral point of built-in resistors. When CMP0FS = 0, the hardware automatically compares the positive inputs C0P, C1P, and C2P with C0M, and when CMP0FS = 1, it automatically

					compares the positive inputs C0P, C1PS and C2PS with C0M. The output results are transferred to CMP0OUT, CMP1OUT, and CMP2OUT respectively.	
				01	Only CMP0 works. The positive input is connected to C0P, and the negative input to C0M. The output results are transferred to CMP0OUT.	
				10	Only CMP1 works. When CMP0FS = 0, the positive input is connected to C1P, and when CMP0FS = 1, it is connected to C1PS. The negative input is connected to the neutral point of the built-in BEMF resistors, and the output results are transferred to CMP1OUT.	
				11	Only CMP2 works. When CMP0FS = 0, the positive input is connected to C2P, and when CMP0FS = 1, it is connected to C2PS. The negative input is connected to the neutral point of the built-in BEMF resistors, and the output results are transferred to CMP2OUT.	
			10	00	CMP0/1/2 work simultaneously. Refer to Figure 28-11. The positive inputs of the three comparators are connected to C0P, C1P, and C2P respectively, and the negative inputs to C0M, C1M, and C2M respectively. The output results are transferred to CMP0OUT, CMP1OUT, and CMP2OUT, respectively.	
				01	Only CMP0 works. The positive input is connected to C0P, and the negative input to C0M. The output results are transferred to CMP0OUT.	
				10	Only CMP1 works. The positive input is connected to C1P, and the negative input to C1M. The output results are transferred to CMP01UT.	
				11	Only CMP2 work. The positive input is connected to C2P, and the negative input to C2M. The output results are transferred to CMP2OUT.	
			11	00	CMP0/1 work simultaneously, refer to Figure 28-12. The positive inputs are connected to C0P and C1PS respectively, and the negative inputs to C0M. The output results are transferred to CMP0OUT and CMP1OUT respectively.	
				01	Only CMP0 works. The positive input is connected to C0P, and the negative input to C0M. The output results are transferred to CMP0OUT.	
				10	Only CMP1 works. The positive input is connected to C1PS, and the negative input to C0M. The output results are transferred to CMP1OUT.	
				11	Reserved	
[2:1]	RSV	Reserved				
[0]	CMP0EN	CMP0 enable 0: Disable 1: Enable				

28.2.4 CMP_CR3 (0xDC)

Bit	7	6	5	4	3	2	1	0
Name	RSV	DBGSEL		SAMSEL		CMPSEL		
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6:5]	DBGSEL	Debug Signal Selection, output to P0.1 00: Disable debug signal output 01: End of diode freewheeling and detection of ZCP 10: ADC trigger signal 11: Comparator sampling interval
[4:3]	SAMSEL	Sampling delay enable of CMP0, CMP1, CMP2 and ADC in PWM ON/OFF modes 00: Sampling at both PWM ON and OFF modes without time delay 01: Sampling at PWM OFF mode, with time delay according to CMP_SAMR 10: Sampling at PWM ON, with time delay according to CMP_SAMR 11: Sampling at both PWM ON and OFF, with time delay according to CMP_SAMR
[2:0]	CMPSEL	Comparator Output Selection Output signals of one selected comparator to P0.7. 000: Disable 001: CMP0 010: CMP1 011: CMP2 100: CMP3 101: CMP4 111: omega start flag (See section 13.1.9.3 for estimator estimated angle output flag bit)

28.2.5 CMP_CR4 (0xE1)

Bit	7	6	5	4	3	2	1	0
Name	CMP4OUT	RSV				FAEN	CMP0FS	RSV
Type	R	—	—	—	—	R/W	R/W	—
Reset	1	—	—	—	—	0	0	—

Bit	Name	Description
[7]	CMP4OUT	CMP4 Comparison Results
[6:3]	RSV	Reserved
[2]	FAEN	Filter Sampling Coefficient Expansion Enable With it enabled, the clock source of TIM1_CR3[T1INM] and CMP_SAMR is increased by 4 times 0: Disable 1: Enable
[1]	CMP0FS	CMP1, CMP2 Functional Switching Enable 0: No function switching. Refer to Figure 28-9 1: CMP1/CMP2 are switched. This bit is valid only when CMP_CR2[CMP0_MOD] = 01. See Figure 28-10.
[0]	RSV	Reserved

