Datasheet

IPM for Three-phase Motor Control
ES9676AH

Fortior Technology 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 Direct Current
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
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
PFC	Power Factor Correction
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:
  - External power supply 12V~20V is connected to VCC pin, and internal LDO supplies VDD5 voltage.
- MOSFET:
  - N-Channel:
    - V_{DS} = 600V
    - I_D = 4A
    - R_{DS} = 1.1Ω
- Dual core: 8051 core and ME core. ME core achieves automatic calculation of FOC or square-wave control by the hardware for BLDC motors, and 8051 core is used for parameter configuration and routine processing.
- 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 LPF module, PI regulator, BLDC module, FOC module, MDU auxiliary computing module
- 1T 16x16 multiplier, 16T 32/16 divider
- 15 interrupt sources with 4 configurable priority levels
- 18*GPIOs
- Timers:
  - 2*Programmable timers with capture feature
  - 1*QEP decoding programmable timer
  - 1*BLDC motor dedicated timer
  - 1*General-purpose timer
  - 1*RTC
- Communication interfaces:
  - 1*SPI
  - 1*UART, supporting single-wire mode
  - Dual-channel DMA: supporting data transmission via SPI/UART
- Analog peripherals:
  - 12-bit ADC, operating with 1μs conversion time and internal VREF or external VREF selectable as reference voltage

- Number of ADC channels: 10
- Internal VREF. 3V, 4V, 4.5V and VDD5 can be selected as the internal reference.
- Three standalone operational amplifiers
- 2-channel analog comparator
- 9-bit DAC

- FOC module supports single/dual/triple-shunt current sampling
- FOC module supports overmodulation
- System clock
  - Built-in 20MHz high-speed RC oscillator
  - Built-in 32.8kHz low-speed RC oscillator
- WDT
- LVD
- TSD
- Two-wire FICE protocol based in-circuit emulation

1.2 Applications

The chip can be used for the drive of sensorless or sensored BLDC/PMSM motors, single-phase/three-phase induction motors and servo motors.

Applications: High-voltage indoor and outdoor units, high-voltage water pumps, high-voltage vacuum cleaners, high-voltage exhausters, etc.

1.3 Overview

ES9676AH is a fully-integrated IPM with motor controller, driver and power IC. Its high-performance motor drive chip incorporates ME core and 8051 core. ME core integrates FOC, MDU, LPF, PI 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, UART, Timers, PWM module and built-in high-voltage LDO, which are suitable for FOC or square-wave based BLDC/PMSM motors.

Package of ES9676AH: WBQHFBP-45L.

1.4 Functional Block Diagram

1.4.1 ES9676AH

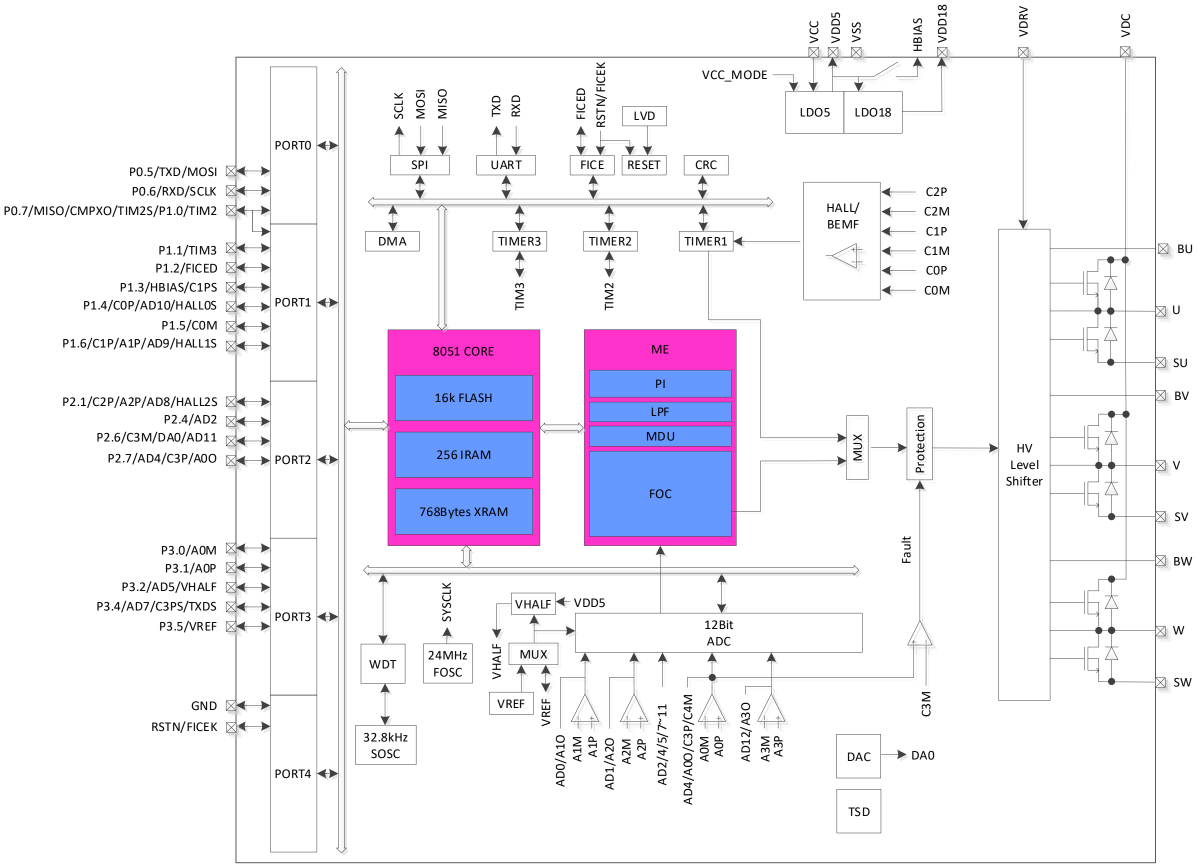


Figure 1-1 ES9676AH Functional Block Diagram

1.5 Memory Organization

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

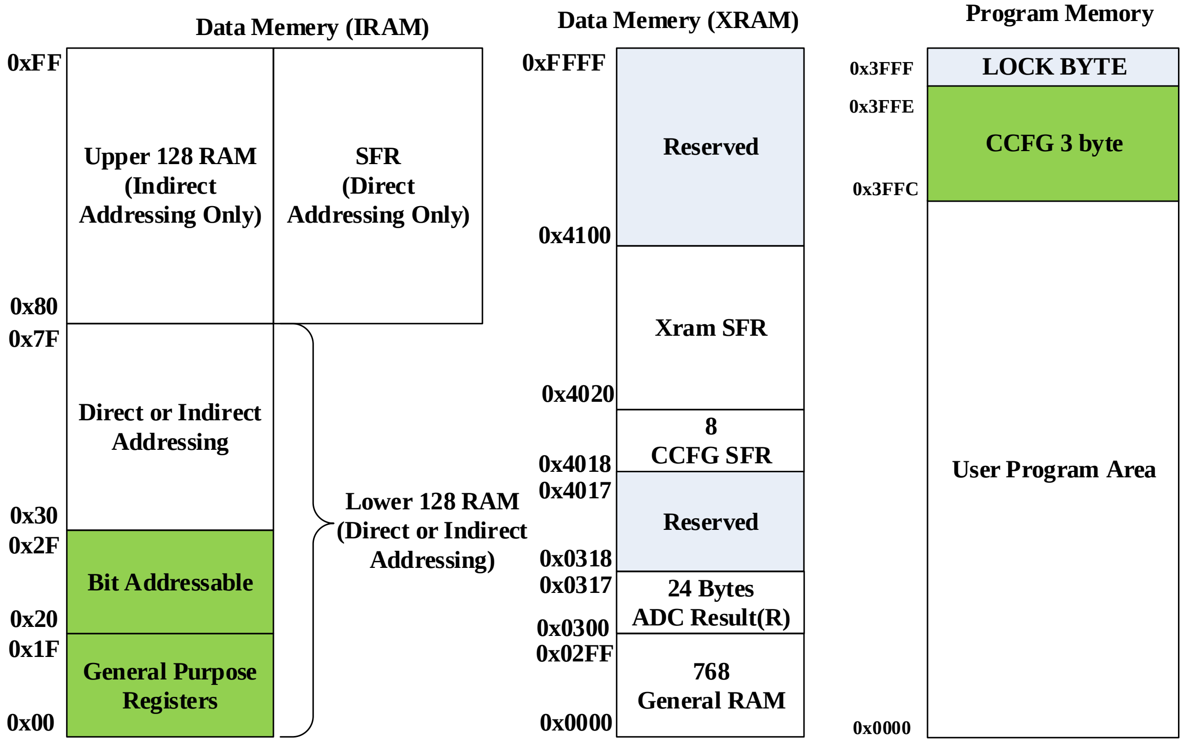


Figure 1-2 Memory Organization

1.5.1 Program Memory

The chip implements this program memory as Flash memory with a block from addresses 0x0000 to 0x3FFF to store control programs. CPU starts from 0x0000 after reset.

1.5.2 Data Memory

The data memory is divided into External Data Memory and Internal Data Memory&SFRs.

The External Data Memory is addressed from 0x0000 to 0x02FF, which can be accessed with MOVX instructions only.

The Internal Data Memory is shown in Figure 1-2. Locations 0x00~0x1F are addressable as 4 banks of general-purpose registers, each bank consisting of 8 registers. Locations 0x20~0x2F are 16-bit addressable, and locations 0x30~0x7F support direct and indirect addressing. 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 SFRs. Stack space is located in the Internal Data Memory.

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_LPF_CR			P0_OE	P1_OE	P2_OE	P3_OE
0xF0	B		PI_KIL	PI_KIH	PI_UKMAXL	PI_UKMAXH	PI_UKMINL	PI_UKMINH
0xE8	P4	P4_OE	PI_EKL	PI_EKH	PI_UKL	PI_UKH	PI_KPL	PI_KPH
0xE0	ACC						LPF_YL	LPF_YH
0xD8	IP3	EVT_FILT	CMP_CR2	LVSR	CMP_CR3	LPF_K	LPF_XL	LPF_XH
0xD0	PSW	P1_IE	P1_IF	P2_IE	P2_IF	CMP_CR0	CMP_CR1	CMP_SR
0xC8	IP2	RST_SR	MDU_MBL	MDU_MBH	MDU_DB0	MDU_DB1		
0xC0	IP1	MDU_CR	MDU_MAL	MDU_MAH	MDU_DA0	MDU_DA1	MDU_DA2	MDU_DA3
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							
0x80	P0	SP	DPL	DPH	FLA_KEY	FLA_CR		PCON

Notes:

- Registers with 4 low-order bits as 0 or 8 support addressing access.
- Registers containing the symbol "__" shall be read using variables. If it is read directly, the value will be incorrect.

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)
0x40f0	EXT0	TIM234_CTRL	CMP_AMP	TSD_ADJ				
0x40e8	FOC_ID_LPFK	FOC_IQ_LPFK	FOC_KFGH	FOC_KFGL			FOC_CR3	
0x40e0	FOC__EMFH	FOC__EMFL	FOC_UDCPSH	FOC_UDCPSL	FOC_UQCPSH	FOC_UQCPSL	FOC__UQEXH	FOC__UQEXL
0x40d8	FOC__POWH	FOC__POWL	FOC__IAMAXH	FOC__IAMAXL	FOC__IBMAXH	FOC__IBMAXL	FOC__ICMAXH	FOC__ICMAXL
0x40d0	FOC__EALPH	FOC__EALPL	FOC__EBETH	FOC__EBETL	FOC__EOMEH	FOC__EOMEL	FOC__ESQUH	FOC__ESQUL
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
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/CMP_SAMR	FOC_THECOMPH	FOC_THECOMPL
0x40a0	FOC_CR1	FOC_CR2	FOC_TSMIN	FOC_TGLI	FOC_TBLO	FOC_TRGDLY	FOC_CSOH	FOC_CSOL
0x4098	FOC__UDCFLTH/TIM1__ITRIPH	FOC__UDCFLTL/TIM1__ITRIPL						
0x4090	FOC_IDREFH	FOC_IDREFL	FOC_IQREFH	FOC_IQREFL	FOC_DQKPH	FOC_DQKPL	FOC_DQKIH	FOC_DQKIL
0x4088	FOC_EK3H/TIM1__RARRH	FOC_EK3L/TIM1__RARRL	FOC_EK4H/TIM1__RCNTRH	FOC_EK4L/TIM1__RCNTRL	FOC_EK1H	FOC_EK1L	FOC_EK2H	FOC_EK2L
0x4080	FOC_FBASEH/TIM1_DBR7H	FOC_FBASEL/TIM1_DBR7L	FOC_EFREQACCH/TIM1__BCNTRH	FOC_EFREQACCL/TIM1__BCNTRL	FOC_EFREQMINH/TIM1__BCCRH	FOC_EFRQMINL/TIM1__BCCRL	FOC_EFREQHOLDH/TIM1__BARRH	FOC_EFREQHOLDL/TIM1__BARRL
0x4078	FOC_KSLIDEH/TIM1_DBR3H	FOC_KSLIDEL/TIM1_DBR3L	FOC_EKLPFMINH/TIM1_DBR4H	FOC_EKLPFMINL/TIM1_DBR4L	FOC_EBMFKH/TIM1_DBR5H	FOC_EBMFKL/TIM1_DBR5L	FOC_OMEKLPFH/TIM1_DBR6H	FOC_OMEKLPFL/TIM1_DBR6L
0x4070	TIM1_BCORH	TIM1_BCORL			FOC_EKPH/TIM1_DBR1H	FOC_EKPL/TIM1_DBR1L	FOC_EKIH/TIM1_DBR2H	FOC_EKIL/TIM1_DBR2L
0x4068	TIM1_CR0	TIM1_CR1	TIM1_CR2	TIM1_CR3	TIM1_CR4	TIM1_ IER	TIM1_SR	
0x4060	DRV_DTR	DRV_SR	DRV_CR		SYST_ARRH	SYST_ARRL		
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				DAC_DR	PH_SEL		AMP_CR	VREF_VHALF_CR
0x4040	CAL_CR0	CAL_CR1						
0x4038	ADC_SCYC	ADC_CR	DMA0_CR0	DMA1_CR0	DMA0_CR1H	DMA0_CR1L	DMA1_CR1H	DMA1_CR1L

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x4030	SPI_CR0	SPI_CR1	SPI_CLK	SPI_DR		DAC_CR	ADC_MASK_SYSCH	ADC_MASK_SYSCL
0x4028					RTC_TMH	RTC_TML	RTC_STA	
0x4020		CRC_DIN	CRC_CR	CRC_DR	CRC_BEG	CRC_CNT	WDT_CR	WDT_REL
0x4018	CCFG7	CCFG6	CCFG5	CCFG4	CCFG3	CCFG2	CCFG1	CCFG0
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			AD7_DRH	AD7_DRL
0x0300	AD0_DRH	AD0_DRL	AD1_DRH	AD1_DRL	AD2_DRH	AD2_DRL		
0x03f8	LPF0_K		LPF0_X		LPF0_YH		LPF0_YL	
0x03f0	LPF1_K		LPF1_X		LPF1_YH		LPF1_YL	
0x03e8	PI0_UKH		PI0_UKL		PI0_UKMAX		PI0_UKMIN	
0x03e0	PI0_KP		PI0_EK1		PI0_EK		PI0_KI	
0x03d8	PI1_UKH		PI1_UKL		PI1_UKMAX		PI1_UKMIN	
0x03d0	PI1_KP		PI1_EK1		PI1_EK		PI1_KI	
0x03C8	MUL0_MA		MUL0_MB		MUL0_MCH		MUL0_MCL	
0x03C0	MUL1_MA		MUL1_MB		MUL1_MCH		MUL1_MCL	
0x03B8	DIV0_DB		DIV0_DQH		DIV0_DQL		DIV0_DR	
0x03B0	DIV1_DQL		DIV1_DR		DIV0_DAH		DIV0_DAL	
0x03A8	DIV1_DAH		DIV1_DAL		DIV1_DB		DIV1_DQH	
0x03A0	LPF2_K		LPF2_X		LPF2_YH		LPF2_YL	
0x0398	LPF3_K		LPF3_X		LPF3_YH		LPF3_YL	
0x0390	PI2_UKH		PI2_UKL		PI2_UKMAX		PI2_UKMIN	
0x0388	PI2_KP		PI2_EK1		PI2_EK		PI2_KI	
0x0380	PI3_UKH		PI3_UKL		PI3_UKMAX		PI3_UKMIN	
0x0378	PI3_KP		PI3_EK1		PI3_EK		PI3_KI	
0x0370	MUL2_MA		MUL2_MB		MUL2_MCH		MUL2_MCL	
0x0368	MUL3_MA		MUL3_MB		MUL3_MCH		MUL3_MCL	
0x0360	DIV2_DB		DIV2_DQH		DIV2_DQL		DIV2_DR	
0x0358	DIV3_DQL		DIV3_DR		DIV2_DAH		DIV2_DAL	

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x0350	DIV3_DAH		DIV3_DAL		DIV3_DB		DIV3_DQH	
0x0348	SCAT0_SIN		SCAT0_THE		SCAT0_RES1		SCAT0_RES2	
0x0340	SCAT1_THE		SCAT1_RES1		SCAT1_RES2		SCAT0_COS	
0x0338	SCAT2_RES1		SCAT2_RES2		SCAT1_COS		SCAT1_SIN	
0x0330	SCAT3_RES2		SCAT2_COS		SCAT2_SIN		SCAT2_THE	
0x0328	SCAT3_COS		SCAT3_SIN		SCAT3_THE		SCAT3_RES1	

Note: Registers containing the symbol “__” shall be read using variables. If it is read directly, the value will be incorrect. The SFR is mapped partly to SFR sector of the Internal Data Memory, and partly to External Data Memory (also known as XSFR).

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 ES9676AH WBQHFBP-45L Pins

Table 2-1 ES9676AH WBQHFBP-45L Pin Descriptions

Pin	ES9676AH WBQHFBP-45L	IO Type	Function Description
VDRV	1	P	6N pre-driver power supply, with a 10μF external capacitor
VCC	2	P	Power input, with an external filter capacitor of 10μF or above. ■ High-voltage single-power supply mode: External power supply 12V~20V is connected to VCC pin, and internal LDO supplies VDD5 voltage.
VDD5	3	P	Medium-voltage power input or 5V LDO power output, with a 1μF ~4.7μF external capacitor. When VCC > 5.5V, VDD5 outputs 5V power supply.
RSTN/ FICEK	4	DI/ DI	Input of external reset; Built-in pull-up resistor; Schmit input FICE SCL
VDD18	5	P	1.85V LDO output with an external 1μF ~ 4.7μF capacitor
P1.2/ FICED	6	DB/ DB	GPIO, configurable as INT1 input FICE SDA
P1.3/ HBIAS/ C1PS	7	DB/ DO/ AI	GPIO Hall bias power supply, internally connected to VDD5 via a switch CMP1 positive input after function switching
P1.4/ C0P/ AD10/ HALL0S	8	DB/ AI/ AI/ DI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP0 positive input Input of ADC channel 10 Hall0 logic level input after function switching
P1.5/ C0M	9	DB/ AI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP0 negative input
P1.6/ C1P/ A1P/ AD9/ HALL1S	10	DB/ AI/ AI/ AI/ DI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP1 positive input AMP1 positive input for voltage signals generated by phase current 1 Input of ADC channel 9 Hall1 logic level input after function switching
P2.1/ C2P/ A2P/ AD8/ HALL2S	11	DB/ AI/ AI/ AI/ DI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP2 positive input AMP2 positive input for voltage signals generated by phase current 2 Input of ADC channel 8 Hall2 logic level input after function switching
P2.4/ AD2	12	DB/ AI	GPIO, configurable as INT1 input Input of ADC channel 2 or bus voltage signal
P2.6/ C3M/ DA0/ AD11	13	DB/ AI/ AO/ AI	GPIO, configurable as INT1 input Over-current reference signal input; CMP3 negative input Internal DAC voltage, without Buffer output Input of ADC channel 11

Pin	ES9676AH WBQHFBP-45L	IO Type	Function Description
P2.7/ AD4/ C3P/ A0O	14	DB/ AI/ AI/ AO	GPIO, configurable as INT1 input Input of ADC channel 4 for collecting the amplified bus current signal CMP3 positive input for bus current sampling to detect overcurrent AMP0 output, the voltage output after the bus current is amplified
P3.0/ A0M	15	DB/ AI	GPIO AMP0 negative input for amplifying the bus current signal
P3.1/ A0P	16	DB/ AI	GPIO AMP0 positive input for amplifying the bus current signal
P3.2/ AD5/ VHALF	17	DB/ AI/ AO	GPIO Input of over-temperature signal or ADC channel 5 1/2 VDD5 or 1/2 VREF output with an external 1μF capacitor
P3.4/ AD7/ C3PS/ TXDS	18	DB/ AI/ AI/ DO	GPIO Analog voltage input for speed control or input of ADC channel7 CMP3 positive input after function switching UART TXD after function switching
P3.5/ VREF	19	DB/ AI	GPIO ADC external VREF input or internal VREF output, with a 1μF~4.7μF external capacitor
GND	20	P	Ground
P0.5/ TXD/ MOSI	21	DB/ DO/ DB	GPIO, configurable as INT0 input UART1 TXD SPI MOSI, master output or slave input
P0.6/ RXD/ SCLK	22	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 SCLK
P0.7/ MISO/ CXO/ TIM2S/ P1.0/ TIM2	23	DB/ DB/ DO/ DB/ DB/ DB	GPIO P0.7 SPI MISO, master input or slave output Test pin for comparator output Timer2 input capture mode or PWM output after function switching GPIO P1.0, configurable as INT1 input Timer2 input capture mode or PWM output
P1.1/ TIM3	24	DB/ DB	GPIO, configurable as INT0/INT1 input Timer3 input capture mode
SU	25	P	U-phase ground
SU	26	P	
U	27	DO	U-phase output
U	28	DO	
VM	29	P	Input of MOS power supply or high-voltage power supply
VM	30	P	
VM	31	P	
VM	32	P	
V	33	DO	V-phase output
V	34	DO	
SV	35	P	V-phase ground
SV	36	P	
VM	37	P	Input of MOS power supply or high-voltage power supply
VM	38	P	
SW	39	P	W-phase ground
SW	40	P	
W	41	DO	W-phase output
W	42	DO	