28.2.6 CMP_SAMR (0x40AD)

Bit	7	6	5	4	3	2	1	0
Name	CSOND				CSOFFD			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	Name	Description
[7:4]	CSOND	CMP0/CMP1/CMP2 ON-delayed Sampling Time When PWM Module switches from OFF to ON or from ON to OFF, turn-on/off of the power device affects signal input of the comparator. In this case, CMP_SAMR[CSOND] is configured to delay the sampling of CMP0/CMP1/CMP2. The delay time can be multiplied by 4 times by setting CMP_CR4[FAEN]. CMP_CR4[FAEN] = 0: ON-delayed sampling time = 8*CMP_SAMR[CSOND]*T CMP_CR4[FAEN] = 1: ON-delayed sampling time = 32*CMP_SAMR[CSOND]*T Notes: ■ CMP_SAMR[CSOND] must be larger than or equal to CMP_SAMR[CSOFFD] ■ For BLDC control, see Sampling. ■ For RSD, see RSD Comparator Sampling
[3:0]	CSOFFD	CMP0, CMP1, CMP2 OFF-lead Sampling Time CMP_SAMR[CSOND] is configured to end the sampling CMP_SAMR[CSOND] – CMP_SAMR[CSOFFD] after the back edge of PWM output to ensure sampling interval enveloped by the PWM interval. Off-lead sampling time OFF-lead sampling time can be multiplied by 4 times by setting CMP_CR4[FAEN]. CMP_CR4[FAEN] = 0: Off-lead sampling time = 8*CMP_SAMR[CSOFFD]*T CMP_CR4[FAEN[= 1: Off-lead sampling time = 32*CMP_SAMR[CSOFFD]*T Notes: ■ CMP_SAMR[CSOND] must be larger than or equal to CMP_SAMR[CSOFFD] ■ For BLDC control, see Sampling. ■ For RSD, see RSD Comparator Sampling

28.2.7 CMP_SR (0xD7)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IF	CMP2IF	CMP1IF	CMP0IF	CMP3OUT	CMP2OUT	CMP1OUT	CMP0OUT
Type	R/W0	R/W0	R/W0	R/W0	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	CMP3IF	CMP3 Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	CMP2IF	CMP2 Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect
[5]	CMP1IF	CMP1 Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to “0” 1: No effect

[4]	CMP0IF	CMP0 Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to "0" 1: No effect
[3]	CMP3OUT	CMP3 comparison result
[2]	CMP2OUT	CMP2 comparison result
[1]	CMP1OUT	CMP1 comparison result
[0]	CMP0OUT	CMP0 comparison result

28.2.8 EVT_FILT (0xD9)

Bit	7	6	5	4	3	2	1	0
Name	RSV			MOEMD		EFSRC	EFDIV	
Type	—	—	—	R/W	R/W	R/W	R/W	R/W
Reset	—	—	—	0	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4:3]	MOEMD	Hardware Clears and Enables MOE MOE is cleared and enabled by hardware upon over-/under-current protection event. 00: MOE is not automatically cleared 01: MOE is automatically cleared 10: MOE is automatically cleared and enabled after 10μs or at an overflow or underflow event of the Driver counter (for square-wave control). 11: MOE is automatically cleared and enabled after 5μs or at an overflow or underflow event of the Driver counter (for square-wave control).
[2]	EFSRC	Input Source of Current Protection Event 0: CMP3 interrupt 1: External interrupt INT0
[1:0]	EFDIV	Filtering Width for Current Protection 00: No filtering 01: 4 system clock cycles 10: 8 system clock cycles 11: 16 system clock cycles

28.2.9 TSD_CR (0x402F)

Bit	7	6	5	4	3	2	1	0
Name	TSDEN	RSV			TSDADJ			
Type	R/W	—	—	—	R/W	R/W	R/W	R/W
Reset	0	—	—	—	1	1	1	1

Bit	Name	Description
[7]	TSDEN	Temperature Detection Feature Enable 0: Disable 1: Enable
[6:4]	RSV	Reserved
[3:0]	TSDADJ	Overtemperature Value (Chip Junction Temperature) 0000: 71°C 0001: 75°C 0010: 80°C 0011: 84°C

		0100: 89°C 0101: 94°C 0110: 99°C 0111: 105°C 1000: 111°C 1001: 116°C 1010: 123°C 1011: 131°C 1100: 136°C 1101: 142°C 1110: 150°C 1111: Reserved

29 Power Supply

29.1 LDO

The chip contains two internal LDO output modules: VDD5 and VDD18.

29.1.1 LDO Operations

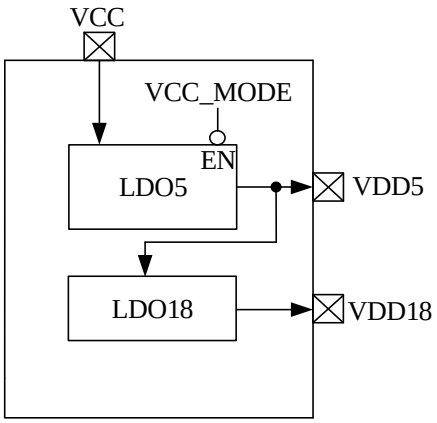


Figure 29-1 Functional Block Diagram of Power Supply

The corresponding input and output ports of the LDO module are shown in Figure 29-1. The LDO module converts the input supply voltage to 5V (VDD5) and 1.8V (VDD18) as the power supply for internal analog and digital modules of the chip, respectively. Internal LDO5 or external supply for VDD5 is selected by configuring VCC_MODE. As shown in Figure 29-2, in the debugging tool, VCC_MODE = 0 if Vcc Mode is unchecked, where internal LDO supplies VDD5 voltage. VCC_MODE = 1 if Vcc Mode is checked, where external 5V power supply is connected to VDD5 pin.

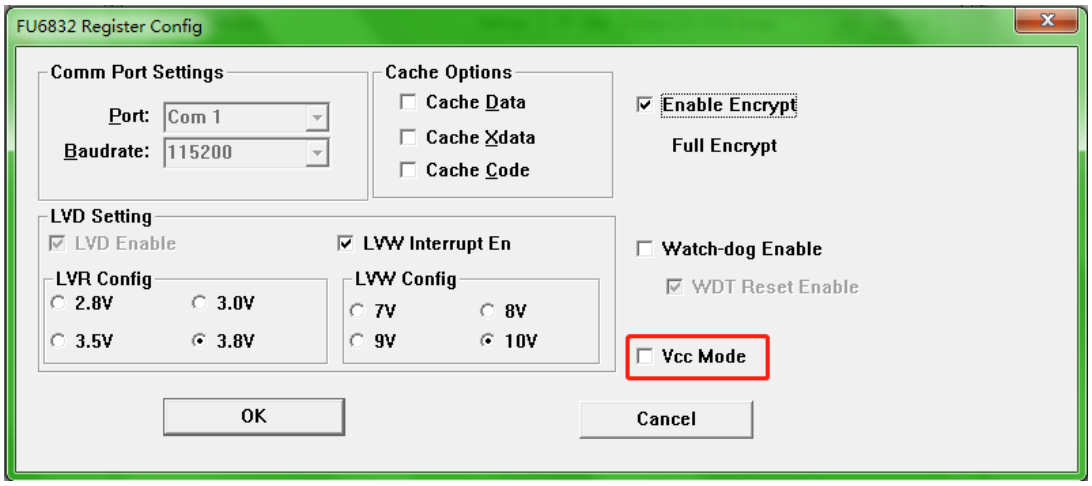


Figure 29-2 VCC_MODE Configuration

29.2 Low Voltage Detector (LVD)

29.2.1 LVD Introduction

The low voltage detector has two main features: low voltage warning and low voltage reset.

29.2.2 LVD Operations

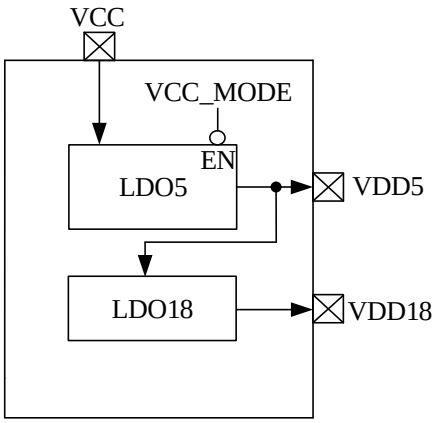


Figure 29-3 LVD Module

The operating instructions for LVD are as follows:

- LV warning and LV reset are always enabled by default.
- 7V/8V/9V/10V can be selected for LV warning threshold. When the interrupt feature is enabled, an interrupt is triggered if VCC voltage is lower than the LV warning threshold.
- 2.8V/3.0V/3.5V/3.8V can be selected for the LV warning threshold. The chip resets when VCC voltage settles below the LV reset voltage threshold.

The LV warning threshold, interrupt configuration and low voltage reset threshold are configured through debugging tool, as shown in Figure 29-4.

LVR Config sets low voltage reset threshold, LVW Interrupt En enables the LV interrupt, and LVW Config sets the low voltage warning threshold.