Pin	ES9676AH WBQHFBP-45L	IO Type	Function Description
BW	43	P	W-phase high-side bootstrap supply
BV	44	P	V-phase high-side bootstrap supply
BU	45	P	U-phase high-side bootstrap supply

2.2 ES9676AH WBQHFBP-45L Pinout Diagram

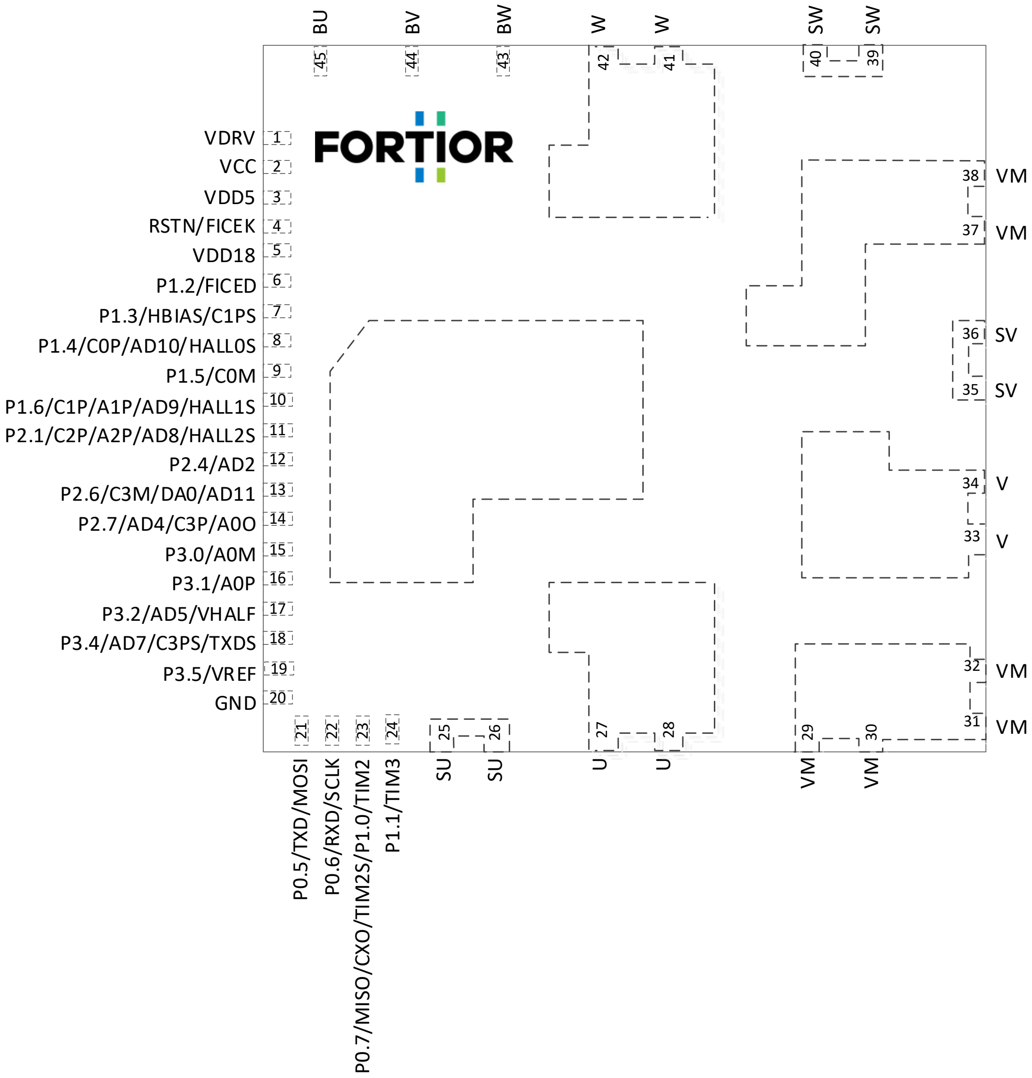


Figure 2-1 ES9676AH WBQHFBP-45L Pinout Diagram

3 Package Information

3.1 WBQHFBP-45L_12X12

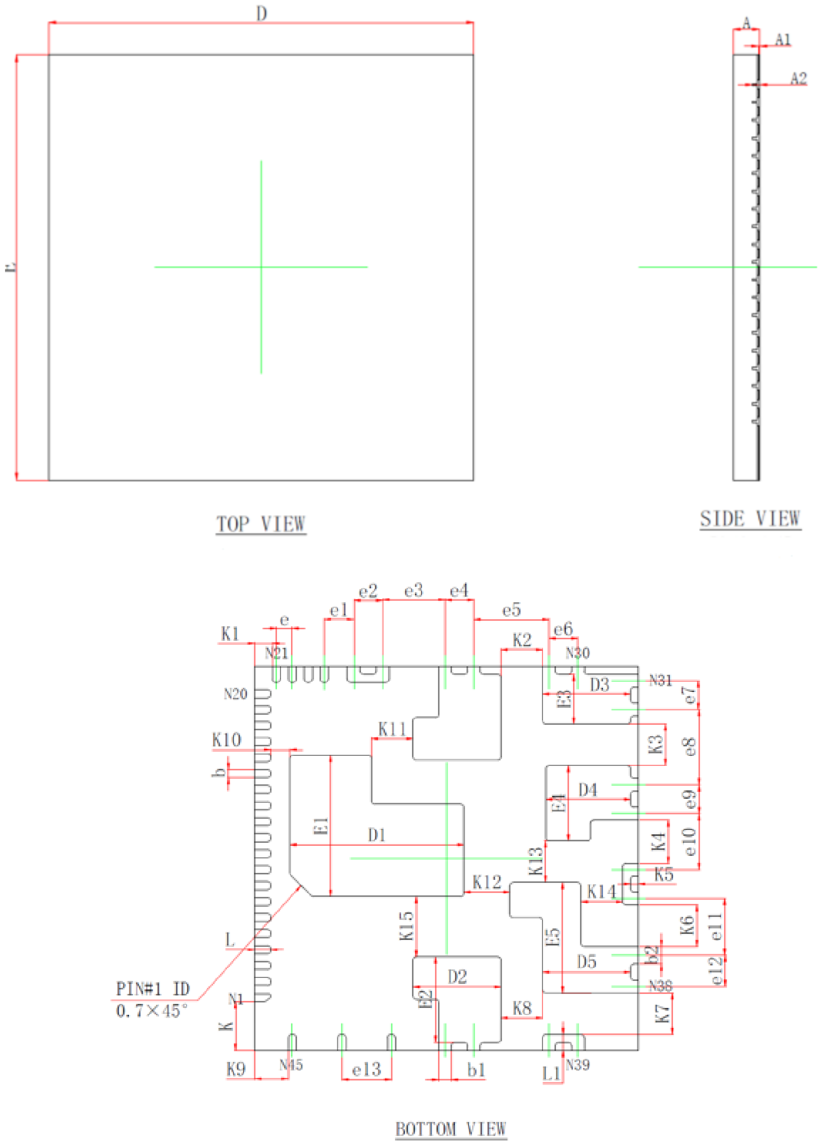
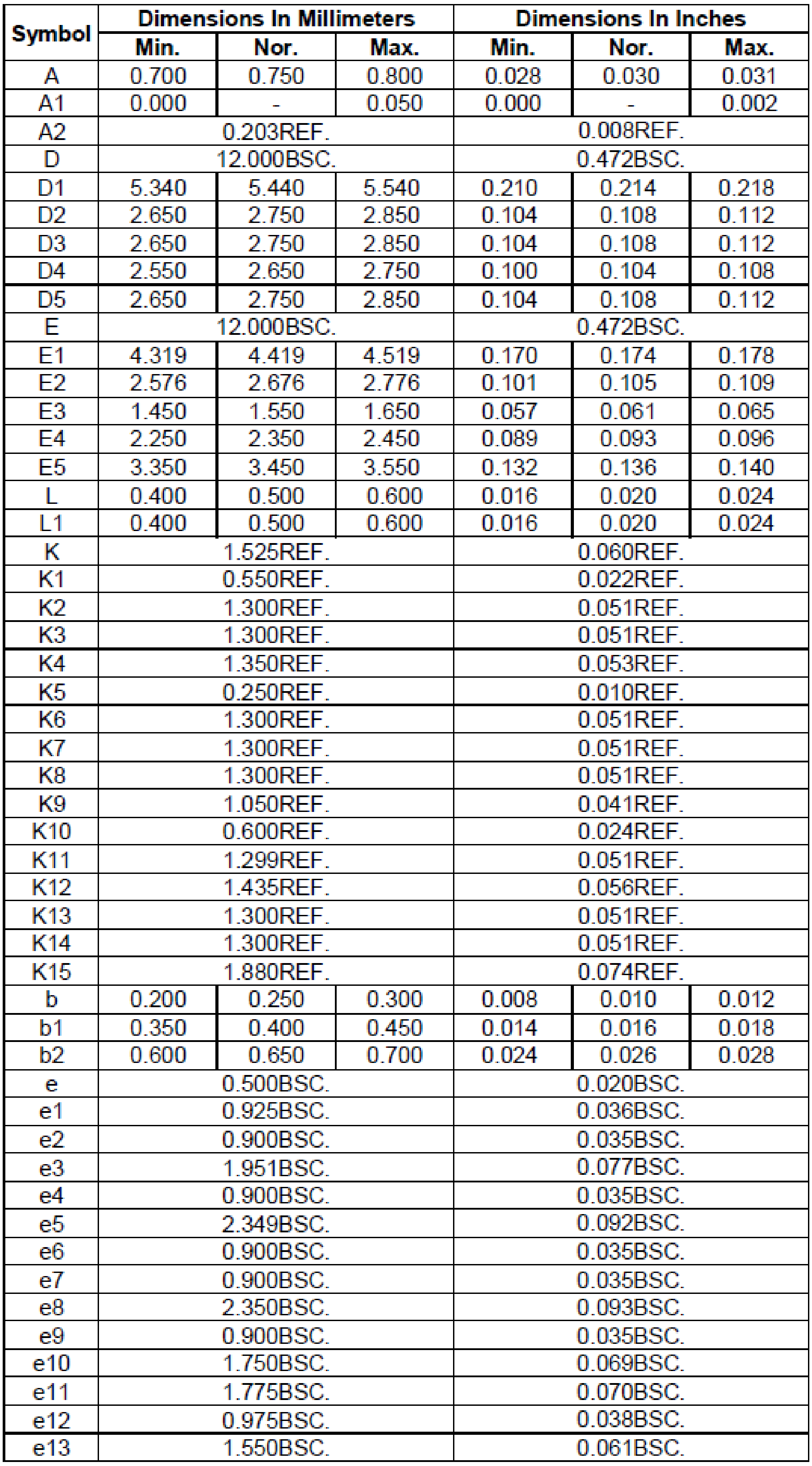


Figure 3-1 WBQHFBP-45L_12X12 Package Drawings

Table 3-1 WBQHFBP-45L Package Dimensions



Symbol	Dimensions In Millimeters			Dimensions In Inches		
	Min.	Nor.	Max.	Min.	Nor.	Max.
A	0.700	0.750	0.800	0.028	0.030	0.031
A1	0.000	-	0.050	0.000	-	0.002
A2	0.203REF.			0.008REF.		
D	12.000BSC.			0.472BSC.		
D1	5.340	5.440	5.540	0.210	0.214	0.218
D2	2.650	2.750	2.850	0.104	0.108	0.112
D3	2.650	2.750	2.850	0.104	0.108	0.112
D4	2.550	2.650	2.750	0.100	0.104	0.108
D5	2.650	2.750	2.850	0.104	0.108	0.112
E	12.000BSC.			0.472BSC.		
E1	4.319	4.419	4.519	0.170	0.174	0.178
E2	2.576	2.676	2.776	0.101	0.105	0.109
E3	1.450	1.550	1.650	0.057	0.061	0.065
E4	2.250	2.350	2.450	0.089	0.093	0.096
E5	3.350	3.450	3.550	0.132	0.136	0.140
L	0.400	0.500	0.600	0.016	0.020	0.024
L1	0.400	0.500	0.600	0.016	0.020	0.024
K	1.525REF.			0.060REF.		
K1	0.550REF.			0.022REF.		
K2	1.300REF.			0.051REF.		
K3	1.300REF.			0.051REF.		
K4	1.350REF.			0.053REF.		
K5	0.250REF.			0.010REF.		
K6	1.300REF.			0.051REF.		
K7	1.300REF.			0.051REF.		
K8	1.300REF.			0.051REF.		
K9	1.050REF.			0.041REF.		
K10	0.600REF.			0.024REF.		
K11	1.299REF.			0.051REF.		
K12	1.435REF.			0.056REF.		
K13	1.300REF.			0.051REF.		
K14	1.300REF.			0.051REF.		
K15	1.880REF.			0.074REF.		
b	0.200	0.250	0.300	0.008	0.010	0.012
b1	0.350	0.400	0.450	0.014	0.016	0.018
b2	0.600	0.650	0.700	0.024	0.026	0.028
e	0.500BSC.			0.020BSC.		
e1	0.925BSC.			0.036BSC.		
e2	0.900BSC.			0.035BSC.		
e3	1.951BSC.			0.077BSC.		
e4	0.900BSC.			0.035BSC.		
e5	2.349BSC.			0.092BSC.		
e6	0.900BSC.			0.035BSC.		
e7	0.900BSC.			0.035BSC.		
e8	2.350BSC.			0.093BSC.		
e9	0.900BSC.			0.035BSC.		
e10	1.750BSC.			0.069BSC.		
e11	1.775BSC.			0.070BSC.		
e12	0.975BSC.			0.038BSC.		
e13	1.550BSC.			0.061BSC.		

4 Ordering Information

Table 4-1 Model Selections

Model	Rdson (H+L) (Ω)	MIPS (Peak)	Flash (kByte)	XRAM (kByte)	Clock Circuit		Driver Type		SPI/UART	DMA	GPIO	Timer	Analog Peripherals								Lead-free	Package
					Internal Fast Clock	Internal Slow Clock	Square Wave	FOC					ADC			DAC		VREF	Operational Amplifier	Comparator		
													Number	Channel	Bits	Number	Bits					
ES9676AH	2.2	24	16	0.75	√	√	√	√	√	√	18	6	1	10	12	1	9	√	3	2	√	WBQHFBP-45L (12x12mm)

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

Table 5-1 Absolute Maximum Ratings[1]

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
MOSFET Drain-to-Source Voltage V_{DSS}		600	-	-	V
Continuous Drain Current $I_{DMAX(DC)}$ of Single MOS	T_C = 25°C	-	-	4[2]	A
Pulse Drain Current $I_{DMAX(PLS)}$ of Single MOS	T_C = 25°C	-	-	11[2][3]	A
Power Dissipation P_d of Single MOS[4]	T_C = 25°C	-	-	64.4	W
High-side VDC Supply Voltage V_{DC}		-0.3	-	600[2]	V
U/V/W-phase Output Voltage V_U, V_V, V_W		-0.3	-	600[2]	V
High-side Floating Absolute Voltage V_{BU}, V_{BV}, V_{BW}		-0.3	-	600[2]	V
High-side Floating Supply Voltage V_{BU} - V_U, V_{BV} - V_V, V_{BW} - V_W		-0.3	-	22	V
Operating Ambient Temperature T_A		-40	-	85	℃
Operating Junction Temperature T_J		-40	-	150	℃
Storage Temperature		-55	-	150	℃
VCC to VSS Voltage		-0.3	-	36	V
VDRV to VSS Voltage		-0.3	-	22	V
VDD5 to VSS Voltage		-0.3	-	6.5	V
RSTN/GPIO to VSS Voltage		-0.3	-	VDD5 + 0.3	V

Notes:

[1] 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.

[2] Power dissipation shall not exceed P_d or ASO;

[3] $P_w \leq 10\mu s$ and duty cycle ≤ 1%;

[4] T0252 package is applied to single-channel MOS at ambient temperature of 25℃.

5.2 Recommended Operating Conditions

Table 5-2 Recommended Operating Conditions

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
High-side VDC Supply Voltage V_{DC}		-	310	400	V
High-side Floating Supply Voltage V_{BU} - V_U, V_{BV} - V_V, V_{BW} - V_W		12	15	20	V
VCC/VDRV to VSS Voltage		11	15	20	V
Junction Temperature T_J		-	-	125	℃

5.3 Global Electrical Characteristics

Table 5-3 Global Electrical Characteristics

(T_A = 25°C and VCC = 12V ~ 20V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VCC/VDRV Operating Voltage[1][2]		12	-	20	V
Drain-to-source Leakage Current IDSS of Single-channel MOS		-	-	100	µA
Static Drain-to-source On Resistance of Single-channel MOS		-	1.1	1.5	Ω
Drain-to-source Diode Forward Voltage Drop of Single-channel MOS		-	-	1.4	V
I_{VCC} Operating Current[3]		-	24	-	mA
I_{VCC} Standby Current[3]		-	6	-	mA
I_{VCC} Sleep-mode Current		-	210	-	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.

5.4 GPIO Electrical Characteristics

Table 5-4 GPIO Electrical Characteristics

(T_A = 25°C and VCC = 12V ~ 20V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Output Rise Time	50pF load, from 10% to 90%, T_A = 25°C	-	15	-	ns
Output Fall Time	50pF load, from 90% to 10%, T_A = 25°C	-	13	-	ns
V_{OH} High-level Output Voltage	I_{OH} = 4mA, T_A = 40°C ~ 85°C	VDD5 - 0.7	-	-	V
V_{OL} Low-level Output Voltage	I_{OL} = 4mA, T_A = 40°C ~ 85°C	-	-	VSS + 0.7	V
V_{IH} High-level Input Voltage[1]		0.7*VDD5	-	-	V
V_{IL} Low-level Input Voltage		-	-	0.2*VDD5	V
Pull-up Resistor[2]		-	33	-	kΩ
Pull-up Resistor[3]		-	5.6	-	kΩ

Notes:

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

[2] GPIOs except P1[6:3] and P2[1].

[3] P1[6:3] and P2[1].

5.5 ADC Electrical Characteristics

Table 5-5 ADC Electrical Characteristics

(T_A = 25°C and VCC = 12V ~ 20V 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
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 and VHALF Electrical Characteristics

(T_A = -40°C ~ 85°C and VCC = 12V ~ 20V)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VREF	VREF_CR[VREFVSEL] = 00	4.3	4.5	4.7	V
	VREF_CR[VREFVSEL] = 01	-	VDD5	-	V
	VREF_CR[VREFVSEL] = 11	-	4	-	V
	VREF_CR[VREFVSEL] = 10	-	3	-	V
VHALF	VREF_CR[VHALFSEL] = 00	-	VREF/8	-	V
	VREF_CR[VHALFSEL] = 01	-	VREF/4	-	V
	VREF_CR[VHALFSEL] = 10	-	25*VREF/64	-	V
	VREF_CR[VHALFSEL] = 11	VREF/2 - 0.2	VREF/2	VREF/2 + 0.2	V

5.7 Operational Amplifier Electrical Characteristics

Table 5-7 Operational Amplifier Electrical Characteristics

(T_A = 25°C and VCC = 12V ~ 20V 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

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Slew Rate (SR)	C_L = 40pF	10	15	-	V/µs
Operational Amplifier Gain[1]	AMP_CR1[AMP0_GAIN] = 001	1.88	2	2.12	-
	AMP_CR1[AMP0_GAIN] = 010	3.76	4	4.24	-
	AMP_CR1[AMP0_GAIN] = 011	7.5	8	8.5	-
	AMP_CR1[AMP0_GAIN] = 100	15	16	17	-

Note:

[1] The operational amplifier gain is measured when both positive and negative inputs of the operational amplifier are connected in series with 1kΩ resistors. The operational amplifier gain varies with external resistors.

5.8 BEMF Electrical Characteristics

Table 5-8 BEMF Electrical Characteristics

(T_A = 25°C, VCC_MODE = 0 and VCC = 12V ~ 20V 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 ~ 85°C, VCC_MODE = 0 and VCC = 12V ~ 20V)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
System Clock Rate		23.5	24	24.5	MHz
Low-speed Clock Rate		29	32.8	37	kHz

Note: SYSCLK refers to system clock rate, and T to system clock cycle. Unless otherwise specified, the system clock rate of chip is 24MHz and T = 1/SYSCLK.

5.10 Reset Electrical Characteristics

Table 5-10 Reset Electrical Characteristics

(T_A = 25°C, VCC_MODE = 0 and VCC = 12V ~ 20V 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 = 3.0V	2.8	3.0	3.2	V

5.11 LDO Electrical Characteristics

Table 5-11 LDO Electrical Characteristics

(T_A = 25°C, VCC_MODE = 0 and VCC = 12V ~ 20V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD5 Voltage	VCC = 12V ~ 20V and VCC_MODE = 0	4.7	5	5.3	V
VDD18 Voltage		1.65	1.85	2.0	V

5.12 Package Thermal Resistance

Table 5-12 Package Thermal Resistance

Parameter	Test Conditions	Value	Unit
Junction-to-ambient Temperature Thermal Resistance θ_{JA}[1]	JEDEC standard, 2S2P PCB	27.7	°C/W

Note:

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

6 Reset Control

6.1 Reset Source (RST_SRC)

The chip includes a reset circuitry with 6 reset sources:

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

The reset flag is queryable and recorded in register RST_SR. Following the last reset, the affected reset flag is set to "1" and all other reset flags are cleared to "0". You can set RST_SR[RSTCLR] flag to "1" in the software to clear the reset flag RST_SR[7:3].

6.2 Reset Enable

See corresponding control registers. Reset sources of LVD and WDT are always enabled.

6.3 External Reset and Power-on Reset

The chip resets if RSTN pin remains low for 50µs. After reset, MCU starts the program from address 0x0000. The chip also resets after the chip powers on and the voltage settles above the reset threshold.

6.4 Low Voltage Detector reset

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

Configuring corresponding register enables VDD5 monitor circuitry and sets V_{RST}.

6.5 Watchdog Timer Reset

After the watchdog timer (WDT) is enabled, the system is reset if WDT is not initialized before it get overflowed. In this case, when WDT reaches its maximum value, it 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 the software using MOVX instruction for read/write/erase operations (See section 34 Code Protection). A Flash error detector reset (RSTFED) occurs if a Flash erase is attempted targeting the last sector (0x3F80~0x3FFF). 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 Reset Registers

6.8.1 RST_SR (0xC9)

Bit	7	6	5	4	3	2	1	0
Name	RSTPOR	RSTEXT	RSTLVD	RSV	RSTWDT	RSTFED	RSTDBG	RSTCLR
Type	R	R	R	-	R	R	R	W1
Reset	-	-	-	-	-	-	-	0

Bit	Name	Description
[7]	RSTPOW	Power-On Reset Flag 0: Last reset was not a power-on reset. 1: Last reset was a power-on reset.
[6]	RSTEXT	External Reset Flag 0: Last reset was not an external reset. 1: Last reset was an external 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]	RSTCLR	Clear Analog Reset Flags Register The reset flags [7:3] are cleared after “1” is written to this bit. This bit has no effect when it is read.

7 Interrupt

7.1 Interrupt Introduction

The chip includes an interrupt system with a total of 15 interrupt sources. Each interrupt source can be individually programmed in IP0 ~ IP3 registers with one of four priority levels. Interrupt flags (IF) are located in an SFRs or XSFRs. The associated IF is set by the hardware to 1 when the internal circuitry or an external source 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 generated and 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 interrupted 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 level, a fixed priority order is used to arbitrate. See Table 7-1 Interrupt Summary 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. The MCU does not respond to any interrupt request when IE[EA] = 0.

Each interrupt source 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. Once the enable bit is set to “1”, MCU immediately enters the interrupt subroutine. 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. They both can be configured as interrupt on rising edge, interrupt on falling edge or interrupt on edge changes (rise or fall).

The digital input signals from P0.5 ~ P0.6 and P1.1 can be used to trigger an INT0. The interrupt source is selected through LVSR[EXT0CFG] bit. These trigger sources share one interrupt entry point, 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 ~ 1.6, P2.1, P2.4 and P2.6 ~ 2.7 can be used to trigger an INT1. P1_IF and P4_IF are interrupt flag bits, and P1_IE and P4_IE 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 P1_IF and P4_IF in the interrupt subroutine. These 15 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 section 7.5.7 P1_IE (0xD1) ~ section 7.5.10 P4_IF (0xD4) for INT1 interrupt flags 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	0	0x0003	LVSR[0]	Y	CCFG1[6]	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 Interrupt	3	0x001B	DRV_SR[5:4]	Y	DRV_SR[2:0]	IP0[7:6]
TIM2 Interrupt	4	0x0023	TIM2_CR1[7:5]	Y	TIM2_CR1[4:3] TIM2_CR0[3]	IP1[1:0]
TIM1 Interrupt	5	0x002B	TIM1_SR[4:0]	Y	TIM_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[7:4]	Y	CMP_CR0[7:0]	IP1[7:6]
RTC	8	0x0043	RTC_STA[6]	Y	IE[6]	IP2[1:0]
TIM3 Interrupt	9	0x004B	TIM3_CR1[7:5]	Y	TIM3_CR1[4:3] TIM3_CR0[3]	IP2[3:2]
TIM4 Interrupt Systick Interrupt	10	0x0053	TIM4_CR1[7:5] DRV_SR[7]	Y	TIM4_CR1[4:3] TIM4_CR0[3] DRV_SR[6]	IP2[5:4]
TSD Interrupt	11	0x005B	TCON[5]	Y	IE[1]	IP2[7:6]
UART Interrupt	12	0x0063	UT_CR[1:0]	Y	IE[4]	IP3[1:0]
SPI Interrupt	13	0x0073	SPI_CR1[7]	Y	SPI_CR1[0]	IP3[5:4]
DMA Interrupt	14	0x007B	DMA0_CR0[7] DMA1_CR0[7]	Y	DMA0_CR0[2]	IP3[7:6]

As shown above, the chip includes an interrupt system with a total of 15 interrupt sources. Each interrupt source can be individually programmed in IP0 ~ IP3 registers with one of four priority levels. A low priority interrupt can be interrupted by a high priority interrupt. If two interrupt sources have the same priority level, they are processed in an order as listed in the above table, where the lower the mark the higher the priority level. An interrupt will not be preempted by another of the same priority level.

IE[EA] is the global interrupt enable bit. The MCU does not respond to any interrupt request when IE[EA] = 0.

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	UART Interrupt Enable 0: Disable 1: Enable
[3]	SPIIE	SPI Interrupt Enable 0: Disable 1: Enable
[2]	EX1	External Interrupt 1 (INT1) Enable 0: Disable 1: Enable
[1]	TSDIE	TSD Interrupt Enable 0: Disable 1: Enable
[0]	EX0	External Interrupt 0 (INT0) Enable 0: Disable 1: Enable

7.5.2 IP0 (0xB8)

Bit	7	6	5	4	3	2	1	0
Name	PDRV		PX1		PX0		PLVW	
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	Driver Interrupt Priority Setting
[5:4]	PX1	External Interrupt 1 (INT1) Priority Setting
[3:2]	PX0	External Interrupt 0 (INT0) Priority Setting
[1:0]	PLVW	LVW Interrupt Priority Setting

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 Setting
[5:4]	PADC	ADC Interrupt Priority Setting
[3:2]	PTIM1	Timer1 Interrupt Priority Setting
[1:0]	PTIM2	Timer2 Interrupt Priority Setting

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	PTSD		PTIM4		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]	PTSD	TSD Interrupt Priority Setting
[5:4]	PTIM4	Timer4 and Systick Interrupt Priority Setting
[3:2]	PTIM3	Timer3 Interrupt Priority Setting
[1:0]	PRTC	RTC Interrupt Priority Setting

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		RSV		PUART	
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]	PDMA	DMA Interrupt Priority Setting
[5:4]	PSPI	SPI Interrupt Priority Setting
[3:2]	RSV	Reserved
[1:0]	PUART	UART Interrupt Priority Setting

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 to “1” by hardware when the chip exceeds the set temperature. Read: 0: No Interrupt Pending 1: Interrupt Pending

		Write: 0: This bit is cleared to “0” 1: No effect
[4:3]	IT1[1:0]	External Interrupt 1 (INT1) Edge Select 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on edge changes (rise or fall)
[2]	IF0	External Interrupt 0 (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[1:0]	External Interrupt 0 (INT0) Edge Select 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on edge changes (rise or fall)

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

7.5.9 P4_IE (0xD3)

Bit	7	6	5	4	3	2	1	0
Name	P04_IE	P46_IE	P45_IE	P44_IE	P36_IE	P42_IE	P41_IE	P07_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]	P04_IE	P0.4 INT1 Enable 0: Disable 1: Enable
[6]	P46_IE	P4.6 INT1 Enable 0: Disable 1: Enable
[5]	P45_IE	P4.5 INT1 Enable 0: Disable 1: Enable
[4]	P44_IE	P4.4 INT1 Enable 0: Disable 1: Enable
[3]	P36_IE	P3.6 INT1 Enable 0: Disable 1: Enable

[2]	P42_IE	P4.2 INT1 Enable 0: Disable 1: Enable
[1]	P41_IE	P4.1 INT1 Enable 0: Disable 1: Enable
[0]	P07_IE	P0.7 INT1 Enable 0: Disable 1: Enable

7.5.10 P4_IF (0xD4)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	P04_IF	P46_IF	P45_IF	P44_IF	P36_IF	P42_IF	P41_IF	P07_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]	P04_IF	P0.4 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[6]	P46_IF	P4.6 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[5]	P45_IF	P4.5 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[4]	P44_IF	P4.4 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[3]	P36_IF	P3.6 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[2]	P42_IF	P4.2 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[1]	P41_IF	P4.1 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[0]	P07_IF	P0.7 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending

8 SPI

8.1 SPI Introduction

SPI provides access to a high-speed, full-duplex synchronous serial bus, with its block diagram shown in Figure 8-1. SPI can operate as a master or slave device in 3-wire or 4-wire mode, and supports multiple masters and slaves on a single SPI bus.

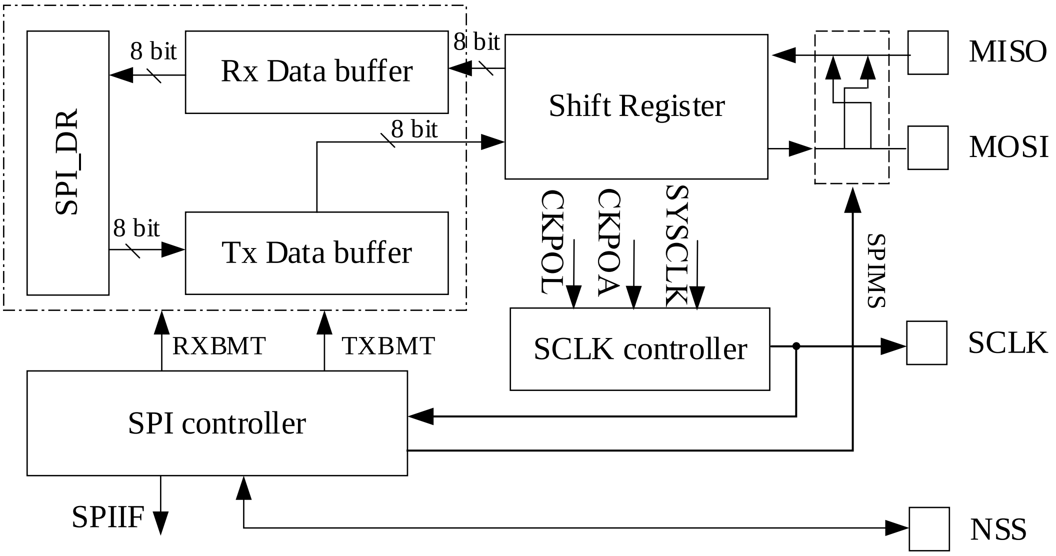


Figure 8-1 SPI Block Diagram

8.2 SPI Operations

8.2.1 Signal Descriptions

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

8.2.1.1 Master Output, Slave Input (MOSI)

The master-out, slave-in (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.

8.2.1.2 Master Input, Slave Output (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 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 MOSI pin.

8.2.1.3 Serial Clock (SCLK)

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

8.2.1.4 Slave Select (NSS)

The slave-select (NSS) is dependent on the setting of SPI_CR1[NSSMOD] bit. SPI may operate in 3-Wire Mode or 4-Wire Single Master Mode. 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 8-2.

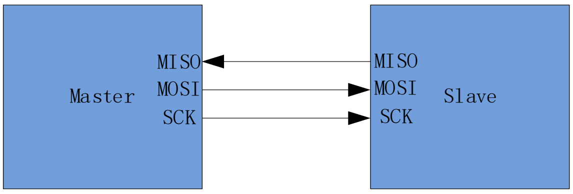


Figure 8-2 Connection Diagram of 3-Wire Single Master/Slave Mode

When SPI_CR1[NSSMOD] = 01, SPI operates in 4-Wire Mode. In this mode, NSS pins on the SPI bus are configured as inputs, waiting to be addressed by the master. When SPI_CR0[SPIMS] = 0, SPI operates in 4-Wire Slave Mode. In this case, data is transferred via SPI after the slave NSS is pulled down by two system cycles. When SPI_CR0[SPIMS] = 1, SPI operates in Multi-Master Mode. In this case, when NSS pin of a master on the SPI bus is pulled low, SPI_CR1[MODF] is set to “1” to disable the SPI and the master acts as a slave. The connection diagram is shown in Figure 8-3.

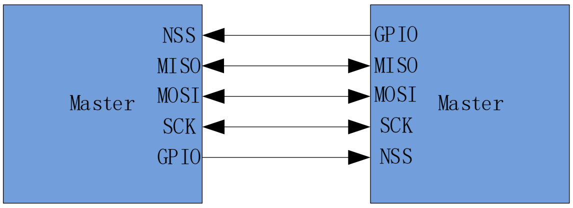


Figure 8-3 Connection Diagram of 4-Wire Multi-Master Mode

When SPI_CR1[NSSMOD] = 1X, SPI operates in 4-Wire Mode that is valid in master mode only. In this case, SPI_CR1[NSSMOD0] setting decides the output level of NSS pin. When SPI_CR1[NSSMOD0] = 1, NSS pin outputs high level, and when SPI_CR1[NSSMOD0] = 0, the pin outputs low level. The connection diagram is shown in Figure 8-4.

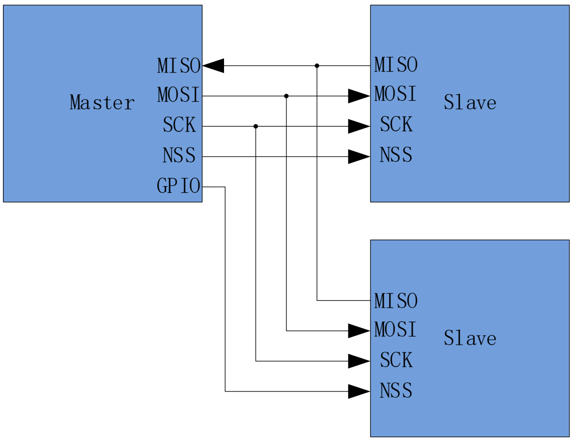


Figure 8-4 Connection Diagram of 4-Wire Master/Single Mode

8.2.2 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, then 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 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] flag serves as both a transmit-complete flag and a receive-data-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] will be set to "1" and the data in the transmit buffer keeps unchanged.

8.2.2.1 Master Mode Configurations

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

8.2.3 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 from MOSI pin and shifted out from MISO pin. If no SCLK signal is input, shift register of the slave is in the stop state. If the signal of SCLK is input, the shift register of slave starts to receive and transmit data through MOSI and MISO pins. The slave device cannot initiate transfers. The data sent to the master device is pre-loaded into the shift register by writing to SPI_DR. If the shift register is empty, the data in the transit buffer is transferred into 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 into receive butter and receive buffer empty flag bit SPI_CR0[RXBMT] is cleared, indicating the new data has not been read. If SPI_CR0[RXBMT] is 0 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.

8.2.3.1 Slave Mode Configurations

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 transmit the clock signal.

8.2.4 SPI Interrupt Sources

After SPI interrupt is enabled (IE[SPIIE] = 1), an interrupt is generated when any of the three following flag bits are set to "1".

Note: These flag bits can be cleared to "0" by software only.

1. SPI interrupt flag SPI_CR1[SPIIF] is set to "1" each time after the byte is transferred. It applies to all SPI modes.
2. 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. It applies to all SPI modes.
3. 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.

8.2.5 Serial Clock Timing

Four combinations of serial clock phase and idle 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).

Note: The timing relationship between SCL and SDA in master mode is shown in Figure 8-5, and that in slave mode is shown in Figure 8-6 and Figure 8-7.

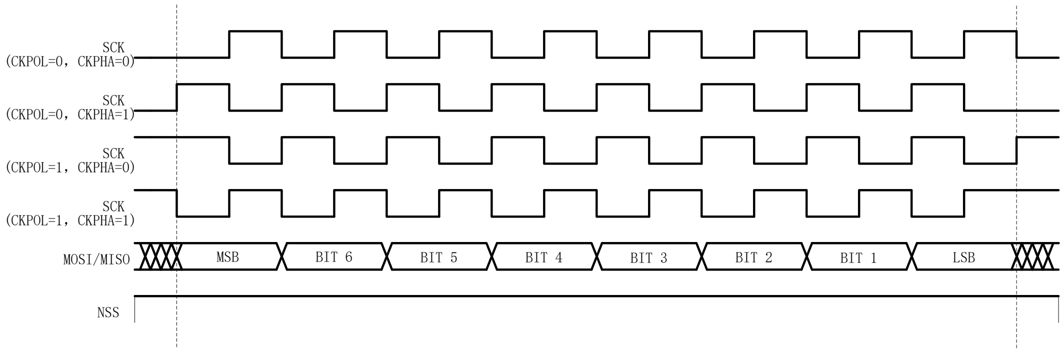


Figure 8-5 SDA/SCL Line Timing Diagram in Master Mode

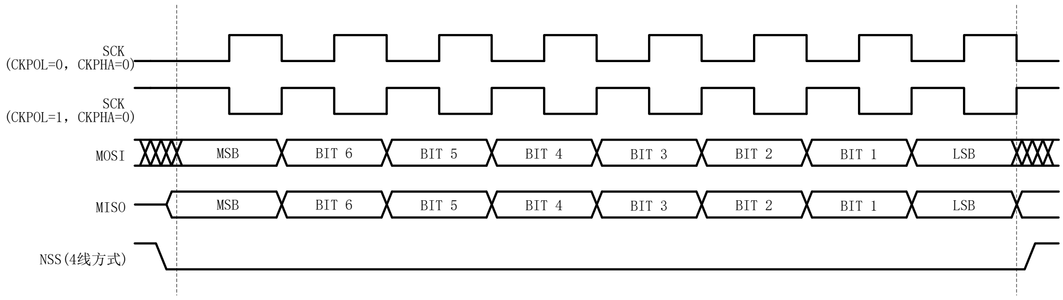


Figure 8-6 SDA/SCL Line Timing Diagram in Slave Mode (SPI_CR0[CPHA] = 0)

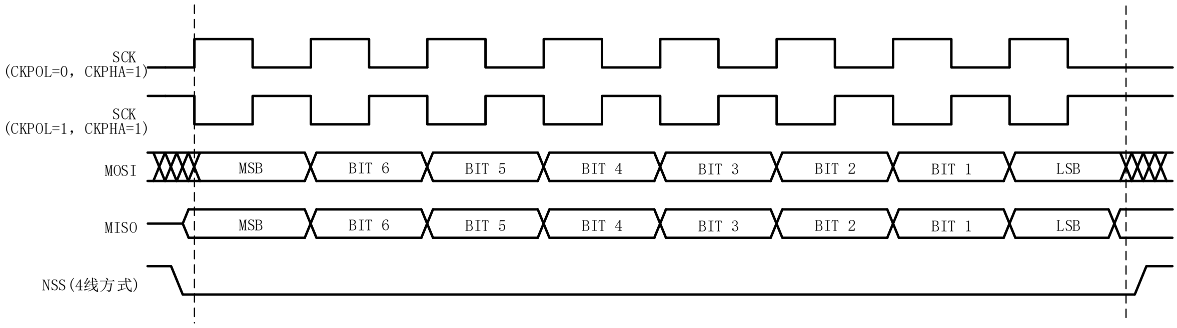


Figure 8-7 SDA/SCL Line Timing Diagram in Slave Mode (SPI_CR0[CPHA] = 1)

8.3 SPI Registers

8.3.1 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	1	0	1	1

Bit	Name	Description
[7]	SPIBSY	This bit is set to “1” when data is being transferred via SPI.
[6]	SPIMS	Master/Slave Mode Selection 0: Slave 1: Master
[5]	CPHA	SPI Clock Phase 0: Collect data at the first edge of SCK 1: Collect data at the second edge of SCK
[4]	CPOL	SPI Clock Idle Polarity 0: Low level in idle state. 1: High level in idle state.
[3]	SLVSEL	This bit is set to “1” when the filtered signal of NSS pin is low, indicating that SPI device is selected as slave. When the filtered signal of NSS pin is high, this bit is cleared to “0”, indicating that SPI device is not selected as slave. This bit does not indicate instant value of the NSS pin, but the de-noising signal input by the pin.
[2]	NSSIN	Real-time signal of NSS pin, unfiltered.
[1]	SRMT	Shift Register Empty Flag (valid only in Slave Mode) This bit is set to “0” after data has been shifted out of the Transit Buffer into the shift register or SCLK changes, and is set to “0” when there is no data in the shift register or transmit and receive buffer. Note: SPI_CR0[SRMT] = 1 in Master Mode.
[0]	RXBMT	Receive Buffer Empty Flag (valid only in Slave Mode) This bit is set to “0” when new data in the receive buffer has not been read, and is set to “1” when data has been read and there is no new data in the receive buffer. Note: SPI_CR0[RXBMT] = 1 in Master Mode.

Note: Clock phase and idle 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.

8.3.2 SPI_CR1 (0x4031)

Bit	7	6	5	4	3	2	1	0
Name	SPIIF	WCOL	MODF	RXOVR	NSSMOD		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. It can be cleared to “0” by software only.
[6]	WCOL	Write Conflict Interrupt Flag When SPI_CR1[TXBMT] is “0”, a write to SPI_DR sets this bit to “1”. This bit can be cleared to “0” by software only.
[5]	MODF	Master Mode Fault Interrupt Flag This bit is set to “1” when a master mode conflict is detected (SPI_CR0[NSSIN] = 0, SPI_CR1[SPIMS] = 1 and SPI_CR1[NSSMOD] = 01). This bit can be cleared to “0” by software only.
[4]	RXOVR	Receive Overflow Interrupt Flag (Slave Mode only) This bit is set to “1” by hardware (and generates a SPI interrupt) when the Receive Buffer still holds unread data from a previous transfer and the last bit of the current transfer has been shifted into the SPI shift register. This bit cannot be clear to “0” automatically by hardware, and can be cleared to “0” by software only.
[3:2]	NSSMOD	SPI Mode Selection 00: 3-Wire Slave or 3-wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave. NSS pin is configured as an input. 1X: 4-Wire Single-Master Mode. NSS pin is configured as output and outputs SPI_CR1[2] value.
[1]	TXBMT	Transmit Buffer Empty Flag This bit is cleared to “0” when new data is written to the Transit Buffer. It is set to “1” when the data in the Transit 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 data 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

8.3.3 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) Example: If baud rate = 2400kHz, then SPI_CLK = (24M/2/2400k) - 1 = 4, i.e. 0x04. Note: When MDU 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 data transmitted from the slave SPI.

8.3.4 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 Register SPI_DR Register is used to transmit and receive SPI data. In master mode, when the date is written to SPI_DR, it’s transferred to Transit Buffer for the transmission. When SPI_DR is read, the data held in Receive Buffer is obtained.

9 UART

9.1 UART Introduction

UART is a full-duplex or half-duplex serial data exchange interface as shown in Figure 9-1. The baud rate is configurable and supports DMA transmission. Figure 9-2 depicts the UART timing.