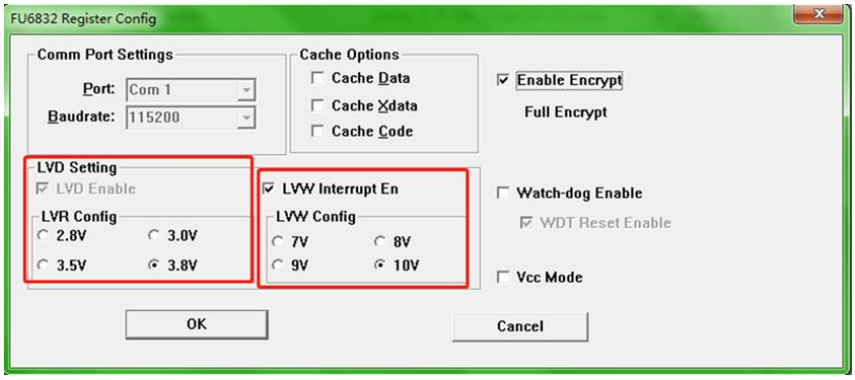


Figure 29-4 Configuration of LV Warning Threshold, LV Interrupt and LV Reset Threshold

29.2.3 LVD Registers

29.2.3.1 LVSR (0xDB)

Bit	7	6	5	4	3	2	1	0
Name	RSV		EXT0CFG			TSDF	LVWF	LVWIF
Type	—	—	R/W	R/W	R/W	R	R	R/W0
Reset	—	—	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:3]	EXT0CFG	INT0 Pin Selection 000: P0.0 001: P0.1 010: P0.2 011: P0.3 100: P0.4 101: P0.5 110: P0.6 111: CMP4 output
[2]	TSDF	Over-temperature State Indicator 0: The current temperature does not exceed the threshold 1: The current temperature exceeds the threshold Note: This flag bit often works with TSD interrupt flag TCON[5]
[1]	LVWF	VCC Low Voltage Flag This bit indicates whether the chip is in the low voltage state 0: There is no low voltage warning 1: There is a low voltage warning
[0]	LVWIF	VCC Low Voltage Interrupt Flag Read: 0: No interrupt pending 1: Interrupt pending Write: 0: This bit is cleared to 0 1: No effect Note: This bit is not be set to 1 when the LVD interrupt is disabled.

30 Flash

30.1 Flash Introduction

The chip provides 16k bytes of on-chip Flash space. It supports chip erase/write and sector erase/write.

Features:

- There are 128 sectors in total, each with a size of 128 bytes.
- The last sector (address range: 0x3F80 to 0x3FFF) cannot be erased at any moment.
- Sector erase and chip erase takes about 120ms ~ 150ms.

30.2 Flash Operations

- All interrupts must be disabled before self-programming to ensure the security of Flash operations and avoid mis-operation of Flash using MOVX instruction during interrupt processing.
- Flash memory shall be unlocked before operation. The Flash software programming feature is activated after 0x5A and 0x1F are written to register FLA_KEY in sequence. If the sequence is incorrect or other values are written, Flash space is frozen until the next reset. After unlocking, any write to FLA_CR causes FLA_KEY to be locked again.
- CRC results change if Flash memory is rewritten during program execution.

30.3 Flash Registers

30.3.1 FLA_CR(0x85)

Bit	7	6	5	4	3	2	1	0
Name	RSV			FLAERR	RSV	FLAPRE	FLAERS	FLAEN
Type	—	—	—	R	—	R/W	R/W	R/W
Reset	—	—	—	0	—	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	FLAERR	Programming Error Flag 0: Programming or pre-programming succeeds when writing to Flash. 1: Programming or pre-programming fails when writing to Flash.
[3]	RSV	Reserved
[2]	FLAPRE	Sector Pre-programming Enable (The sector must be pre-programmed before erasing) 0: Disable 1: Enable Note: FLA_CR[FLAPRE] works only when FLA_CR[FLAEN] is set to 1.
[1]	FLAERS	Erase Enable 0: Disable 1: Enable Note: FLA_CR[FLAERS] works only when FLA_CR[FLAEN] is set to 1.
[0]	FLAEN	Flash Operation Enable 0: Disable 1: Enable

30.3.2 FLA_KEY (0x84)

Bit	7	6	5	4	3	2	1	0
Name	FLA_KEY							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FLA_KEY	Write 0x5A and 0x1F to FLA_KEY in sequence to unlock Flash operation; Write any value to FLA_CR to lock Flash operation.

Bit	7	6	5	4	3	2	1	0
Name	RSV						FLAKSTA	
Type	—	—	—	—	—	—	R	R
Reset	—	—	—	—	—	—	0	0

Bit	Name	Description
[7:2]	RSV	Reserved
[1:0]	FLAKSTA	Read: Flash Unlock Status 00: Locked 01: Write of 0x5A is done, waiting for write of 0x1F 10: Frozen 11: Unlocked