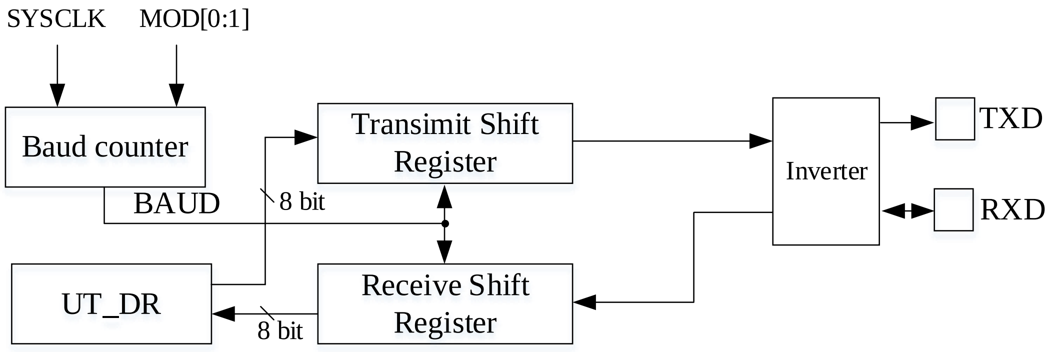


Figure 9-1 UART Block Diagram

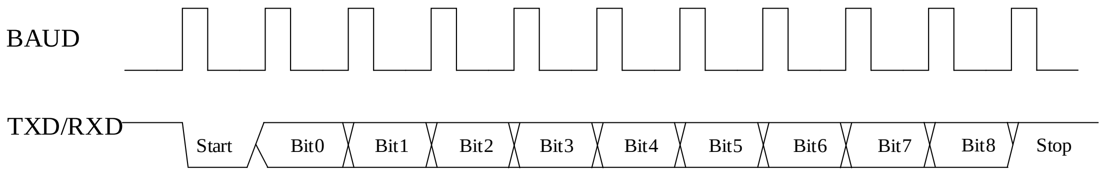


Figure 9-2 UART Timing Diagram

9.2 UART Operations

The corresponding registers shall be enabled before using UART feature. See 22.3.12 PH_SEL (0x404C) (bits [6] and [5]) for details.

9.2.1 UART Mode0

UART1 mode0, or shifting mode, is mainly used to expand the IO ports. In this mode, TXD pin is configured as clock output, and RXD data bus. The clock frequency is set at SYS_CLK/12, and the data is sent starting from the least significant bit. UT_CR[REN] decides the data is received or sent. When UT_CR[REN] = 0, the data is sent, and when UT_CR[REN] = 1, the data is received.

Sending Data: Write the data to UT_DR and resent UT_CR[TI]. TXD pin outputs shift pulse, and RXD pin sends the data held by UT_DR. The output clock frequency is SYS_CLK/12. 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]. TXD pin outputs shift pulse, and RXD pin receives the data. The shift pulse frequency is SYS_CLK/12. After the data is received, UT_CR[RI] is set to “1” and UT_DR is read to obtain the data.

9.2.2 UART 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 pin 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 pin. After the data is received, UT_CR[RB8] and UT_CR[RI] are set to "1" and UT_DR is read to obtain the data.

9.2.3 UART Mode2

UART1 mode2 works in full/half duplex mode. TXD pin is configured as an output (Transmit Data Bus), and RXD pin as an input (Receive Data Bus). It uses a total of 11 bits, namely 1-bit start, 9-bit data (UT_DR + UT_CR[RB8]/ UT_CR[TB8]) and 1-bit stop, to receive or transmit data. The baud rate is configured by UT_BAUD[UT_BAUD].

Sending Data: Write the data to UT_DR, set UT_CR[TB8] and reset 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 pin. 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 bits.

9.2.4 UART Mode3

The operations are the same as those for UART mode 2, and the baud rate is different from that for UART mode1.

9.2.5 UART Interrupt Sources

After UART interrupt is enabled (IE[ES0] = 1), an interrupt is generated when any of the two following flag bits are set to “1”.

Note: These flag bits can be cleared to “0” by software only.

- After UART sends the data (8-bit data for UART mode0 and mode1 and 9-bit data for UART mode2 and mode3), UT_CR[TI] is set to “1” by hardware.
- After UART receives the data and STOP, UT_CR[RI] is set to “1” by hardware.

9.3 UART Registers

9.3.1 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: UART Mode0——Shift Register Baud rate: SYS_CLK/12 01: UART Mode1——8-bit UART Baud rate: SYS_CLK / (16 /(1 + UT_BAUD[BAUD_SEL])) / (UT_BAUD + 1) 10: UART Mode2——9-bit UART Baud rate: SYS_CLK/(32 – 16*UT_BAUD[BAUD_SEL]) 11: UART Mode3——9-bit UART Baud rate: SYS_CLK /(16 / (1 + UT_BAUD[BAUD_SEL])) / (UT_BAUD + 1)
[5]	SM2	0: Single-device Communication 1: Multi-device Communication
[4]	REN	UART Receive Enable 0: Disable 1: Enable. This bit can be cleared to “0” by software only.
[3]	TB8	Bit9 of the sent data in UART Mode2 and UART Mode3. This bit can be cleared by hardware as required.
[2]	RB8	Bit9 of the received data in UART Mode2 and UART Mode3. If the bit SM2 is set as “0”, it serves as the STOP bit. It does not work in UART Mode0.
[1]	TI	Data Sending Completed Interrupt Flag. This bit is set to “1” after the data is sent, and can be cleared to “0” by software only.
[0]	RI	Data Receiving Completed Interrupt Flag. This bit is set to “1” after the data is received, and can be cleared to “0” by software only.

9.3.2 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

9.3.3 UT_BAUD (0x9A,0x9B)

UT_BAUDH(0x9B)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	BAUD_SEL	RSV			UT_BAUDH[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	UT_BAUDL[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
[14:12]	RSV	Reserved
[11:0]	UT_BAUD	Baud rate setting in UART Mode1 and UART Mode3

10 MDU

Multiply and Division Unit (MDU) is a built-in multiply and division co-processor for 16bit×16bit multiplication and 32bit/16bit division. It supports both unsigned and signed multiplication and unsigned division. MDU mode is determined by register MD_CR.

Multiply Mode: MD_MC3 ~0 = {MD_MAH and MD_MAL} * {MD_MBH and MD_MBL}

Note: Since MD_MC3 ~ 2 multiplexes MD_MAH and MD_MAL, and MD_MC1 ~ 0 multiplexes MD_MBH and MD_MBL, multiplicand and multiplier are written to MD_MAH and MD_MAL, and MD_MBH and MD_MBL respectively, and the result is accessed by reading the register.

Division Mode: MD_DC3 ~ 0, MD_DD1 ~ 0 = MD_DA3 ~ 0 / MD_DB1 ~ 0

The quotient is saved in MD_DC3 ~ 0 and the remainder is saved in MD_DD1 ~ 0.

Note: Since MD_DC3 ~ 0 multiplexes MD_DA3 ~ 0 and MD_DD1 ~ 0 multiplexes MD_DB1 ~ 0, dividend and divisor are written to MD_DA3 ~ 0 and MD_DB1 ~ 0 respectively, and the quotient and the remainder are accessed by reading the corresponding registers.

One multiplication process takes about one clock cycle, and the result can be shifted to the right according to MDU_CR[ALIGN]. One division process takes about 16 clock cycles. MDU_CR[DIVSTA] shall be configured to start the division. The software determines whether the division is completed by MDU_CR[DIVDONE].

10.1 Multiplication Configurations

1. Set MD_CR[MDSN] according to the multiplication type. It is set to 0 for unsigned multiplication, and 1 for signed multiplication. The right-shift bit of the result is set by MD_CR[ALIGN];
2. Write multiplicand to MD_MA, and multiplier to MD_MB;
3. Read MD_MA to obtain 16 high-order bits of the product, and MD_MB to obtain 16 low-order bits of the product
4. Start from step 2 if multiplication type and right-shift bit are the same.

10.2 Division Configurations

1. Write dividend to MD_DA, and divisor to MD_DB;
2. Write “1” to MDU_CR[DIVSTA] to start 32-bit/16-bit division;
3. The division is completed after about 16 clock cycles. MDU_CR[DIVDONE] = 1 means the division process has been completed, otherwise, you have to wait for the result;
4. Read MD_DA to obtain the quotient, and MD_DB to obtain the remainder.

10.3 Important Matters

- ■ If “0” is written to MD_DB as the divisor, MDU produces MDU_CR[DIVERR] flag bit and keeps it stays at “1” until a non-zero divisor is written;

- The quotient and the remainder are uncertain or incorrect when the divider of MDU module is working (MDU_CR[DIVDONE] = 0). They are stable and correct only when division process is completed, that is the divider is idle.
- When the divider of MDU module is working (MDU_CR[DIVDONE] = 0), changing the value of divisor or division will not affect the final result, unless MD_CR[DIVSTA] is configured to “1” again to restart a new division process.
- Because data input register of the multiplier or divider in MDU module is only one level deep, interrupts can change the result. For example, an interrupt is generated after the multiplicand is written to MD_MA but before the multiplier is written to MD_MB. The interrupt requires the multiplier in the service routine. In this case, after the interrupt is implemented, MD_MA is changed and the multiplication result is wrong. Therefore, software developers shall take proper measures to avoid such situation.

10.4 MDU Registers

10.4.1 MDU_CR (0xC1)

Bit	7	6	5	4	3	2	1	0
Name	DIVDONE	DIVERR	RSV		ALIGN		MDSN	DIVSTA
Type	R	R	-	-	R/W	R/W	R/W	R/W
Reset	1	0	-	-	0	0	0	0

Bit	Name	Description
[7]	DIVDONE	0: The division is in progress. 1: The division is completed.
[6]	DIVERR	0: Last division is correct (a non-zero divisor is written). 1: Last division is wrong (“0” is written as the divisor).
[5:4]	RSV	Reserved
[3:2]	ALIGN	Right-shift Bit for the Result of Multiplication 00: Right shift is not supported. 01: The result is right shifted by 8 bits. 10: The result is right shifted by 12 bits. 11: The result is right shifted by 15 bits.
[1]	MDSN	Multiplication Type Selection 0: Unsigned Multiplication 1: Signed Multiplication
[0]	DIVSTA	Division Start Bit. This bit is valid only for division and is set to “1” by software during division. It is automatically cleared to “0” by hardware after the division is completed. 0: The divider does not start. 1: 32-bit division starts.

10.4.2 MD_MBL (0xCA)

Bit	7	6	5	4	3	2	1	0
Name	MD_MBL							
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]	MD_MBL	Multiplier bit [7:0] (write only) or product bit [7:0] (read only)

10.4.3 MD_MBH (0xCB)

Bit	7	6	5	4	3	2	1	0
Name	MD_MBH							
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]	MD_MBH	Multiplier bit [15: 8] (write only) or product bit [15: 8] (read only)

10.4.4 MD_MAL (0xC2)

Bit	7	6	5	4	3	2	1	0
Name	MD_MAL							
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]	MD_MAL	Multiplicand bit [7:0] (write only) or product bit [23:16] (read only)

10.4.5 MD_MAH (0xC3)

Bit	7	6	5	4	3	2	1	0
Name	MD_MAH							
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]	MD_MAH	Multiplicand bit [15:8] (write only) or product bit [31:24] (read only)

10.4.6 MD_DA0 (0xC4)

Bit	7	6	5	4	3	2	1	0
Name	MD_DA0							
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]	MD_DA0	Dividend bit [7:0] (write only) or quotient bit [7:0] (read only)

10.4.7 MD_DA1 (0xC5)

Bit	7	6	5	4	3	2	1	0
Name	MD_DA1							
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]	MD_DA1	Dividend bit [15:8] (write only) or quotient bit [15:8] (read only)

10.4.8 MD_DA2 (0xC6)

Bit	7	6	5	4	3	2	1	0
Name	MD_DA2							
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]	MD_DA2	Dividend bit [23:16] (write only) or quotient bit [23:16] (read only)

10.4.9 MD_DA3 (0xC7)

Bit	7	6	5	4	3	2	1	0
Name	MD_DA3							
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]	MD_DA3	Dividend bit [31:24] (write only) or quotient bit [31:24] (read only)

10.4.10 MD_DB0 (0xCC)

Bit	7	6	5	4	3	2	1	0
Name	MD_DB0							
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]	MD_DB0	Divisor bit [7:0] (write only) or remainder bit [7:0] (read only)

10.4.11 MD_DB1 (0xCD)

Bit	7	6	5	4	3	2	1	0
Name	MD_DB1							
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]	MD_DB1	Divisor bit [15:8] (write only) or remainder bit [15:8] (read only)

11 PI

11.1 PI Introduction

PI regulator is a linear controller, where the output is generated by linear combination of error proportional, integral and differential actions, and then implemented by an actuator. In motor control system, it is used to for speed and position control.

PI algorithm:

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

Where,

U_k: Output for round k of calculation

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

E_k: Deviation for round k of input

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

K_p and K_i: Proportional (P) and integral (I) coefficients of regulator

The maximum U_k is represented as PIx_UKMAX (x=0 ~ 3) and the minimum value as PIx_UKMIN

(Uk_min < U(k) < Uk_max)

11.2 PI Operations

1. Configuring PI_LPF_CR[PISTA] = 1 to enable PI. After four clock cycles, the calculation is completed and PI_UK is updated.
2. Q12 (default) or Q15 can be selected for PI parameters. It means data format of PI_KP and PI_KI is Q12 and that of other registers are Q15.
3. U_{K-1} and E_{K-1} default to the previous U_k and E_k. You can write the data to PI_EK register and start PI to change E_{K-1}, or write the data to PI_UK register to change U_{k-1}.

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

```
PI_EK = X;                          //Initialize E(k-1)
SetBit(PI_LPF_CR , PISTA);         //Start PI
_nop_();
_nop_();
_nop_();
_nop_();                            //Wait for PI regulator to complete operations
PI_UK = X;                          //Initialize U(k-1)
```

11.3 PI Registers

11.3.1 PI_LPF_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2SS	RSV				PIRANGE	PISTA	LPFSTA
Type	R/W	-	-	-	-	R/W	R/W	R/W
Reset	0	-	-	-	-	0	0	0

Bit	Name	Description
[7]	T2SS	Input Mode Selection of TIM2 Step Motor 0: Direction + Pulse Input Mode. P1.0 for direction input, and P0.7 for pulse input 1: Bidirectional Pulse Input Mode. P1.0 for backward pulse input, and P0.7 for forward pulse input
[6:3]	RSV	Reserved
[2]	PIRANGE	Data Format of PI Parameters 0: Q12. The value range of KP and KI (-32768,32767) corresponds to range (-8,8). 1: Q15. The value range of KP and KI (-32768,32767) corresponds to range (-1,1).
[1]	PISTA	PI Regulator Enable Bit. It is set to "1" by software, and can be automatically cleared to "0" by hardware. 0: Disable 1: Enable
[0]	LPFSTA	LPF Enable Bit. It is set to "1" by software, and can be automatically cleared to "0" by hardware. 0: Disable 1: Enable

11.3.2 PI_EK (0xEA, 0xEB)

PI_EKH(0xEB)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI_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
PI_EKL(0xEA)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI_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]	PI_EK	Input Deviation; Range (-32768, 32767)

11.3.3 PI_UK (0xEC, 0xED)

PI_UKH(0xED)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI_UK[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
PI_UKL(0xEC)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI_UK[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]	PI_UK	Calculation result of PI regulator Range (-32768, 32767)

11.3.4 PI_KP (0xEE, 0xEF)

PI_KPH(0xEF)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI_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
PI_KPL(0xEE)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI_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]	PI_KP	KP Coefficient Range (-32768, 32767)

11.3.5 PI_KI (0xF2, 0xF3)

PI_KIH(0xF3)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI_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
PI_KIL(0xF2)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI_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]	PI_KI	KI Coefficient Range (-32768,32767)

11.3.6 PI_UKMAX (0xF4, 0xF5)

PI_UKMAXH(0xF5)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI_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
PI_UKMAXL(0xF4)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI_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]	PI_UKMAX	UK Maximum Value; Range (-32768,32767)

11.3.7 PI_UKMIN (0xF6, 0xF7)

PI_UKMINH(0xF7)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI_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
PI_UKMINL(0xF6)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI_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]	PI_UKMIN	UK Minimum Value Range (-32768,32767)

12 LPF

A low-pass filter (LPF) is a filter circuit that passes low-frequency signals while attenuates high-frequency signals.

The calculation formula of LPF:

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

Where,

Y_k: Filtered value

Y_{k-1}: Previous filtered output

K: Filter coefficient

X_k: Value to be filtered

12.1 LPF Operations

1. Configuring PI_LPF_CR[LPFSTA] = 1 to enable LPF. After four clock cycles, LPF_Y is updated.
2. Y_{k-1} default to the previous Y_k. You can write the data to LPF_Y register to change Y_{k-1}.

12.2 LPF Registers

12.2.1 PI_LPF_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2SS	RSV				PIRANGE	PISTA	LPFSTA
Type	R/W	-	-	-	-	R/W	R/W	R/W
Reset	0	-	-	-	-	0	0	0

Bit	Name	Description
[7:1]		See PI_LPF_CR (0xF9) for details.
[0]	LPFSTA	LPF Enable Bit. It is set to “1” by software, and can be automatically cleared to “0” by hardware. 0: Disable 1: Enable

12.2.2 LPF_K (0xDD)

Bit	7	6	5	4	3	2	1	0
Name	LPF_K							
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]	LPF_K	LPF Coefficient The value range (-128,127) corresponds to range (-1,1).

12.2.3 LPF_X (0xDE, 0xDF)

LPF_XH(0xDF)								
Bit	7	6	5	4	3	2	1	0
Name	LPF_X[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

LPF_XL(0xDE)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF_X[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]	LPF_X	Input Value Range (-32768, 32767)						

12.2.4 LPF_Y (0xE6, 0xE7)

LPF_YH(0xE7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF_Y[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
LPF_YL(0xE6)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF_Y[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]	LPF_Y	Output Value Range (-32768, 32767)						

13 SMDU

SMDU is a built-in computing co-processor that assists the CPU in processing complex operations efficiently. It supports multiplication, division, trigonometric operation, LPF operation and PI operation. SMDU module can be invoked in different interrupt services and master programs, and the results are independent from each other.

13.1 SMDU Features

The SMDU module has the following 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 left by one-bit)
  - 16-bit unsigned multiplication
  - 32-bit/16-bit unsigned division
  - SMDU LPF
  - Coordinate transformation (SIN/COS)
  - Arctangent (ATAN)
  - PI

13.2 SMDU Instructions

13.2.1 SMDU Operations

SMDU is operated as follows.

1. Set TIM234_CTRL[MDU_EN_N] to “1” to enable SMDU module. After that, MDU, LPF and PI modules are disabled;
2. Select an operation mode by MDU_CR[MDUMOD];
3. Write the data to the associated computing units, select computing unit of the SMDU module by MDU_CR[MDUSTA], and start SMDU computing;
4. Wait for MDU_CR[MDUBUSY] to be cleared to “0” by hardware.

Note: When using SMDU, ensure that the computing mode and other data have been written before configuring MDU_CR[MDUSTA].

13.2.2 16-bit Signed Multiplication with the Result Shifted Left by One-bit

When MDU_CR[MDUMOD] = 000, SMDU module works in the 16-bit signed multiplication mode with the result shifted left by one-bit. As shown in Table 13-1, after 16-bit signed data is written to

MULx_MA and MULx_MB as multiplicand and multiplier respectively, 32-bit signed data is obtained by the product shifting left by one bit. The result is accessed by reading MULx_MC register.

Table 13-1 Register Definitions in 16-bit Signed Multiplication Mode with the Result Shifted Left by One-bit

Data Register	Input	Output
MULx_MA	Multiplicand	-
MULx_MB	Multiplier	-
MULx_MC	-	Product

13.2.3 16-bit Signed Multiplication

When MDU_CR[MDUMOD] = 001, SMDU module works in the 16-bit signed multiplication mode. As shown in Table 13-2, 32-bit signed data is obtained after 16-bit signed data is written to MULx_MA and MULx_MB as multiplicand and multiplier respectively. The result is accessed by reading MULx_MC register.

Table 13-2 Register Definitions in 16-bit Signed Multiplication Mode

Data Register	Input	Output
MULx_MA	Multiplicand	-
MULx_MB	Multiplier	-
MULx_MC	-	Product

13.2.4 16-bit Unsigned Multiplication

When MDU_CR[MDUMOD] = 010, SMDU module works in the 16-bit unsigned multiplication mode. As shown in Table 13-3, 32-bit unsigned data is obtained after 16-bit unsigned data is written to MULx_MA and MULx_MB as multiplicand and multiplier respectively. The result is accessed by reading MULx_MC register.

Table 13-3 Register Definitions in 16-bit Unsigned Multiplication Mode

Data Register	Input	Output
MULx_MA	Multiplicand	-
MULx_MB	Multiplier	-
MULx_MC	-	Product

13.2.5 32-bit/16-bit Unsigned Division

When MDU_CR[MDUMOD] = 011, SMDU module works in the 32-bit/16-bit unsigned division mode. As shown in Table 13-4, 32-bit unsigned quotient and 16-bit unsigned remainder is obtained after 32-bit unsigned dividend and a 16-bit unsigned divisor are written to DIVx_DA and DIVx_DB registers respectively. The quotient and remainder are accessed by reading DIVx_DQ and DIVx_DR registers respectively.

Table 13-4 Register Definitions in the Unsigned Division Mode

Data Register	Input	Output
DIVx_DA	Dividend	-
DIVx_DB	Divisor	-

Data Register	Input	Output
DIVx_DQ	-	Quotient
DIVx_DR	-	Remainder

13.2.6 SMDU LPF

When MDU_CR[MDUMOD] = 110, SMDU module works in LPF mode.

SMDU LPF has the same function as LPF. LPF writes or reads data in SFR sector, while SMDU LPF in XSFR sector.

As shown in Table 13-5, *K* is 16-bit signed data. Y_k is obtained after Y_{k-1} is written to LPFx_Y, *K* to LPFx_K and X_k to LPFx_X, and is accessed by reading LPFx_Y.

Table 13-5 Register Definitions in LPF Mode

Data Register	Input	Output
LPFx_X	X_k	-
LPFx_K	K	-
LPFx_Y	Y_{k-1}	Y_k

13.2.7 Coordinate Transformation (SIN/COS)

When MDU_CR[MDUMOD] = 100, SMDU module works in Coordinate Transformation mode. As shown in Figure 13-1, 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:

$$cos_o = cos_i \times \cos\theta$$
$$sin_o = cos_i \times \sin\theta$$

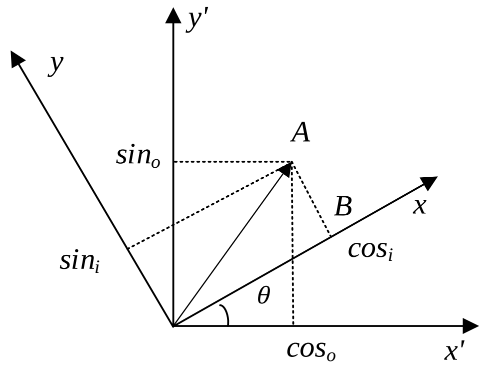


Figure 13-1 Coordinate Transformation

As shown in Table 13-6, cos_i, sin_i, θ, cos_o and sin_o are all 16-bit signed data. cos_i is written to SCATx_COS, sin_i to SCATx_SIN and θ to SCATx_THE to calculate cos_o and sin_o. The results cos_o and sin_o are accessed by reading SCATx_RES1 and SCATx_RES2 respectively.

Table 13-6 Register Definitions in the Coordinate Transformation Mode

Data Register	Input	Output
SCATx_COS	cos_i	-
SCATx_SIN	sin_i	-
SCATx_THE	θ	-
SCATx_RES1	-	cos_o
SCATx_RES2	-	sin_o

13.2.8 Arctangent

When MDU_CR[MDUMOD] = 101, SMDU module works in arctangent (ATAN) mode.

ATAN calculates the amplitude and angle of a vector based on sine and cosine inputs. 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,

$Usin\theta$: Sin component of the vector

$Ucos\theta$: Cosine component of the vector

θ: Calculated vector angle

U: Calculated vector amplitude

As shown in Table 13-7, $Ucos\theta$ and $Usin\theta$, U and θ are 16-bit signed data. $Ucos_\theta$ is written to SCATx_COS and $Usin_\theta$ to SCATx_SIN to calculate U and θ. U and θ are accessed by reading SCATx_RES1 and SCATx_RES2 respectively.

Table 13-7 Register Definitions in ATAN Mode

Data Register	Input	Output
SCATx_COS	$Ucos\theta$	-
SCATx_SIN	$Usin\theta$	-
SCATx_RES1	-	U
SCATx_RES2	-	θ

13.2.9 PI

13.2.9.1 PI Introduction

PI regulator is a linear controller, where the output is generated by linear combination of error proportional, integral and differential actions, and then implemented by an actuator. In motor control system, it is used to for speed and position control.

PI algorithm:

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

Where,

U_k: Output for round k of calculation

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

E_k: Deviation for round k of input

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

K_p and K_i: Proportional (P) and integral (I) coefficients of regulator

The maximum U_k is represented as PIx_UKMAX (x=0 ~ 3) and the minimum value as PIx_UKMIN.

13.2.9.2 PI Features

- Parameter range is configurable
- Support multiple invocations but not with nested interrupt
- Produce a 32-bit result PIx_UK
- Results are read after Busy Flag is reset to “0”.

13.2.9.3 PI Operations

1. Initialize SMDU before operations, and configure K_p, K_i and the maximum and minimum values of U_K;
2. Set MDU_CR[MDUMOD] to 111, and then configure MDU_CR[MDUSTA] bit to select the desired computing unit and start PI computing. At this time, busy flag MDU_CR[MDUBUSY] is automatically set to “1”;
3. Read MDU_CR[MDUBUSY] bit by software. When this bit is 0, it 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 are Q15.
- PIx_UK and PIx_EK1 values default to the previous calculated U_K and $E_{K.}$ The related values change after PIx_EK1 and PIx_UK are written.

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

```
PIx_KP = KP;                //Initialize Kp
PIx_KI = KI;                //Initialize Ki
PIx_UKMAX = UKMAX;          //Initialize maximum output
PIx_UKMIN = UKMIN;          //Initialize minimum output
PIx_EK1 = X;                //Initialize $E_{k-1}$
PIx_UKH = Y1;               //Initialize 16 high-order bits of $U_{k-1}$
PIx_UKL = Y2;               //Initialize 16 low-order bits of $U_{k-1}$
```

13.3 SMDU Registers

13.3.1 MDU_CR (0xC1)

MDU_CR(0xC1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MDUBSY	MDUSTA3	MDUSTA2	MDUSTA1	MDUSTA0	MDUMOD2	MDUMOD1	MDUMOD0
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

Bit	Name	Description
[7]	MDUBSY	After SMDU module is enabled, this bit is used to indicate operating status of the module. 0: Idle 1: Busy
[6]	MDUSTA3	After SMDU module is enabled, this bit is used to enable or disable computing unit 3. 0: Disable 1: Enable
[5]	MDUSTA2	After SMDU module is enabled, this bit is used to enable or disable computing unit 2. 0: Disable 1: Enable
[4]	MDUSTA1	After SMDU module is enabled, this bit is used to enable or disable computing unit 1. 0: Disable 1: Enable
[3]	MDUSTA0	After SMDU module is enabled, this bit is used to enable or disable computing unit 0. 0: Disable 1: Enable
[2:0]	MDUMOD [2:0]	After SMDU module is enabled, this bit is used to select operating mode of the module. 000: Signed Multiplication (the result shifted left by one-bit) 001: Signed Multiplication 010: Unsigned Multiplication 011: 32-bit/16-bit Unsigned Division 100: Coordinate Transformation (SIN/COS) 101: ATAN 110: LPF 111: PI

Note: Configuring TIM234_CTRL[6] enables or disables SMDU module.

13.3.2 TIM234_CTRL(0x40f1)

TIM234_CTRL(0x40f1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RSV	MDU_EN_N	TIM2_FAST_DIR	TIM2_DR_SEL	TIM4_RCTRL	TIM3_RCTRL	TIM2_RCTRL	TIM3_48M
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]	MDU_EN_N	When this bit is set to “1”, SMDU module is enabled and MDU module is disabled. 0: SMDU module is disabled. 1: SMDU module is enabled.
[5]	TIM2_FAST_DIR	After this bit is enabled in TIM2 QEP mode, SMDU module can quickly identify the direction. 0: Disable 1: Enable
[4]	TIM2_DR_SEL	After this bit is enabled in TIM2 QEP mode, the timer is reset to 0 or TIM2__DR when it reaches to TIM2__DR or 0. 0: Disable 1: Enable
[3]	TIM4_RCTRL	After this bit is enabled, DR = 0/0xffff when TIM4__CNTR overflows. 0: Disable 1: Enable
[2]	TIM3_RCTRL	After this bit is enabled, DR = 0/0xffff when TIM3_ARR overflows. 0: Disable 1: Enable
[1]	TIM2_RCTRL	After this bit is enabled, DR = 0/0xffff when TIM2_ARR overflows. 0: Disable 1: Enable
[0]	TIM3_48M	TIM3_48M Input Capture Enable 0: Disable 1: Enable

13.3.3 MUL0_MA (0x03C8, 0x03C9)

MUL0_MAH(0x03C8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL0_MA[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
MUL0_MAL(0x03C9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL0_MA[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]	MUL0_MA	Data register A of MUL0; Multiplicand of the multiplication

13.3.4 MUL0_MB (0x03CA, 0x03CB)

MUL0_MBH(0x03CA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL0_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
MUL0_MBL(0x03CB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL0_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]	MUL0_MB	Data register B of MUL0; Multiplier of the multiplication

13.3.5 MUL0_MC (0x03CC, 0x03CD, 0x03CE, 0x03CF)

MUL0_MCHH(0x03CC)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	MUL0_MC[31:24]							
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
MUL0_MCHL(0x03CD)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	MUL0_MC[13:16]							
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
MUL0_MCLH(0x03CE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL0_MC[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
MUL0_MCLL(0x03CF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL0_MC[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**
[31:0]	MUL0_MC	Product of MUL0. The 16 high-order bits of the data are held by MUL0_MCH and the 16 low-order bits by MUL0_MCL.

13.3.6 MUL1_MA (0x03C0, 0x03C1)

MUL1_MAH(0x03C0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL1_MA[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
MUL1_MAL(0x03C1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL1_MA[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]	MUL1_MA	Data register A of MUL1; Multiplicand of the multiplication

13.3.7 MUL1_MB (0x03C2, 0x03C3)

MUL1_MBH(0x03C2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL1_MB[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
MUL1_MBL(0x03C3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL1_MB[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]	MUL1_MB	Data register B of MUL1; Multiplier of the multiplication

13.3.8 MUL1_MC (0x03C4, 0x03C5, 0x03C6, 0x03C7)

MUL1_MCHH(0x03C4)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	MUL1_MC[31:24]							
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
MUL1_MCHL(0x03C5)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	MUL1_MC[23:16]							
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
MUL1_MCLH(0x03C6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL1_MC[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
MUL1_MCLL(0x03C7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL1_MC[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**
[31:0]	MUL1_MC	Product of MUL1. The 16 high-order bits of the data are held by MUL1_MCH and the 16 low-order bits by MUL1_MCL.

13.3.9 MUL2_MA (0x0370, 0x0371)

MUL2_MAH(0x0370)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL2_MA[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
MUL2_MAL(0x0371)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL2_MA[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]	MUL2_MA	Data register A of MUL2; Multiplicand of the multiplication

13.3.10 MUL2_MB (0x0372, 0x0373)

MUL2_MBH(0x0372)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL2_MB[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
MUL2_MBL(0x373)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL2_MB[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]	MUL2_MB	Data register B of MUL2; Multiplier of the multiplication

13.3.11 MUL2_MC (0x0374, 0x0375, 0x0376, 0x0377)

MUL2_MCHH(0x0374)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	MUL2_MC[31:24]							
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
MUL2_MCHL(0x0375)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	MUL2_MC[23:16]							
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
MUL2_MCLH(0x0376)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL2_MC[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
MUL2_MCLL(0x0377)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL2_MC[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**
[31:0]	MUL2_MC	Product of Register MUL2. The 16 high-order bits of the data are held by MUL2_MCH and the 16 low-order bits by MUL2_MCL.

13.3.12 MUL3_MA (0x0368, 0x0369)

MUL3_MAH(0x0368)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL3_MA[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
MUL3_MAL(0x0369)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL3_MA[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]	MUL3_MA	Data register A of MUL3; Multiplicand of the multiplication

13.3.13 MUL3_MB (0x036A, 0x036B)

MUL3_MBH(0x036A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL3_MB[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
MUL3_MBL(0x036B)								
Bit	**7**	6	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL3_MB[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]	MUL3_MB	Data register B of MUL3; Multiplier of the multiplication

13.3.14 MUL3_MC (0x036C, 0x036D, 0x036E, 0x036F)

MUL3_MCHH(0x036C)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	MUL3_MC[31:24]							
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
MUL3_MCHL(0x036D)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	MUL3_MC[23:16]							
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
MUL3_MCLH(0x036E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL3_MC[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
MUL3_MCLL(0x036F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL3_MC[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

Bit	**Name**	**Description**
[31:0]	MUL3_MC	Product of MUL3. The 16 high-order bits of the data are held by MUL3_MCH and the 16 low-order bits by MUL3_MCL.

13.3.15 DIV0_DA (0x03B4, 0x03B5, 0x03B6, 0x03B7)

DIV0_DAHH(0x03B4)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	DIV0_DA[31:24]							
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
DIV0_DAHL(0x03B5)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	DIV0_DA[23:16]							
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
DIV0_DALH(0x03B6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV0_DA[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
DIV0_DALL(0x03B7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV0_DA[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**
[31:0]	DIV0_DA	Dividend of DIV0. The 16 high-order bits of the data are held by DIV0_DAH and the 16 low-order bits by DIV0_DAL.

13.3.16 DIV0_DB (0x03B8, 0x03B9)

DIV0_DBH(0x03B8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV0_DB[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
DIV0_DBL(0x03B9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV0_DB[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]	DIV0_DB	Data register B of DIV0; Divisor of the division

13.3.17 DIV0_DQ (0x03BA, 0x03BB, 0x03BC, 0x03BD)

DIV0_DQHH(0x03BA)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	DIV0_DQ[31:24]							
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
DIV0_DQHL(0x03BB)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	DIV0_DQ[23:16]							
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
DIV0_DQLH(0x03BC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV0_DQ[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
DIV0_DQLL(0x03BD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV0_DQ[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
[31:0]	DIV0_DQ[31:0]	Quotient of DIV0. The 16 high-order bits of the data are held by DIV0_DQH and the 16 low-order bits by DIV0_DQL.

13.3.18 DIV0_DR (0x03BE, 0x03BF)

DIV0_DRH(0x03BE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV0_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
DIV0_DRL(0x03BF)								
Name	DIV0_DR[7:0]							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**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]	DIV0_DR	Remainder of DIV0

13.3.19 DIV1_DA (0x03A8, 0x03A9, 0x03AA, 0x03AB)

DIV1_DAHH(0x03A8)								
Bit	31	30	29	28	27	26	25	24
Name	DIV1_DA[31:24]							
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
DIV1_DAHL(0x03A9)								
Bit	23	22	21	20	19	18	17	16
Name	DIV1_DA[23:16]							
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
DIV1_DALH(0x03AA)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DA[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
DIV1_DALL(0x03AB)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DA[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
[31:0]	DIV1_DA	Dividend of DIV1. The 16 high-order bits of the data are held by DIV1_DA and the 16 low-order bits by DIV1_DA.

13.3.20 DIV1_DB (0x03AC, 0x03AD)

DIV1_DBH(0x03AC)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DB[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
DIV1_DBL(0x03AD)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DB[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]	DIV1_DB	Data register B of DIV1; Divisor of the division

13.3.21 DIV1_DQ (0x03AE, 0x03AF, 0x03B0, 0x03B1)

DIV1_DQHH(0x03AE)								
Bit	31	30	29	28	27	26	25	24
Name	DIV1_DQ[31:24]							
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
DIV1_DQHL(0x03AF)								
Bit	23	22	21	20	19	18	17	16
Name	DIV1_DQ[23:16]							
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

DIV1_DQLH(0x03B0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV1_DQ[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
DIV1_DQLL(0x03B1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV1_DQ[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
[31:0]	DIV1_DQ	Quotient of DIV1. The 16 high-order bits of the data are held by DIV1_DQH and the 16 low-order bits by DIV1_DQL.

13.3.22 DIV1_DR (0x03B2, 0x03B3)

DIV1_DRH(0x03B2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV1_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
DIV1_DRL(0x03B3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV1_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]	DIV1_DR	Remainder of DIV1

13.3.23 DIV2_DA (0x035C, 0x035D, 0x035E, 0x035F)

DIV2_DAHH(0x035C)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	DIV2_DA[31:24]							
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
DIV2_DAHL(0x035D)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	DIV2_DA[23:16]							
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
DIV2_DALH(0x035E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV2_DA[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
DIV2_DALL(0x035F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV2_DA[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
[31:0]	DIV2_DA	Dividend of DIV2. The 16 high-order bits of the data are held by DIV2_DA and the 16 low-order bits by DIV2_DA.

13.3.24 DIV2_DB (0x0360, 0x0361)

DIV2_DBH(0x0360)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV2_DB[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
DIV2_DBL(0x0361)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV2_DB[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]	DIV2_DB	Data register B of DIV2; Divisor of the division

13.3.25 DIV2_DQ (0x0362, 0x0363, 0x0364, 0x0365)

DIV2_DQHH(0x0362)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	DIV2_DQ[31:24]							
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
DIV2_DQHL(0x0363)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	DIV2_DQ[23:16]							
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
DIV2_DQLH(0x0364)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV2_DQ[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
DIV2_DQLL(0x0365)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV2_DQ[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**
[31:0]	DIV2_DQ	Quotient of DIV2. The 16 high-order bits of the data are held by DIV2_DQH and the 16 low-order bits by DIV2_DQL.

13.3.26 DIV2_DR (0x0366, 0x0367)

DIV2_DRH(0x0366)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV2_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
DIV2_DRL(0x067)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV2_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]	DIV2_DR	Remainder of DIV2

13.3.27 DIV3_DA (0x0350, 0x0351, 0x0352, 0x0353)

DIV3_DAHH(0x0350)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	DIV3_DA[31:24]							
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
DIV3_DAHL(0x0351)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	DIV3_DA[23:16]							
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
DIV3_DALH(0x0352)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV3_DA[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
DIV3_DALL(0x0353)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV3_DA[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**
[31:0]	DIV3_DA	Dividend of DIV3. The 16 high-order bits of the data are held by DIV3_DAH and the 16 low-order bits by DIV3_DAL.

13.3.28 DIV3_DB (0x0354, 0x0355)

DIV3_DBH(0x0354)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV3_DB[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
DIV3_DBL(0x0355)								
Bit	**7**	6	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV3_DB[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]	DIV3_DB	Data register B of DIV3; Divisor of the division

13.3.29 DIV3_DQ (0x0356, 0x0357, 0x0358, 0x0359)

DIV3_DQHH(0x0356)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	DIV3_DQ[31:24]							
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
DIV3_DQHL(0x0357)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	DIV3_DQ[23:16]							
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

DIV3_DQLH(0x0358)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV3_DQ[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
DIV3_DQLL(0x0359)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV3_DQ[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**
[31:0]	DIV3_DQ	Quotient of DIV3. The 16 high-order bits of the data are held by DIV3_DQH and the 16 low-order bits by DIV3_DQL.

13.3.30 DIV3_DR (0x035A, 0x035B)

DIV3_DRH(0x035A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DIV3_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
DIV3_DRL(0x035B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DIV3_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]	DIV3_DR	Remainder of DIV3

13.3.31 SCAT0_COS (0x0346, 0x0347)

SCAT0_COSH(0x0346)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT0_COS[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
SCAT0_COSL(0x0347)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT0_COS[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

Bit	**Name**	**Description**
[15:0]	SCAT0_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT0

13.3.32 SCAT0_SIN (0x0348, 0x0349)

SCAT0_SINH(0x0348)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT0_SIN[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
SCAT0_SINL(0x0349)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Name	SCAT0_SIN[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]	SCAT0_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT0

13.3.33 SCAT0_THE (0x034A, 0x034B)

SCAT0_THEH(0x034A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT0_THE[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
SCAT0_THEL(0x034B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT0_THE[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]	SCAT0_THE	THE input in SIN/COS mode of computing unit SCAT0

13.3.34 SCAT0_RES1 (0x034C, 0x034D)

SCAT0_RES1H(0x034C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT0_RES1[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
SCAT0_RES1L(0x034D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT0_RES1[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]	SCAT0_RES1	COS output in SIN/COS mode of computing unit SCAT0; *U* output in ATAN mode

13.3.35 SCAT0_RES2 (0x034E, 0x034F)

SCAT0_RES2H(0x034E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT0_RES2[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
SCAT0_RES2L(0x034F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT0_RES2[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]	SCAT0_RES2	SIN output in SIN/COS mode of computing unit SCAT0; θ output in ATAN mode

13.3.36 SCAT1_COS (0x033C, 0x033D)

SCAT1_COSH(0x033C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT1_COS[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
SCAT1_COSL(0x033D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT1_COS[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]	SCAT1_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT1

13.3.37 SCAT1_SIN (0x033E, 0x033F)

SCAT1_SINH(0x033E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT1_SIN[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
SCAT1_SINL(0x033F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT1_SIN[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]	SCAT1_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT1

13.3.38 SCAT1_THE (0x0340, 0x0341)

SCAT1_THEH(0x0340)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT1_THE[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
SCAT1_THEL(0x0341)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT1_THE[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]	SCAT1_THE	THE input in SIN/COS mode of computing unit SCAT1

13.3.39 SCAT1_RES1 (0x0342, 0x0343)

SCAT1_RES1H(0x0342)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT1_RES1[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

SCAT1_RES1L(0x0343)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT1_RES1[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]	SCAT1_RES1	COS output in SIN/COS mode of computing unit SCAT1; *U* output in ATAN mode

13.3.40 SCAT1_RES2 (0x0344, 0x0345)

SCAT1_RES2H(0x0344)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT1_RES2[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
SCAT1_RES2L(0x0345)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT1_RES2[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]	SCAT1_RES2	SIN output in SIN/COS mode of computing unit SCAT1; θ output of ATAN mode

13.3.41 SCAT2_COS (0x0332, 0x0333)

SCAT2_COSH(0x0332)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT2_COS[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
SCAT2_COSL(0x0333)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT2_COS[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]	SCAT2_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT2

13.3.42 SCAT2_SIN (0x0334, 0x0335)

SCAT2_SINH(0x0334)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT2_SIN[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
SCAT2_SINL(0x0335)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT2_SIN[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]	SCAT2_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT2

13.3.43 SCAT2_THE (0x0336, 0x0337)

SCAT2_THEH(0x0336)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT2_THE[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
SCAT2_THEL(0x0337)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT2_THE[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]	SCAT2_THE	THE input in SIN/COS mode of computing unit SCAT2

13.3.44 SCAT2_RES1 (0x0338, 0x0339)

SCAT2_RES1H(0x0338)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT2_RES1[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
SCAT2_RES1L(0x0339)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT2_RES1[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]	SCAT2_RES1	COS output in SIN/COS mode of computing unit SCAT2; U output in ATAN mode

13.3.45 SCAT2_RES2 (0x033A, 0x033B)

SCAT2_RES2H(0x033A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT2_RES[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
SCAT2_RES2L(0x033B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT2_RES[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]	SCAT2_RES2	SIN output in SIN/COS mode of computing unit SCAT2; θ output in ATAN mode

13.3.46 SCAT3_COS (0x0328, 0x0329)

SCAT3_COSH(0x0328)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT3_COS[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

SCAT3_COSL(0x0329)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT3_COS[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]	SCAT3_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT3

13.3.47 SCAT3_SIN (0x032A, 0x032B)

SCAT3_SINH(0x032A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT3_SIN[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
SCAT3_SINL(0x032B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT3_SIN[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]	SCAT3_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT3

13.3.48 SCAT3_THE (0x032C, 0x032D)

SCAT3_THEH(0x032C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT3_THE[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
SCAT3_THEL(0x032D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT3_THE[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]	SCAT3_THE	THE input in SIN/COS mode of computing unit SCAT3

13.3.49 SCAT3_RES1 (0x032E, 0x032F)

SCAT3_RES1H(0x032E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT3_RES1[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
SCAT3_RES1L(0x032F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT3_RES1[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]	SCAT3_RES1	COS output in SIN/COS mode of computing unit SCAT3; *U* output in ATAN mode

13.3.50 SCAT3_RES2 (0x0330, 0x0331)

SCAT3_RES2H(0x0330)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	SCAT3_RES[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
SCAT3_RES2L(0x0331)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCAT3_RES[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]	SCAT3_RES2	SIN output in SIN/COS mode of computing unit SCAT3; θ output in ATAN mode						

13.3.51 LPF0_K (0x03F8, 0x03F9)

LPF0_KH(0x03F8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF0_K[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
LPF0_KL(0x03F9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF0_K[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]	LPF0_K[15:0]	K input of LPF0						

13.3.52 LPF0_X (0x03FA 0x03FB)

LPF0_XH(0x03FA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF0_X[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
LPF0_XL(0x03FB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF0_X[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]	LPF0_X[15:0]	X input of LPF0						