31 CRC

31.1 CRC Functional Block Diagram

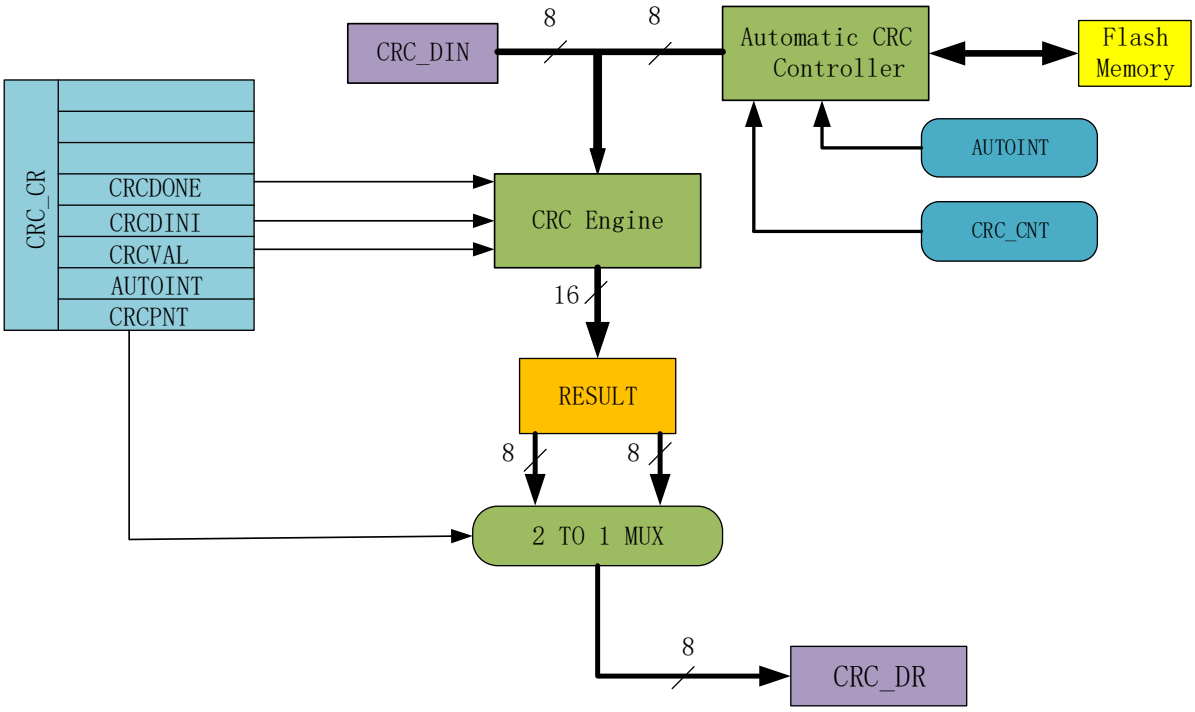


Figure 31-1 CRC Functional Block Diagram

CRC module outputs the result of CRC calculation for any 8-bit data based on a fixed polynomial. As shown in Figure 31-1, CRC receives the 8-bit data from CRC_DIN and sends the 16-bit result to the internal register after the calculation is completed. The result can be indirectly accessed through CRC_CR[CRCPNT] and CRC_DR.

Table 31-1 CRC Criteria and Polynomials

No.	CRC Criteria	Polynomial	Hexadecimal Representation
1	CRC12	x^12+x^11+x^3+x^2+x+1	0x80F
2	CRC16	x^16+x^15+x^2+1	0x8005
3	CRC16/ CCITT-FALSE	x^16+x^12+x^5+1	0x1021
4	CRC32	x^32+x^26+x^23+x^22+x^16+x^12 +x^11+x^10+x^8+x^9+x^5+x^4+x+1	0x04C11DB7

31.2 CRC16 Polynomial

The chip uses CRC16/CCITT-FALSE polynomial: x^16+x^12+x^5+1.

31.3 CRC16 Logic Diagram

Figure 31-2 presents the schematics of CRC16. The chip implementation is based on parallel algorithm. For

each input byte, MCU calculates the results within 1 system clock cycle.

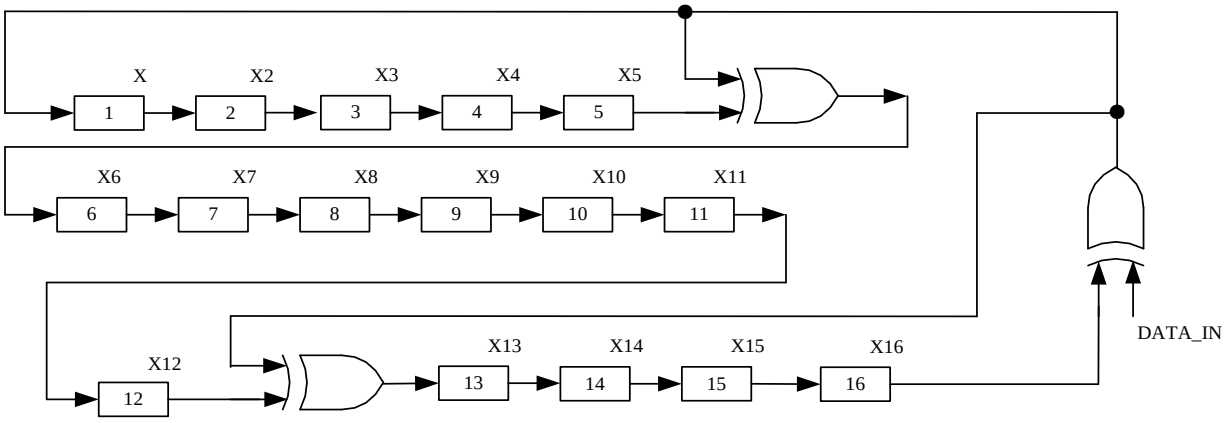


Figure 31-2 CRC16 Schematic Diagram

31.4 CRC Operations

31.4.1 CRC Calculation of a Single Byte

CRC of a single byte is calculated as follows:

1. Initialize CRC_DR with two options: Configure CRC_CR[CRCVAL] and set CRC_CR[CRCDINI] to 1, with an initial value of 0x0000 or 0xFFFF. Or configure CRC_CR[CRCPNT] and CRC_DR, where any initial value can be set;
2. Write data to CRC_DIN, and CRC calculation is completed in the next system clock cycle;
3. Read CRC value: Configure CRC_CR[CRCPNT] = 1, and read off CRC_DR in software to get the high-order bytes. Configure CRC_CR[CRCPNT] = 0, and read off CRC_DR to get the low-order bytes.

31.4.2 CRC Calculation of ROM Sector

CRC of a continuous area of data in the ROM is calculated as follows:

1. Initialize CRC_DR in the same way as that of single-byte CRC calculation;
2. Configure CRC_BEG to define starting sector of the ROM to be calculated;
3. Configure CRC_CNT to set the offset from the starting sector to the ending sector;
4. Write 1 to CRC_CR[AUTOINT] and keep other bits unchanged. The calculation starts automatically;
5. Read the CRC results.

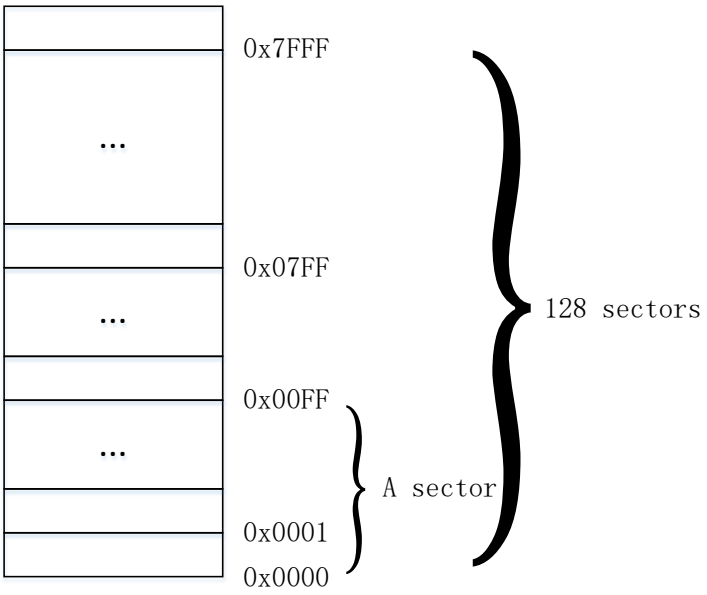


Figure 31-3 ROM Sectors

As shown in Figure 31-3, ROM has 16k bytes and is divided into 128 sectors numbered from sector0 to sector127. Each sector contains 128 bytes. For CRC calculation of sectors, the value of CRC_BEG (the starting sector) can be any value between 0x00 ~ 0x7F, including 0x00 and 0x7F. the value of CRC_CNT (the total number of sectors to be calculated) can be any value between 0x00 ~ 0x7F, including 0x00 and 0x7F.

As CRC_BEG increases, CRC_CNT decreases accordingly. For example, if CRC_BEG is 0x7F, CRC_CNT can be 0x00 only, i.e. the CRC value of the data in the last sector is calculated. In this case, if the value of CRC_CNT is large, CRC controller automatically limits the number of sectors to be calculated. Finally, CRC module only calculates CRC value of the last sector.