13.3.53 LPF0_Y (0x03FC, 0x03FD, 0x03FE, 0x03FF)

LPF0_YHH(0x03FC)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	LPF0_Y[31:24]							
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
LPF0_YHL(0x03FD)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	LPF0_Y[23:16]							
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

LPF0_YLH(0x03FE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF0_Y[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
LPF0_YLL(0x03FF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF0_Y[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**
[31:0]	LPF0_Y[31:0]	Input and output of the register in LPF0 Input: LPF0_Y_{k-1} Output: LPF0_Y_k

13.3.54 LPF1_K (0x03F0, 0x03F1)

LPF1_KH(0x03F0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF1_K[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
LPF1_KL(0x03F1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF1_K[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]	LPF1_K[15:0]	K input of LPF1

13.3.55 LPF1_X (0x03F2, 0x03F3)

LPF1_XH(0x03F2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF1_X[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
LPF1_XL(0x03F3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF1_X[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]	LPF1_X[15:0]	X input of LPF1

13.3.56 LPF1_Y (0x03F4, 0x03F5, 0x03F6, 0x03F7)

LPF1_YHH(0x03F4)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	LPF1_Y[31:24]							
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

LPF1_YHL(0x03F5)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	LPF1_Y[23:16]							
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
LPF1_YLH(0x03F6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF1_Y[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
LPF1_YLL(0x03F7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF1_Y[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**
[31:0]	LPF1_Y[31:0]	Input and output of the register in LPF1 Input: LPF1_Y_{k-1} Output: LPF1_Y_{k}

13.3.57 LPF2_K (0x03A0, 0x03A1)

LPF2_KH(0x03A0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF2_K[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
LPF2_KL(0x03A1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF2_K[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]	LPF2_K[15:0]	K input of LPF2

13.3.58 LPF2_X (0x03A2, 0x03A3)

LPF2_XH(0x03A2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF2_X[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
LPF2_XL(0x03A3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF2_X[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]	LPF2_X[15:0]	X input of LPF2

13.3.59 LPF2_Y (0x03A4, 0x03A5, 0x03A6, 0x03A7

LPF2_YHH(0x03A4)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	LPF2_Y[31:24]							
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
LPF2_YHL(0x03A5)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	LPF2_Y[23:16]							
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
LPF2_YLH(0x03A6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF2_Y[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
LPF2_YLL(0x03A7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF2_Y[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**
[31:0]	LPF2_Y[31:0]	Input and output of the register in LPF2 Input: LPF2_Y_{k-1} Output: LPF2_Y_{k}

13.3.60 LPF3_K (0x0398, 0x0399)

LPF3_KH(0x0398)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF3_K[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
LPF3_KL(0x0399)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF3_K[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]	LPF3_K[15:0]	K input of LPF3

13.3.61 LPF3_X (0x039A, 0x039B)

LPF3_XH(0x039A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF3_X[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
LPF3_XL(0x39B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF3_K[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]	LPF3_K[15:0]	X input of LPF3

13.3.62 LPF3_Y (0x039C, 0x039D, 0x039E, 0x039F)

LPF3_YHH(0x039C)								
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Name	LPF3_Y[31:24]							
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
LPF3_YHL(0x039D)								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	LPF3_Y[23:16]							
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
LPF3_YLH(0x039E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LPF3_Y[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
LPF3_YLL(0x039F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF3_Y[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**
[31:0]	LPF3_Y[31:0]	Input and output of the register in LPF3 Input: LPF3_Y_{k-1} Output: LPF3_Y_{k}

13.3.63 PI0_KP (0x03E0, 0x03E1)

PI0_KPH(0x03E0)								
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(0x03E1)								
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

13.3.64 PI0_EK1 (0x03E2, 0x03E3)

PI0_EK1H(0x03E2)								
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(0x03E3)								
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

13.3.65 PI0_EK (0x03E4, 0x03E5)

PI0_EKH(0x03E4)								
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(0x03E5)								
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						

13.3.66 PI0_KI (0x03E6, 0x03E7)

PI0_KIH(0x03E6)								
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(0x03E7)								
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						

13.3.67 PI0_UKH (0x03E8, 0x03E9)

PI0_UKHH(0x03E8)								
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(0x03E9)								
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						

13.3.68 PI0_UKL (0x03EA, 0x03EB)

PI0_UKLH(0x03EA)								
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(0x03EB)								
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

13.3.69 PI0_UKMAX (0x03EC, 0x03ED)

PI0_UKMAXH(0x03EC)								
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(0x03ED)								
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

13.3.70 PI0_UKMIN (0x03EE, 0x03EF)

PI0_UKMINH(0x03EE)								
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(0x03EF)								
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

13.3.71 PI1_KP (0x03D0, 0x03D1)

PI1_KPH(0x03D0)								
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(0x03D1)								
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

13.3.72 PI1_EK1 (0x03D2, 0x03D3)

PI1_EK1H(0x03D2)								
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(0x03D3)								
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						

13.3.73 PI1_EK (0x03D4, 0x03D5)

PI1_EKH(0x03D4)								
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(0x03D5)								
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						

13.3.74 PI1_KI(0x03D6, 0x03D7)

PI1_KIH(0x03D6)								
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(0x03D7)								
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						

13.3.75 PI1_UKH (0x03D8, 0x03D9)

PI1_UKHH(0x03D8)								
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(0x03D9)								
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

13.3.76 PI1_UKL (0x03DA, 0x03DB)

PI1_UKLH(0x03DA)								
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(0x03DB)								
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

13.3.77 PI1_UKMAX (0x03DC, 0x03DD)

PI1_UKMAXH(0x03DC)								
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(0x03DD)								
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

13.3.78 PI1_UKMIN (0x03DE, 0x03DF)

PI1_UKMINH(0x03DE)								
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(0x03DF)								
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

13.3.79 PI2_KP (0x0388, 0x0389)

PI2_KPH(0x0388)								
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(0x0389)								
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

13.3.80 PI2_EK1 (0x038A, 0x038B)

PI2_EK1H(0x038A)								
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(0x038B)								
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

13.3.81 PI2_EK (0x038C, 0x038D)

PI2_EKH(0x038C)								
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(0x038D)								
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

13.3.82 PI2_KI (0x038E, 0x038F)

PI2_KIH(0x038E)								
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(0x038F)								
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

13.3.83 PI2_UKH (0x0390, 0x0391)

PI2_UKHH(0x0390)								
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(0x0391)								
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

13.3.84 PI2_UKL (0x0392, 0x0393)

PI2_UKLH(0x0392)								
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(0x0393)								
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

13.3.85 PI2_MAX (0x0394, 0x0395)

PI2_MAXH(0x0394)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_MAX[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_MAXL(0x0395)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_MAX[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_MAX	Maximum output of PI2

13.3.86 PI2_MIN (0x0396, 0x0397)

PI2_MINH(0x0396)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_MIN[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_MINL(0x0397)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_MIN[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_MIN	Minimum output of PI2

13.3.87 PI3_KP (0x0378, 0x0379)

PI3_KPH(0x0378)								
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 (0x0379)								
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

13.3.88 PI3_EK1 (0x037A, 0x037B)

PI3_EK1H(0x037A)								
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(0x037B)								
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

13.3.89 PI3_EK (0x037C, 0x037D)

PI3_EKH(0x037C)								
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(0x037D)								
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						

13.3.90 PI3_KI (0x037E, 0x037F)

PI3_KIH(0x037E)								
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(0x037F)								
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						

13.3.91 PI3_UKH (0x0380, 0x0381)

PI3_UKHH(0x0380)								
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(0x0381)								
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						

13.3.92 PI3_UKL (0x0382, 0x0383)

PI3_UKLH(0x0382)								
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(0x0383)								
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						

13.3.93 PI3_UKMAX (0x0384, 0x0385)

PI3_UKMAXH(0x0384)								
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(0x0385)								
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

13.3.94 PI3_UKMIN (0x0386, 0x0387)

PI3_UKMINH(0x0386)								
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(0x0387)								
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

14 FOC

14.1 FOC Overview

14.1.1 FOC Introduction

FOC/SVPWM module is used in sensorless and sensored FOC motor drive applications and SVPWM-based motor control applications. Since SVPWM is a subset of FOC module, the following FOC/SVPWM module is referred to as FOC module for short. When DRV_CR[FOC_EN] = 0, FOC module is inactivated, and FOC clock stops. The relevant FOC registers are forced into the reset state and cannot be written.

FOC module consists of angle estimator, PI controller, coordinate transform module and PWM output module. The internal angle estimator can be used to implement sensorless FOC-based control. MCU can also process Hall signals to implement sensored FOC-based control. Moreover, FOC module contains a closed current loop, which outputs six channel PWM signals to drive the motor based on user-defined ID and IQ. Meanwhile, ADC automatically samples current signals to fulfill closed loop current control.

- Sensorless FOC: Angle for coordinate transformation is obtained by angle estimator, the motor speed is estimated for speed closed-loop control and BEMF is sensed for startup detection.
- Sensor-based FOC (Single/Dual/Triple Hall Sensors): FOC module provides the angle input interface. MCU samples Hall signals and calculates electrical angle of the motor. Software sends the result to FOC module for coordinate transformation.

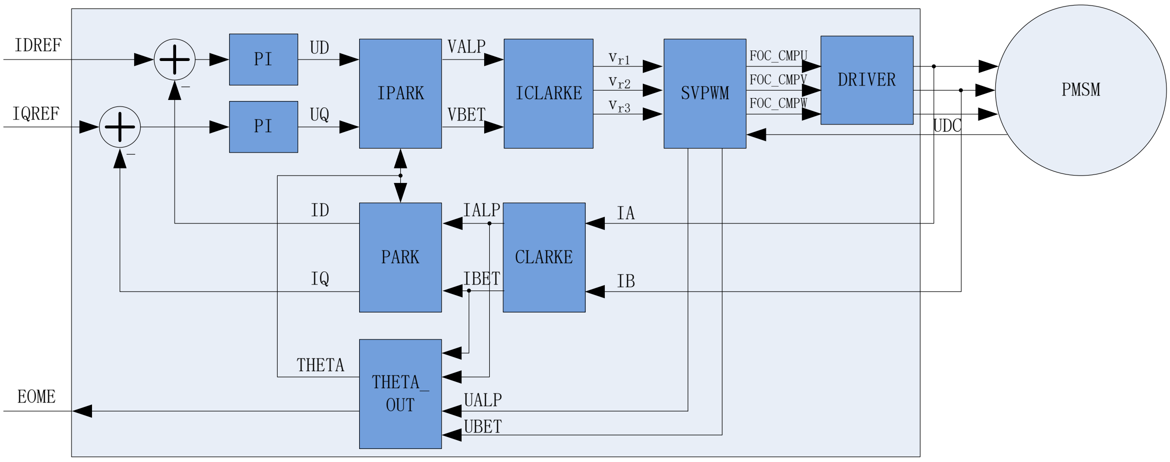


Figure 14-1 FOC Block Diagram

14.1.2 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. FOC module outputs real-time estimated motor speed FOC__EOME. MCU or PI module can use FOC__EOME as the feedback to implement the speed-current dual closed loop control.

14.1.3 PI Controller

FOC module uses 4 PI controllers, which are respectively applied to:

1. Rotor flux control: PI controller of axis D, reference current FOC_IDREF minus feedback current FOC__ID as deviation input, proportional coefficient FOC_DKP and integral coefficient FOC_DKI to adjust the performance of PI controller, FOC_DMAX and FOC_DMIN to limit the output, and finally output D axis voltage 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.
3. Angle estimation: PI controller of the estimator, proportional coefficient FOC_EKP and integral coefficient FOC_EKI adjust the performance of PI controller, and finally output estimated angle FOC__ETHETA.
4. PLL estimation: PI controller of PLL estimator, proportional coefficient FOC_PLLKP and integral coefficient FOC_PLLKI adjust the performance of PI controller, and finally output estimated BEMF FOC__EALP and FOC__EBET.

14.1.4 Coordinate Transformation

14.1.4.1 Inverse Park Transformation

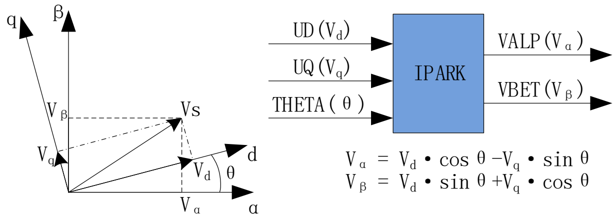


Figure 14-2 Park Transformation

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.

14.1.4.2 Inverse Clarke Transformation

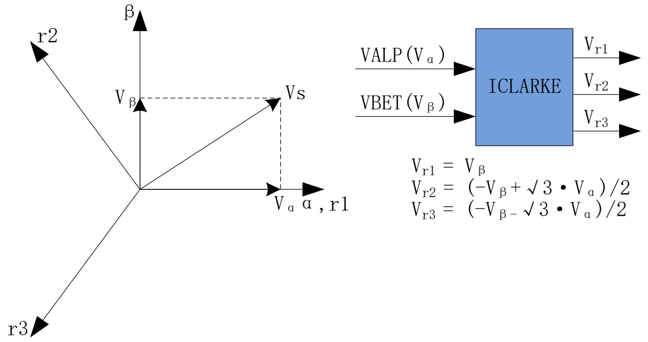


Figure 14-3 Inverse Clarke Transformation

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

14.1.4.3 Clarke Transformation

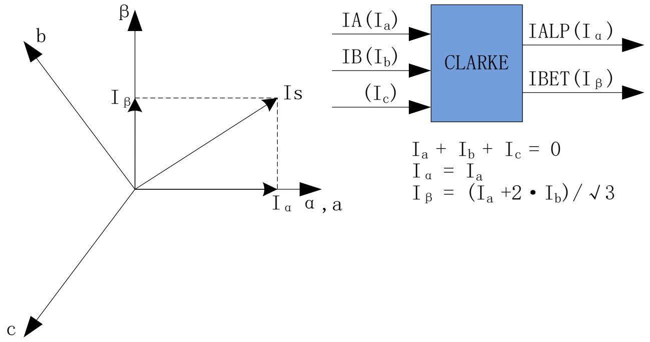


Figure 14-4 Clarke Transform

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

14.1.4.4 Park Transformation

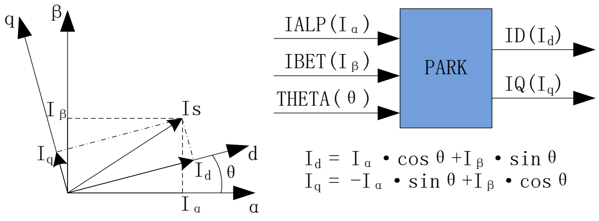

q
β
I_β
Is
I_q
I_d
θ
d
I_α
α
IALP（I_α）
IBET（I_β）
THETA（θ）
PARK
ID（I_d）
IQ（I_q）
$I_d = I_\alpha\cdot\cos\theta + I_\beta\cdot\sin\theta$
$I_q = -I_\alpha\cdot\sin\theta + I_\beta\cdot\cos\theta$

Figure 14-5 Park Transform

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.

14.1.5 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 three-phase motor voltage control, whose process can be reduced to a few simple equations. Since high side and low side of the three-phase inverter cannot be turned on simultaneously, there are two states for a phase, i.e., phase connected to bus voltage (+) or phase connected to ground (-). Therefore, voltage vector output of the inverter has a total of $2^3 = 8$ possible states. When all three phases are connected to bus voltage (+) or ground (-), no voltage drop exists between two phases and the two states are called inactive state or zero voltage vector. The other six states which have voltage output are active voltage vectors with an adjacent state rotation offset of 60 degrees.

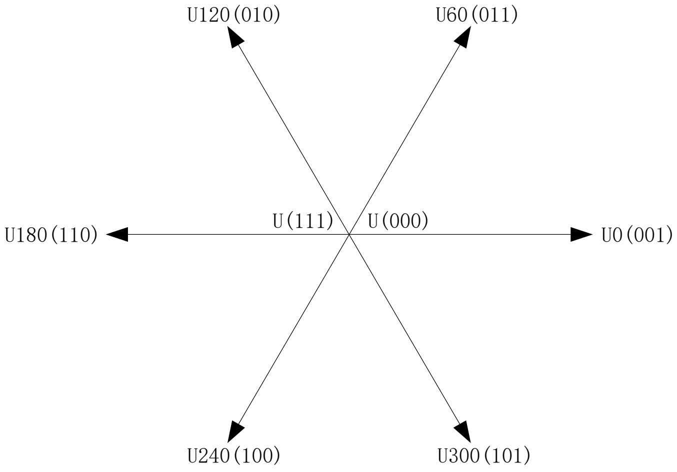


Figure 14-6 SVPWM Voltage Vector

SVPWM uses the sum of two adjacent vectors to generate any voltage vector located in the voltage vector space. As shown in Figure 14-7, U_{OUT} is the desired vector and it is in the sector between U60 and U0. In defined PWM cycle (T), the effect, U0 applied 2T1/T time and U60 applied 2T2/T time, is equivalent to the U_{OUT}. The rest of time (T0) is applied by zero voltage vector.

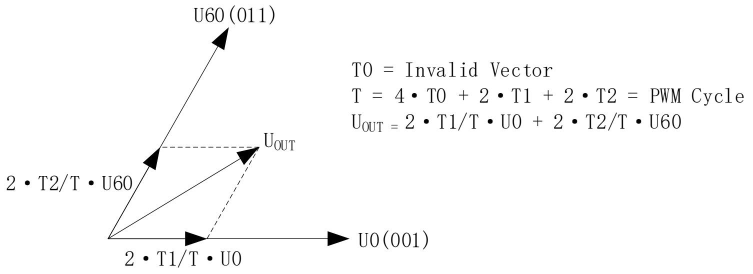

U60(011)
T0 = Invalid Vector
T = 4 • T0 + 2 • T1 + 2 • T2 = PWM Cycle
UOUT = 2 • T1/T • U0 + 2 • T2/T • U60
UOUT
2 • T2/T • U60
U0(001)
2 • T1/T • U0

Figure 14-7 SVPWM Voltage Vector Synthesis

Table 14-1 States of SVPWM Inverter

Phase C	Phase B	Phase A	V_{ab}	V_{bc}	V_{ca}	V_{ds}	V_{qs}	Vector
0	0	0	0	0	0	0	0	U(000)
0	0	1	V_{DC}	0	$-V_{DC}$	$2/3V_{DC}$	0	U0
0	1	1	0	V_{DC}	$-V_{DC}$	$1/3V_{DC}$	$1/3V_{DC}$	U60
0	1	0	$-V_{DC}$	V_{DC}	0	$-1/3V_{DC}$	$1/3V_{DC}$	U120
1	1	0	$-V_{DC}$	0	V_{DC}	$-2/3V_{DC}$	0	U180
1	0	0	0	$-V_{DC}$	V_{DC}	$-1/3V_{DC}$	$-1/3V_{DC}$	U240
1	0	1	V_{DC}	$-V_{DC}$	0	$1/3V_{DC}$	$-1/3V_{DC}$	U300
1	1	1	0	0	0	0	0	U(111)

14.1.5.1 Continuous SVPWM

In single-shunt current sampling mode, continuous SVPWM is always used. In dual-shunt current sampling mode, FOC_CR2[F5SEG] is set to “0” to select continuous SVPWM as the output mode.

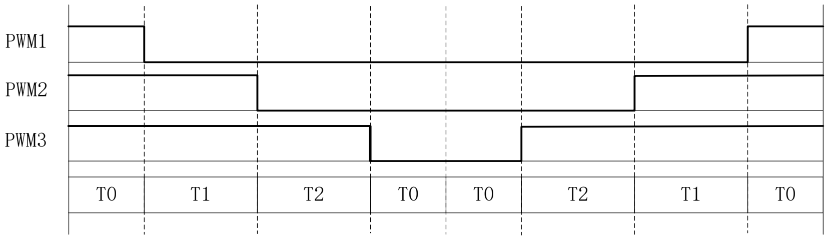


Figure 14-8 Output Level of Continuous SVPWM

14.1.5.2 Discontinuous SVPWM

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

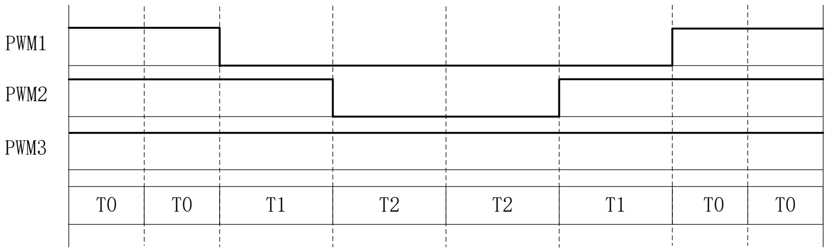


Figure 14-9 Output Level of Discontinuous SVPWM

14.1.6 Overmodulation

Overmodulation is available in single/dual/triple-shunt current sampling mode. FOC_CR1[OVMDL] is set to “1” to enable overmodulation feature. The voltage output, FOC__UD, FOC__UQ and related limit amplitudes (MAX/MIN) are be multiplied by 1.15 in this mode.

14.1.7 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 sinusoidal waveform of the current.

14.1.8 Current and Voltage Sampling

In FOC mode, bus voltage and phase current are sampled by hardware automatically. Before 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 scan mode. Single/dual/triple-shunt current sampling mode is selected by setting FOC_CR1[CSM]. In single-shunt current sampling mode, ADC channel 4 is the default sampling channel of the bus current (itrip). In dual-shunt current sampling mode, ADC channel 0 and channel 1 are the default sampling channels of A-phase current (ia) and B-phase current (ib) respectively. In triple-shunt current sampling mode, ADC channel 0, channel 1 and channel 4 are the default sampling channels of ia, ib and C-phase current (ic) respectively. Channel 2 can be selected for bus voltage sampling.

14.1.8.1 Single-shunt Current Sampling Mode

FOC_CR1[CSM] is set to “0” to select the single-shunt current sampling mode. In this mode, FOC module samples itrip (channel 4) twice during the timer counting-up operation, and samples bus voltage during the timer counting-down operation and 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 advances or delays the current sampling time, and this register shall be configured reasonably to ensure sampling is completed within T1’ and T2’. For example, when MCU clock rate = 24MHz(41.67ns) and

FOC_TRGDLY = 5, the sampling time is delayed by 41.67*5 = 208ns; and when FOC_TRGDLY = 0xFB(–5), the sampling time is advanced by 208ns.

The time of single-shunt current 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 compensation value of deadtime, and FOC module adjusts the PWM waveform automatically.

14.1.8.2 Dual/Triple-shunt Current Sampling Mode

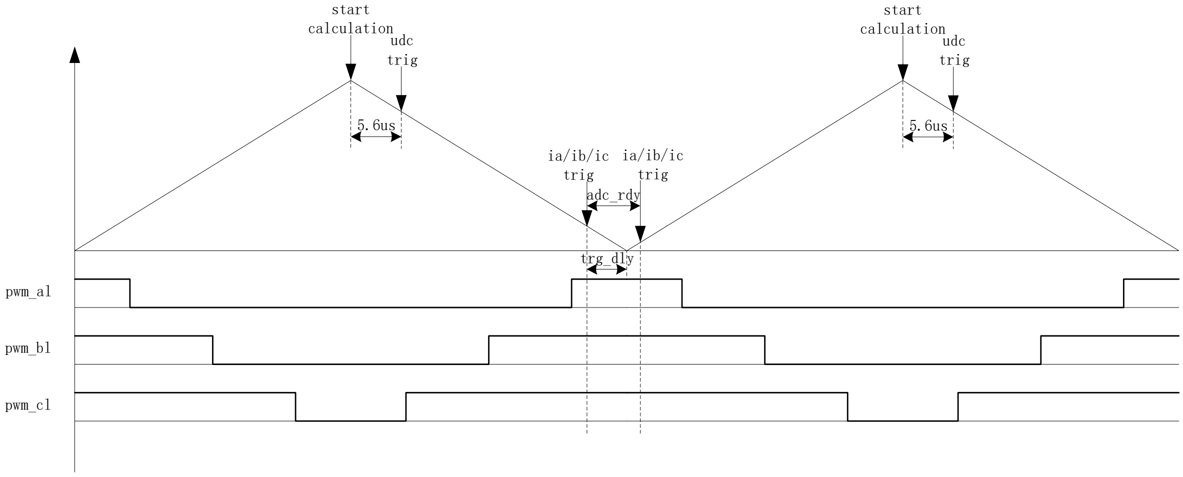


Figure 14-10 Dual/Triple-shunt Sequential Current Sampling Mode

FOC_CR1[CSM] is set to “10/11” and FOC_CR2[DSS] to “0” to select dual/triple-shunt sequential current sampling 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. The bus voltage is sampled during the timer counting-down operation and after FOC module completes the calculation. FOC_TRGDLY shall be configured reasonably to ensure current sampling time is within zero voltage vector (i.e. pwm_al,pwm_bl, pwm_cl=111). For example, when MCU clock rate = 24MHz(41.67ns), FOC_TRGDLY = 0xB2 and FOC timer counts down, ia/ib/ic is sampled at 41.67*50=2.08μs before an underflow event, and then the other phases of ia/ib/ic are sampled.

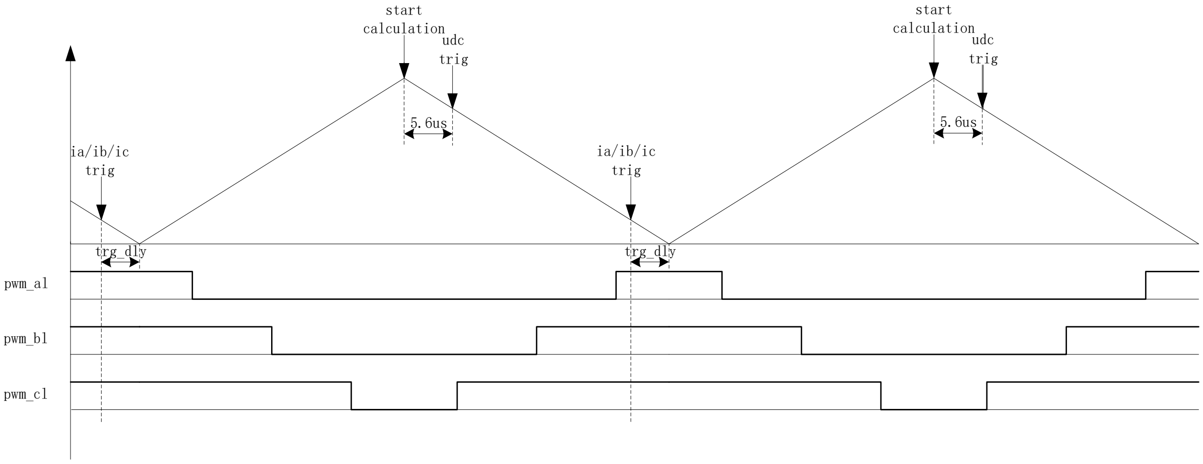


Figure 14-11 Dual/Triple-shunt Alternate Current Sampling Mode

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). FOC_TRGDLY shall be configured reasonably to ensure sampling time for the ia/ib/ic current is within zero voltage vector (i.e. pwm_al,pwm_bl,pwm_cl = 111). For example, when MCU clock rate = 24MHz(41.67ns), FOC_TRGDLY = 0xB2 and FOC timer counts-down, phase current is sampled at 41.67*50 = 2.08μs before an underflow event.

14.1.8.3 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 FOC module does not work and there is no current in three phases, MCU starts to sample the corresponding channel several times, averages all the sampled value, and writes the averaged value 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_CR2[CSOC] = 00/11, FOC_CSO is written to modify the offset of itrip and ic.
- When FOC_CR2[CSOC] = 01, FOC_CSO is written to modify the offset of ia.
- When FOC_CR2[CSOC] = 10, FOC_CSO is written to modify the offset of ib.

14.1.9 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 14-2 Sources of Angle

RFAE	ANGM	EFAE	Source
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

14.1.9.1 Forced Ramping Angle

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

FOC__RTHESTEP (32-bit) = FOC__RTHESTEP (32-bit) + FOC_RTHEACC (32-bit, 16 high-order bits are always 0 and 16 low-order bits are configurable)

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

Forced ramping angle has the highest priority. Configuring FOC_CR1[RFAE] to “1” enables the ramping feature. 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).

14.1.9.2 Forced Pulling Angle

Forced pulling angle is controlled by angle FOC__THETA and speed FOC__RTHESTEP.

The formula is: THETA_OL (16-bit) = THETA_OL (16-bit) + FOC__RTHESTEP (16 high-order bits)

There are two scenarios for forced pulling angle.

- When FOC_CR1[RFAE] is set to “1” and FOC_CR1[ANGM] to “0”, MCU switches to forced pulling angle mode after forced ramping angle mode. The speed is the cumulative result after ramp force angle mode. 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 FOC__RTHESTEP is the initial speed written by software. Configuring FOC__RTHESTEP = 0 enables the pre-position feature. Configuring FOC__RTHESTEP != 0

implements Hall-based FOC (Principle of Hall-based FOC: MCU calculates the angle and speed based on the received Hall signals, and writes them to FOC__THETA and FOC__RTHESTEP for calibration).

14.1.9.3 Estimator Output Angle

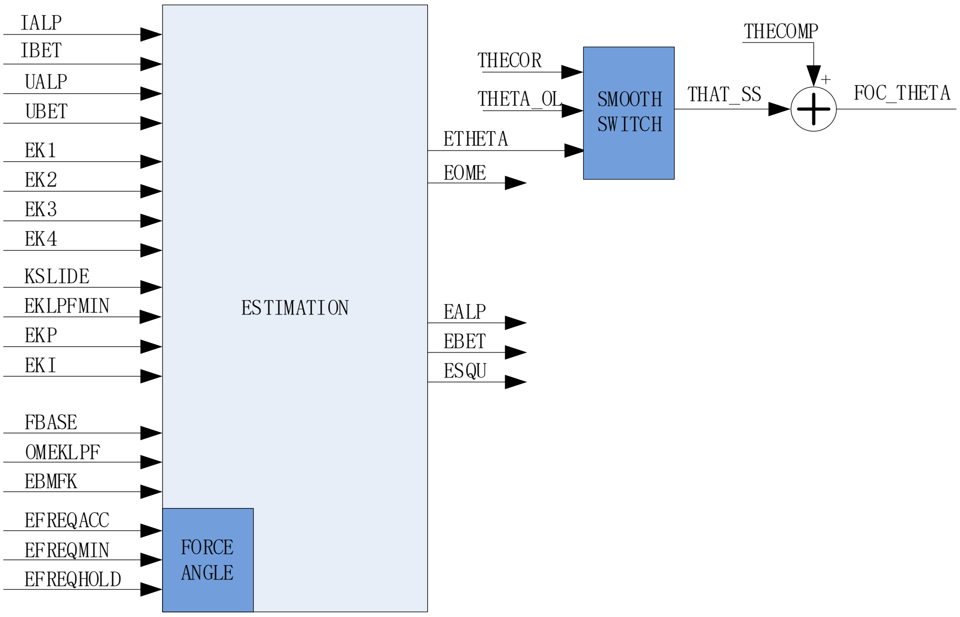


Figure 14-12 Schematic Block Diagram of Estimator

The estimator collects the motor current and voltage and outputs the angle, speed and BEMF based on user-defined motor parameters and other control parameters.

14.1.9.3.1 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 sampled current and voltage. The estimator works in PLL mode or SMO mode by configuring the FOC_CR2[ESEL].

14.1.9.3.2 Forced Angle of Estimator

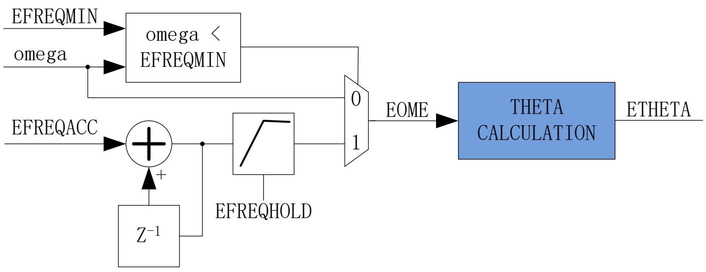


Figure 14-13 Schematic Diagram of Forced Angle of Estimator

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. When FOC_CR1[EFAE] = 1, if 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 EOME.

14.1.9.3.3 Angle Smooth Switching

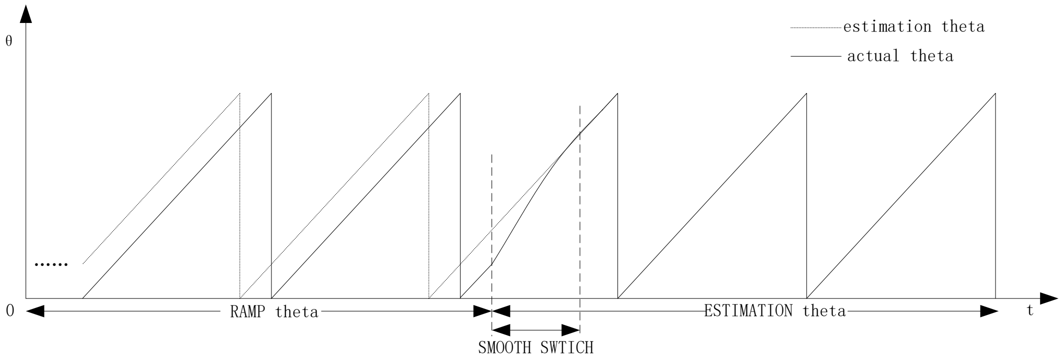


Figure 14-14 Angle Smooth Switching Curve

hen FOC_CR1[RFAE] is set to “1” and FOC_CR1[ANGM] to “1”, the motor starts with ramping feature, and it switches to estimator angle mode after the ramping. 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.

After 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 FOC_THECOR, FOC__ETHETA is selected as the output angle.

14.1.9.3.4 Angle Compensation

Angle compensation value FOC_THECOMP is used to compensate for the estimated angle FOC__ETHETA. If FOC_THECOMP is negative (MSB is 1), the lag angle is compensated; if it is positive (MSB is 0), the lead angle is compensated.

14.1.10 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
- Square of BEMF FOC__ESQU
- Magnitude of BEMF FOC__EMF
- Motor power FOC__POW

14.1.10.1 Tailwind/headwind Detection

FOC module provides tailwind/headwind detection feature. FOC module starts to operate when FOC_CR3[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 in the headwind state 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 in the tailwind state and can be started using estimated angle directly.

14.1.10.2 BEMF Detection

Estimator estimates α-axis BEMF FOC__EALP and β-axis BEMF FOC__EBET with the motor parameters, and calculates FOC__ESQU ($e\alpha^2 + e\beta^2$), which implements protection features, such as motor lock protection, phase loss protection, etc.

14.1.10.3 Motor Power

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

14.2 FOC Registers

14.2.1 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 x0: Single-shunt Current Sampling 01: Dual-shunt Current Sampling 11: Triple-shunt Current Sampling
[1]	RSV	Reserved
[0]	SVPWMEN	SVPWM Module Enable 0: Disable 1: Enable

14.2.2 FOC_CR2 (0x40A1)

Bit	7	6	5	4	3	2	1	0
Name	ESEL	RSV	F5SEG	DSS	CSOC		UQD	UDD
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]	ESEL	Angle Estimator Mode Selection 0: SMO 1: PLL (phase-locked loop). FOC_KSILDE register is FOC_PLLKP of PI controller, and FOC_KLPFMIN register is FOC_PLLKI of PI controller.
[6]	RSV	Reserved

[5]	F5SEG	SVPWM Mode Selection in Dual/Triple-shunt Current Sampling Mode 0: Continuous SVPWM 1: Discontinuous SVPWM (cannot be selected in single-shunt current sampling mode)
[4]	DSS	Dual/Triple-shunt Current Sampling Mode 0: Sequential Sampling Mode, where the current of two phases are sampled in each carrier period. 1: Alternate Sampling Mode, where FOC module completes the calculation in every PWM cycle, and the current of two phases are sampled alternately in two adjacent PWM cycles.
[3:2]	CSOC	Current Sampling Offset Calibration 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 & ic 01: ia 10: ib 11: itrip & 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

14.2.3 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}

Example: Assuming that T_{window} = 1μs, T_{DT} = 1μs, TSMIN = 2μs and carrier period = 62.5μs, then FOC_TSMIN = (1 + 1)/62.5*4096 = 131.

14.2.4 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 designed for 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)

Example: Assuming that it is required to remove narrow pulses with less than 1μs width, deadtime T_{DT} = 1μs, and carrier period = 62.5μs, then FOC_TGLI = (1 + 1)/62.5*4096 =131.

14.2.5 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 Current Sampling Mode If 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)
Example: Assuming that the phase current is not sampled if the low side is turned on for less than 1μs, then FOC_TBLO = 1000ns/41.67ns = 24.		

14.2.6 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	Single-shunt Current Sampling Mode: Trigger Delay for ADC Current Sampling Range (-128,127) Dual/Triple-shunt Current Sampling Mode: Time for Current Sampling TRGDLY[7] decides the sampling occurs in falling or rising interval of the timer: TRGDLY[7] = 0: rising interval; TRGDLY[7] = 1: falling interval Range (0, DRV_ARR[6:0]) Single-shunt Current Sampling Mode: If FOC_TRGDLY = 5, it delays by 5*T = 208ns to sample the current, and if FOC_TRGDLY = 0xFB (complement) or FOC_TRGDLY = -5, it advances by 5*T=208ns. Dual-shunt/Triple-shunt Current Sampling Mode: If FOC_TRGDLY = 0x85 (the highest bit, and the remaining 7 bits are absolute values) and Driver timer counts down, it samples the current at 5*T = 208ns before an overflow event occurs. If OC_TRGDLY = 5 and Driver timer counts up, it samples the current at 5*T = 208ns after an overflow event occurs.

14.2.7 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 current sampling mode, ia, ib in dual-shunt current sampling mode and ia, ib and ic in triple-shunt current

		sampling mode. Range (0,32767), the MSB is always 0
Example: Assuming that the ADC voltage falls within 0V ~ 5V with a reference value of 2.5V, then FOC_CSO = 2.5V/5V*32768 = 16384(0x4000)		

14.2.8 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; The format is the same as that of FOC__THETA Write: Initial speed Read: Current speed Range (-32768,32767) Note: FOC__RTHESTEP is an internal 32-bit variable. MSB is sign bit. High-order 16 bits are written by software.
FOC__RTHESTEP (32 bits) = FOC__RTHESTEP (32 bits) + FOC_RTHEACC (32 bits, 16 high-order bits are always 0 and 16 low-order bits are configurable) THETA (16 bits) = THETA (16 bits) + FOC__RTHESTEP (16 high-order bits)		

14.2.9 FOC_RTHEACC (0x40AA, 0x40AB)

FOC_RTHEACCH(0x40AA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_RTHEACC[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_RTHEACCL(0x40AB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_RTHEACC[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_RTHEACC	Ramping Acceleration; The format is the same as that of FOC__THETA Range (-32768,32767) FOC_RTHEACC is an internal 32-bit variable. MSB is sign bit. Low-order 15 bits are written by software.
FOC__RTHESTEP (32 bits) = FOC__RTHESTEP (32 bits) + FOC_RTHEACC (32 bits, 16 high-order bits are always 0 and 15 low-order bits are configurable) THETA (16 bits) = THETA_OL + FOC__RTHESTEP (16 high-order bits)		

14.2.10 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	Max. ramping counts = FOC_RTHECNT*256 When ramping feature is enabled (FOC_CR1[RFAE] = 1), the ramping angle increases in each PWM cycle. After FOC_RTHECNT*256 times, ramping feature is disabled. Range (0,255)

14.2.11 FOC_THECOR (0x40AD) (Shared with BLDC Motors)

Bit	7	6	5	4	3	2	1	0
Name	FOC_THECOR[7:0]							
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	Angle smooth switching correction: The step value of angle smooth switching after ramping. The format is the same as FOC__THETA. Range (0, 255)

14.2.12 FOC_THECOMP (0x40AE, 0x40AF)

FOC_THECOMPH(0x40AE)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_THECOMP[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_THECOMPL(0x40AF)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_THECOMP[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_THECOMP	Angle Compensation Value The output angle FOC__THETA is derived from the estimator estimated angle + compensation value; the format is same with that of FOC__THETA. Range (-32768,32767)

14.2.13 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	Max. output UD of d-axis PI controller Range (-32768,32767)

14.2.14 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	Min. output UD of d-axis PI controller Range (-32768,32767)

14.2.15 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	Max. output UQ of q-axis PI controller Range (-32768,32767)

14.2.16 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	Min. output UQ of q-axis PI controller; Range (-32768,32767)

14.2.17 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	d-axis voltage calculated by d-axis PI controller Range (-32768,32767)

14.2.18 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	q-axis voltage calculated by q-axis PI controller Range (-32768,32767)

14.2.19 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	ID by Park Transformation Range (-32768,32767)

14.2.20 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	IQ by Park Transformation Range (-32768,32767)

14.2.21 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	Current IBETA Range (-32768,32767)

14.2.22 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	VBETA by Inverse Park Transformation Range (-32768,32767)

14.2.23 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	VALPHA by Inverse Park Transformation Range (-32768,32767)

14.2.24 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 Phase-C Current Range (-32768,32767)

14.2.25 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 Phase-B Current Range (-32768,32767)

14.2.26 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 Phase-A Current; Range (-32768,32767)						

14.2.27 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	Output Angle Write: Forced Pulling Angle Read: Current Angle by FOC Module Range (-32768,32767)						
The bit value -32768 ~ 32767 corresponds to angle range -180°~ 180°. Example: Assuming that FOC__THETA = 8192, the output angle is 8192/32768*180°= 45°.								

14.2.28 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: Output Angle of Estimator (angle before FOC_THECOMP is applied); the format is same as that of FOC__THETA.						

14.2.29 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	Estimated EALPHA by estimator Range (-32768,32767)						

14.2.30 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	Estimated EBETA by estimator Range (-32768,32767)						

14.2.31 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	Output Speed OMEGA of Estimator Range (-32768,32767)						

14.2.32 FOC__ESQU (0x40D6, 0x40D7)

FOC__ESQUH(0x40D6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__ESQU[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__ESQUL(0x40D7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__ESQU[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__ESQU	Square of EBETA + square of EALPHA. The result is obtained from 16 high-order bits, with MSB being always 0. Range (0,32767)

14.2.33 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); An error occurs if this value is negative.

14.2.34 FOC_EKP (0x4074, 0x4075) (Shared with BLDC Motors)

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 of PI controller used by the estimator. Range (0,32767). MSB is always 0. Q12 format.

14.2.35 FOC_EKI (0x4076, 0x4077) (Shared with BLDC Motors)

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 of PI controller used by the estimator Range (0,32767); MSB is always 0; Q15 format.

14.2.36 FOC_EBMFK (0x407C, 0x407D) (Shared with BLDC Motors)

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	Coefficient used to calculate BEMF low-pass filter coefficient EKLPF Range (0,32767); Q15 format.

EKLPF = EBMFK * OMEGA
EBMFK = 2 * pi * fbase*△T		

14.2.37 FOC_KSLIDE (0x4078, 0x4079) (Shared with BLDC Motors)

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: KSLIDE coefficient; Q15 format
		FOC_CR2[ESEL] = 1: KP of PI controller on PLL; Q12 format
		Range (0,32767); MSB is always 0.

14.2.38 FOC_EKLPFMIN (0x407A, 0x407B) (Shared with BLDC Motors)

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_EKLPFMINL(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 factor. EKLPF is forced to be this value when it is lower than this value. Q15 format. FOC_CR2[ESEL] = 1: PI controller KI coefficient on PLL. Q15 format. Range (0,32767); MSB is always 0.

14.2.39 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 factor of estimated speed of the estimator. Range (0,32767); MSB is always 0. Q15 format.

14.2.40 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	Coefficient of angular increment DELTA THETA that is calculated based on OMEGA
FBASE = fbase * △T * 32768 Assuming that fbase = 200Hz and △T = 62.5μs, then FBASE = 409		

14.2.41 FOC_EFREQACC (0x4082, 0x4083) (Shared with BLDC Motors)

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 Range (0,65535) FOC_EFREQACC is an internal 24-bit variable and MSB is sign bit. Low-order 16 bits are written by software.
Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. If speed increment = 3rpm, then FOC_EFREQACC = 3rpm/speed_base*32768*256 = 8388(0x20C4).		

14.2.42 FOC_EFREQMIN (0x4084 0x4085) (Shared with BLDC Motors)

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	Minimum value of Omega: 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) Note: FOC_EFREQMIN is an internal 24–bit variable, and MSB is sign bit. High-order 16 bits are written by software.
Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. Assuming that the minimum Omega = 30rpm, then FOC_EFREQMIN = 30rpm/speed_base*32768 = 327(0x147).		