31.5 CRC Registers

31.5.1 CRC_CR (0x4022)

Bit	7	6	5	4	3	2	1	0
Name	RSV			CRCDONE	CRCDINI	CRCVAL	AUTOINT	CRCPNT
Type	—	—	—	R	W1	R/W	W1	R/W
Reset	—	—	—	1	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	CRCDONE	Automatic CRC Calculation Completed Flag During the calculation, this bit is automatically set to 0 by the hardware, and the software program stops. In other cases, this bit is automatically set to 1 by the hardware, so the software always returns 1 when reading this bit.
[3]	CRCDINI	CRC Result Initialization Trigger 0: No effect 1: CRC result initialization is triggered.
[2]	CRCVAL	CRC Result Initialization Selection 0: CRC result is initialized to 0x0000 1: CRC result is initialized to 0xFFFF

[1]	AUTOINT	Sector CRC Auto-calculation Start 0: No effect 1: Sector CRC auto-calculation starts See CRC Calculation of ROM Sector.
[0]	CRCPNT	CRC Result Pointer 0: Read CRC_DR to accesses 8 low-order bits of the 16-bit CRC result 1: Read CRC_DR to accesses 8 high-order bits of the 16-bit CRC result

Note: CRC_CR[AUTOINT] is se to 0 to perform single-byte CRC checksum.

31.5.2 CRC_DIN (0x4021)

Bit	7	6	5	4	3	2	1	0
Name	CRC_DIN							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	CRC_DIN	CRC Input Data Each time a data frame is written to this register, CRC module automatically calculates a new CRC result based on the existing CRC result and overwrites the original one. Note: This register is a virtual register, so the written data is not saved. 0x00 is returned when the address is accessed.

31.5.3 CRC_DR (0x4023)

Bit	7	6	5	4	3	2	1	0
Name	CRC_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	CRC_DR	CRC Result Output Each time this register is read or written, CRC module determines to access 8 high-order bits or 8 lower-order bits of the CRC result according to CRC_CR[CRCPNT].

31.5.4 CRC_BEG (0x4024)

Bit	7	6	5	4	3	2	1	0
Name	RSV	CRC_BEG						
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6:0]	CRC_BEG	First ROM Sector Pending Auto CRC Calculation For example, if CRC_BEG is set to 1, CRC calculation starts from the location 1*128 = 128, or rather from the first byte of sector 2.

31.5.5 CRC_CNT (0x4025)

Bit	7	6	5	4	3	2	1	0
Name	RSV	CRC_CNT						
Type	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	—	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6:0]	CRC_CNT	Offset of Sector Pending Automatic CRC Calculation This bit defines the offset of ROM sector for CRC calculation and determines the last sector pending CRC calculation.

32 Sleep Mode

32.1 Introduction

The chip operates in three modes: Normal, Standby and Sleep. These modes are selected by setting PCON[IDLE] and PCON[STOP].

The operating states of the module under different power modes are summarized in Table 32-1.

Table 32-1 Power Modes

Power Mode	Description	Wake-up Source	Power Consumption Performance
Normal	All modules work at full speed except for peripherals that are disabled	NA	High power consumption with best performance
Standby	CPU clock stops and the other functional modules are enable or disabled, depending on their control bit setting. Watchdog Timer stops.	Any interrupt, Reset/Debug on external interrupt	Low power consumption with flexible performance
Sleep	Flash Deep Sleep. The analog fast clock circuit is disconnected and software shall be operated to check if ADC, FOC and driver modules are disabled before the chip enters the Sleep mode. Watchdog Timer is disabled.	External interrupt, RTC interrupt, Reset/Debug on external interrupt	Extremely low power consumption with flexible performance

Note: It is recommended to insert 3 null statements in the sleep mode.

```
PCON = 0x02;
_nop_();
_nop_();
_nop_();
```

32.2 Sleep Mode Register

32.2.1 PCON(0x87)

Bit	7	6	5	4	3	2	1	0
Name	RSV		GF3	GF2	GF1	RSV	STOP	IDLE
Type	—	—	R/W	R/W	R/W	—	R/W	R/W
Reset	—	—	0	0	0	—	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	GF3	General Flag Bit 3
[4]	GF2	General Flag Bit 2
[3]	GF1	General Flag Bit 1
[2]	RSV	Reserved
[1]	STOP	A write of "1" makes the chip enter the Sleep mode. This bit is automatically cleared to "0" by the hardware after wakeup.
[0]	IDLE	A write of "1" makes the chip enter the Standby mode. This bit is automatically cleared to "0" by the hardware after wakeup.

Power consumption mode PCON[STOP:IDLE]:

00: Normal

01: Standby

1X: Sleep

33 Code Protection

33.1 Introduction

The chip supports full Flash space encryption to protect your software intellectual property and avoid unauthorized access. When Flash memory is encrypted, the data inside cannot be read, and data consistency can be evaluated by CRC check module only.

33.2 Operating Instructions

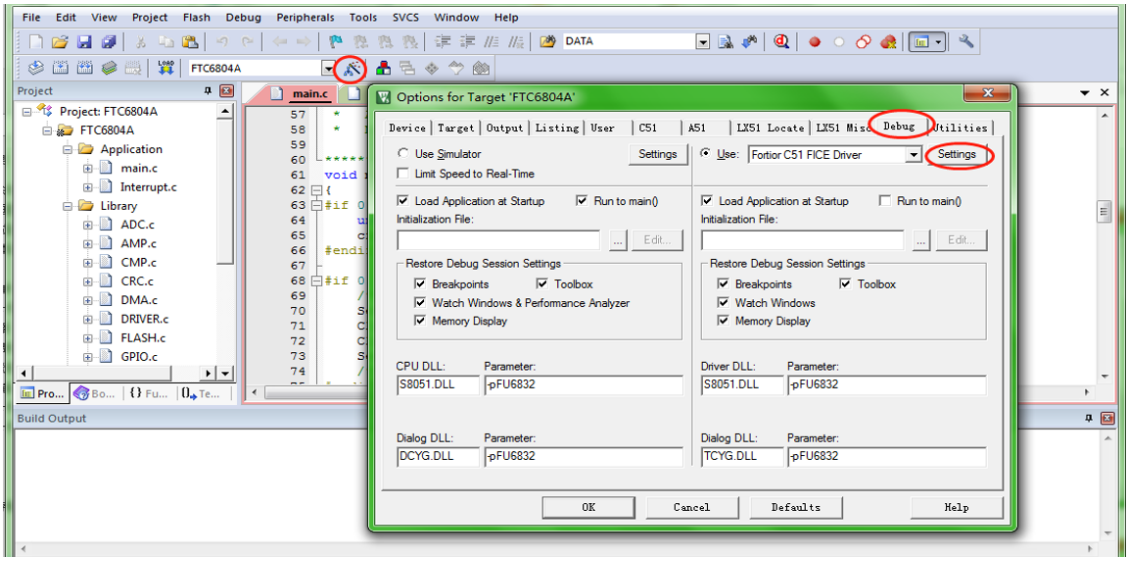


Figure 33-1 Code Protection Configurations

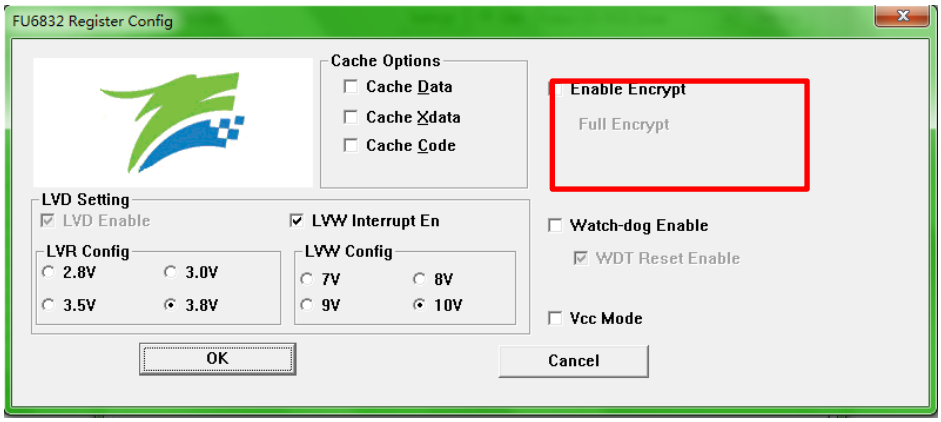


Figure 33-2 Full Code Protection Mode

Operation steps are as follows:

1. Start 8051 IDE, enter Target Options and select Debug tab. As shown in Figure 33-1, click Settings to proceed with the setting;
2. Select the options as shown in Figure 33-2, and click OK. Then compile the project and download it. Get the BIN file and program it to Flash.

34 Revision History

Rev.	Description	Date	Prepared By
V1.2	First release, translated from Chinese version 1.2.	2023/05/17	Eric Deng
V1.3	1. Deleted descriptions on LIN module; 2. Proofread the overall document.	2023/09/26	Eric Deng

Copyright Notice

Right to make changes — Fortior Technology (Shenzhen) Co., Ltd. reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. The information contained in this manual is provided for the general use by our customers. Our customers shall ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Fortior Technology (Shenzhen) Co., Ltd. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

If there are any differences between the Chinese and the English contents, please take the Chinese version as the standard.

Fortior Technology (Shenzhen) Co., Ltd.

Room 203, 2/F, Building No.11, Keji Central Road 2,

Software Park, High-Tech Industrial Park, Shenzhen, P.R. China 518057

Tel: 0755-26867710

Fax: 0755-26867715

URL: http://www.fortiortech.com