14.2.43 FOC_EFREQHOLD (0x4086, 0x4087) (Shared with BLDC Motors)

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	Hold value of Omega: When Omega rises to this value, it is kept unchanged. Range (-32768,32768) FOC_EFREQHOLD is an internal 24-bit variable, and MSB is sign bit. High-order 16 bits are written by the software.
Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. If hold value of Omega = 60rpm, then FOC_EFREQHOLD = 60rpm/speed_base*32768 = 655(0x028F).		

14.2.44 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; Range (0, 32767) and MSB is always 0. Q15 format

14.2.45 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.

14.2.46 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; Range (0,32767) and MSB is always 0. Q15 format.

14.2.47 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; Range (0,32767) and MSB is always 0. Q15 format.

14.2.48 FOC_IDREF (0x4090, 0x4091) (Shared with BLDC Motors)

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 Current ID Range (-32768,32767)

14.2.49 FOC_IQREF (0x4092, 0x4093) (Shared with BLDC Motors)

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 Current IQ Range (-32768,32767)

14.2.50 FOC_DQKP (0x4094, 0x4095) (Shared with BLDC Motors)

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 of dq-axis PI Controller Range (0,32767); MSB is always 0. Q12 format.

14.2.51 FOC_DQKI (0x4096, 0x4097) (Shared with BLDC Motors)

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 of dq-axis PI Controller Range (0,32767) ; MSB is always 0. Q15 format.

14.2.52 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 for software use. The default channel is ADC channel 2. Range (0,32767)
Example: The bus voltage is scaled down by 1/6 before feeding into the ADC module, ADC VREF = 5V (namely, the sampling range is [0V ~ 30V]) and FOC__UDCFLT = 19661(0x4CCD), then bus voltage = 19661/32768*5V*6 = 18V.		

14.2.53 FOC_CR3 (0x40ee)

FOC_CR3(0x40ee)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ICLR	RSV	MFP_EN	FOCUSTA	FOCFEN	ESCMS	TSMINH9	TSMINH8
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]	ICLR	Clear FOC__IA/B/CMAX to "0" A write of "1" clears FOC__IA/B/CMAX to "0". The bit is automatically set to "0" after FOC__IA/B/CMAX are cleared to "0". 0: Disable 1: Enable
[6]	RSV	Reserved
[5]	MFP_EN	Adaptive Observer Enable 0: Disable 1: Enable
[4]	FOCUSTA	Single-shunt Off Sampling Enable 0: Disable 1: Enable
[3]	FOCFEN	FOC Force Enable Bit When DRV_CR[MESEL] is set to "1", FOC module performs calculation even if DRV_CR[OCS] = 0. 0: Disable 1: Enable
[2]	ESCMS	ATAN Estimation Enable 0: Disable 1: Enable
[1:0]	TSMINH [9:8]	Scale up by two bits of FOC_TSMIN

14.2.54 FOC_DKP (0x409c, 0x409d)

FOC_DKPH(0x409c)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_DKP[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_DKPL(0x409d)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DKP[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]It	FOC_DKP	KP of d-axis PI Controller Range (0,32767) MSB is always 0, Q12 format It is effective when FOC_DKP! = 0. Otherwise, the original FOC_DQKP/I is applied.

14.2.55 FOC_DKI (0x409e, 0x409f)

FOC_DKIH(0x409e)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_DKI[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_DKIL(0x409f)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DKI[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_DKI	KI of d-axis PI Controller Range (0,32767) MSB is always 0. Q15 format

14.2.56 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	Max. Phase-A Current This value may be unreliable unless the motor rotates in a full electrical period. It is also incorrect if the phase current is collected improperly. This maximum value will not be cleared to “0” automatically unless FOC_CR3[ICLR] is set to “1”. Range (0,32767)

14.2.57 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	Max. Phase-B Current This value may be unreliable unless the motor rotates in a full electrical period. It is also incorrect if the phase current is collected improperly. This maximum value will not be cleared to “0” automatically unless FOC_CR3[ICLR] is set to “1”. Range (0,32767)

14.2.58 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	Max. Phase-C Current This value may be unreliable unless the motor rotates in a full electrical period. It is also incorrect if the phase current is collected improperly. This maximum value will not be cleared to “0” automatically unless FOC_CR3[ICLR] is set to “1”. Range (0,32767)

14.2.59 FOC__EMF (0x40e0, 0x40e1)

FOC__EMFH(0x40E0)								
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(0x40E1)								
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	Estimated BEMF of Estimator This value is the root of sum of square of FOC__EALP and square of FOC__ EBET Range (0,32767)

14.2.60 FOC__UDCPS (0x40e2,0x40e3)

FOC__UDCPSH(0x40e2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__UDCPS[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__UDCPSL(0x40e3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__UDCPS[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__UDCPS	d-axis Voltage Compensation Value The result of d-axis PI controller (FOC__UD) added to FOC_UDCPS is transferred to the next module. Range (-32768,32767)

14.2.61 FOC__UQCPS (0x40e4, 0x40e5)

FOC__UQCPSH(0x40e4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__UQCPS [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__UQCPSL(0x40e5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__UQCPS [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__UQCPS	q-axis Voltage Compensation Value The result of q-axis PI controller (FOC__UD) added to FOC_UQCPS is transferred to the next module. Range (-32768,32767)

14.2.62 FOC__UQEX (0x40e6, 0x40e7)

FOC__UQEXH(0x40e6)								
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(0x40e7)								
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	q-axis Overflow Value for Flux-weakening Control Equation: FOC_UQ - FOC_QMAX FOC__UQEX is positive when FOC_UQ > FOC_QMAX; FOC__UQEX is negative when FOC_UQ < FOC_QMAX; FOC__UQEX can be used to realize flux-weakening control. Range (-32768，32768)

14.2.63 FOC_ID_LPFK (0x40e8)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_ID_LPFK [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**
[7:0]	FOC_ID_LPFK	LPF coefficient of FOC__ID Range 0 ~ 255

14.2.64 FOC_IQ_LPFK (0x40e9)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_IQ_LPFK[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**
[7:0]	FOC_IQ_LPFK	LPF coefficient of FOC__IQ Range 0 ~ 255

14.2.65 FOC_KFG (0x40ea, 0x40eb)

FOC_KFGH(0x40ea)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_KFG [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_KFGL(0x40eb)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_KFG[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_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. Note: When FOC_ KFG = 0, this feature is disabled. Range (0,65535); The clock division factor TIM4_CR0[T4PSC] of Timer4 shall be adjusted if FOC_KFG overflows.

15 Timer1

15.1 Timer1 Operations

Timer1 consists of an internal 16-bit up-counting Base Timer and an internal 16-bit up-counting Reload Timer. Timer1 can be used in the applications of square-wave controlled BLDC motor drive or Hall signals processing.

Timer1 features as follows.

- The 16-bit up-counting Base Timer is used to record the time between two position detected events or writing timings, that is, two phase commutations (60 degree time);
- The 16-bit up-counting Reload Timer is used to control the time between position detection and reload timer overflow, that is, masking time for diode freewheeling as well as the time from ZCP to phase commutation.
- The 3-bit programmable frequency prescaler divides the system clock. The divided clock is used as the clock source of the two timers.
- Input filtering and sampling
- Position detection module generates the position signal required according to the input signal
- The output status register is updated by the write timing module
- 7 groups state register control comparators and outputs
- 6 interrupt sources
  - Basic timer overflow interrupt
  - Reload timer overflow interrupt
  - Writing timing interrupt
  - Position detection interrupt
  - Shield freewheeling end interrupt
  - ADC detection interrupt

The internal structure of Timer1 is shown in Figure 15-1.

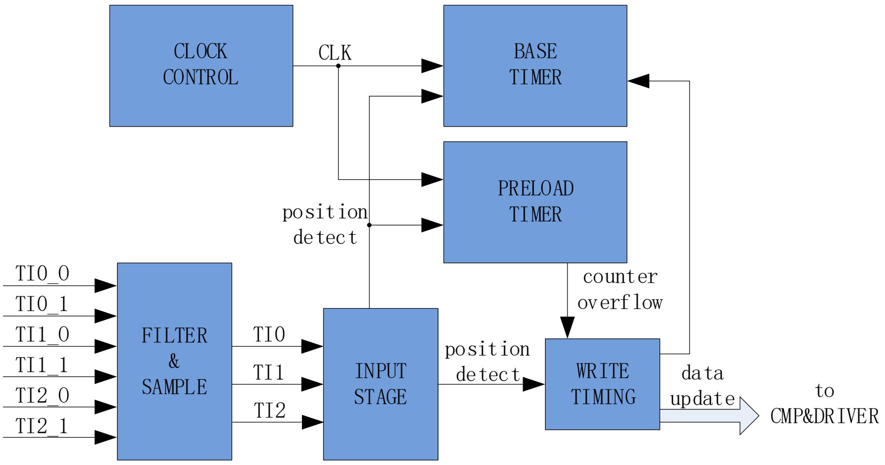


Figure 15-1 Timer1 Internal Structure

15.1.1 Timer1 Counter Module

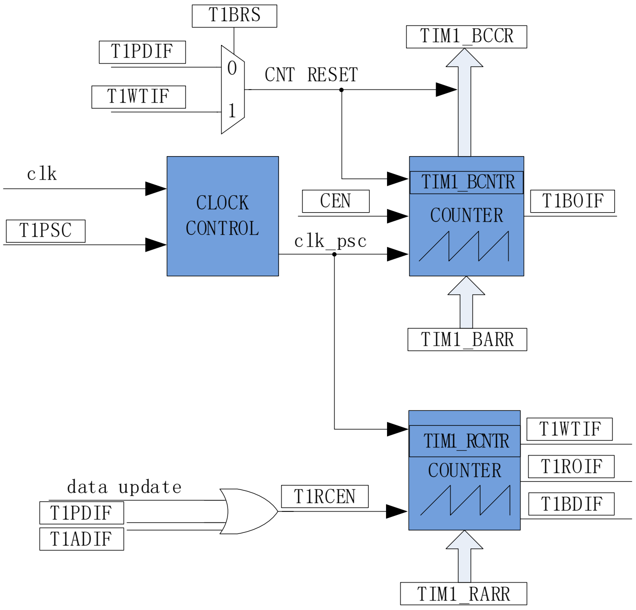


Figure 15-2 Timebase Unit

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

15.1.1.1 Prescaler

Prescaler divides the system clock frequency and generates the counter clock source for Base Timer and Reload Timer. It offers 8 division coefficients and can be selected through 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. Clock rate clk_psc1 = SYSCLK/(2^TIM1_CR3[T1PSC]). The rate corresponding to TIM1_CR3[T1PSC] is shown in Table 15-1.

Table 15-1 Mapping between Clock Rate and TIM1_CR3[T1PSC]

PSC	Coefficient (Hexadecimal)	CLK(Hz)	PSC	Coefficient (Hexadecimal)	CLK(Hz)
000	0x1	24M	100	0x10	1.5M
001	0x2	12M	101	0x20	750K
010	0x4	6M	110	0x40	375K
011	0x8	3M	111	0x80	187.5K

15.1.1.2 Base Timer

The Base Timer is a 16-bit up timer with its count value held in TIM1__BCNTR. When count value of TIM1__BCNTR increases to TIM1__BARR, overflow interrupt flag TIM1_SR[T1BOIF] of the Basic Timer is set to “1” and TIM1__BCNTR continues (instead of being cleared to “0” to restart the counter cycle). TIM1__BCNTR value is loaded into TIM1__BCCR upon a Position Detected Interrupt or a Write Timing Interrupt (selected by TIM1_CR2[T1BRS]). Meanwhile, TIM1__BCNTR is cleared to “0” and restarts the counter cycle.

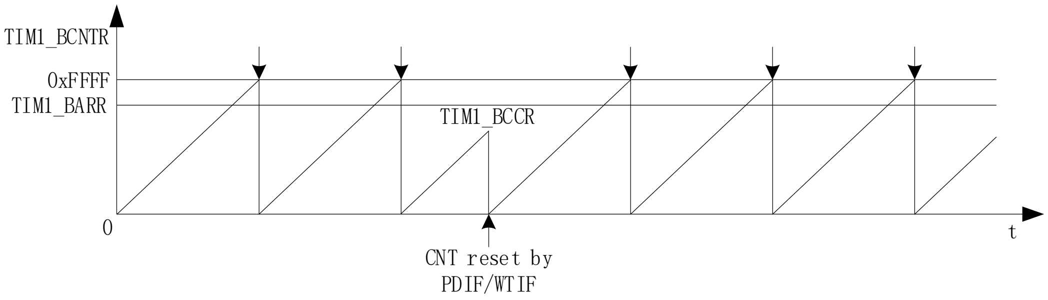


Figure 15-3 Waveform of Base Timer

TIM1_BARR register shall be updated when the basic timer stops working, so that its value is immediately applied to the timer. The overflow event occurs only when TIM1_BCNTR is equal to the value held by TIM1_BARR. When TIM1_BCNTR is larger than the value held by TIM1_BARR, TIM1_BCNTR restarts from 0 after it reaches 0xFFFF. In this case, TIM1_BCNTR cannot be greater than TIM1_BARR after the register is initialized.

15.1.1.3 Reload Timer

The Reload Timer is a 16-bit up timer with its count value held in TIM1__RCNTR. The timer overflows when TIM1__RCNTR increases to TIM1__RARR. In this case, 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”. The timer restarts after TIM1_CR0[T1RCEN] is set to “1”.

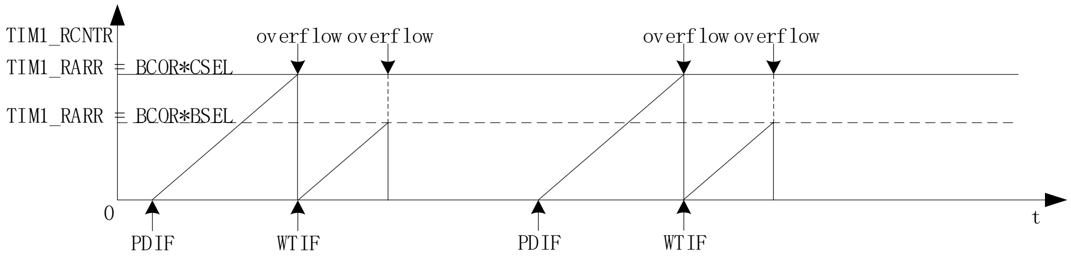


Figure 15-4 Waveform of Reload Timer

TIM1_CR0[T1RCEN] is set to “1” upon a position detection interrupt or write timing interrupt. After the Reload Timer overflows, TIM1_CR0[T1RCEN] is cleared to “0” by hardware and the timer stops counting. The Reload Timer is mainly used to realize diode freewheeling masking and delay commutation after ZCP for square-wave control over BLDC motors. It does not work in other situations.

15.1.2 Position Detection

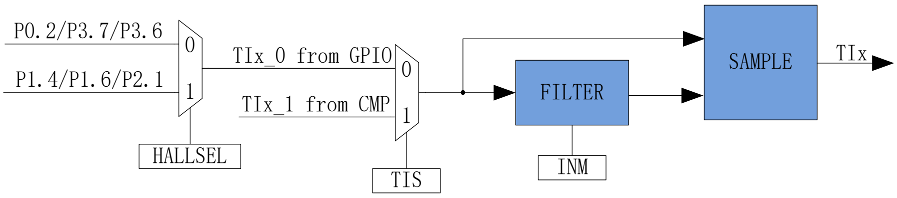


Figure 15-5 Functional Block Diagram of Position Detection

The TIM1_CR3[T1TIS] bit selects the sources of Position Detection signal, including comparator or GPIO. CMP_CR1[HALLSEL] bit is used to configure GPIO sourced by P1.4/P1.6/P2.1. TIM1_CR3[T1INM] bit decides whether CMP/GPIO signal is filtered and CMP_CR3[SAMSEL] bit determines sampling delay feature is enabled or disabled.

15.1.2.1 Position Detection Event

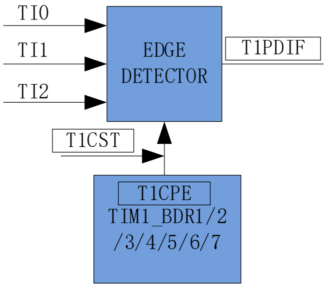


Figure 15-6 Functional Block Diagram of Position Detection

The register bank TIM1_DBR1/2/3/4/5/6/7[T1CPE] is configured to select the active edge of position detection signal. An position detection event is generated after an input active edge (TI2/TI1/TI0) is detected. TIM1_CR4[T1CST] bit selects TIM1_DBR1/2/3/4/5/6/7[T1CPE] timing.

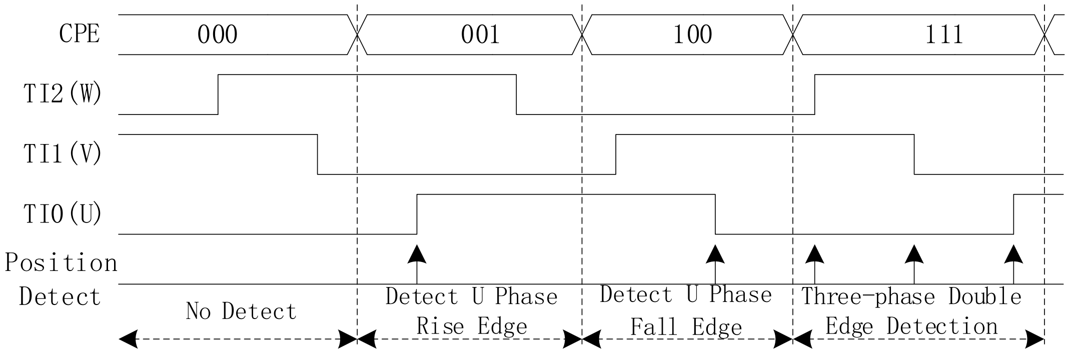


Figure 15-7 Timing Diagram of CMP/GPIO Position Detection

The relation between active edge and TIM1_DBR1/2/3/4/5/6/7[T1CPE] is shown in Table 15-2.

Table 15-2 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.

15.1.2.1.1 Sampling

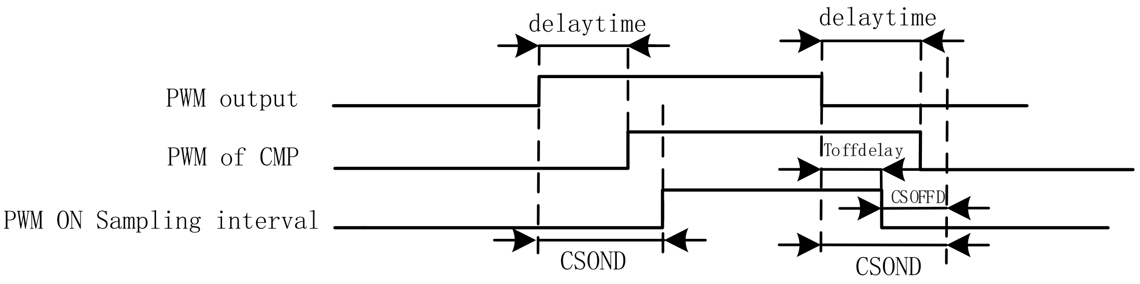


Figure 15-8 PWM ON Sampling Mode

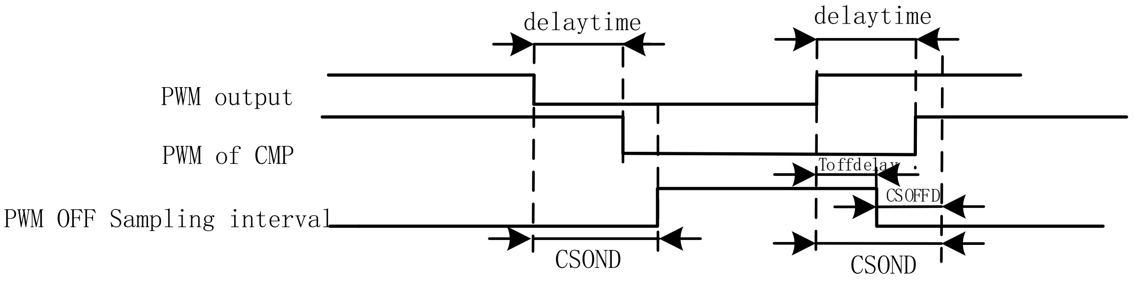


Figure 15-9 PWM OFF Sampling Mode

In the square wave control mode, the input of TI2/TI1/TI0 comes from the comparator. Due to MOS on-off from driving circuit, PWM signals create electromagnetic interference. Therefore, CMP_CR3[SAMSEL], CMP_SAMR[CSOFFD] and CMP_SAMR[CSOND] bits shall be set reasonably to adjust the sampling interval and obtain the valid position detection signal.

There is a delay from PWM output to the output of the comparator, which is mainly affected by the following factors: resistance value of drive resistor, MOS on-off speed, and input delay and hysteresis settings of the comparator. In this case, CMP_SAMR[CSOFFD] is set to delay comparator sampling by offdelay, in order to avoid the interference from comparators, where offdelay = CSOND-CSOFFD and means the sampling delay for CMP0, CMP1 and CMP2.

For example, the delay from PWM output to the output of the comparator is 2μs and the interference width is 1μs, then

CSOFFD >1μs = 1000ns/41.67ns/8 = 3

CSOND > (2 + 1) μs = 3000ns/41.67ns/8 = 9

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. Enable 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.

Method for measuring interference width: The operations are the same as above.

15.1.2.1.2 Filtering

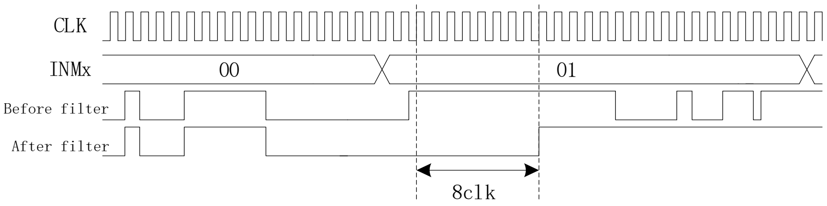


Figure 15-10 Timing Diagram of Filtering Module

According to TIM1_CR3[T1INM], the filtered pulse width of input noise can be selected as 8/32/64 system clock. After this feature is enabled, the signal is lagged behind about 8 ~ 9/32 ~ 33/64 ~ 65 system clock.

15.1.3 Write Timing Interrupt

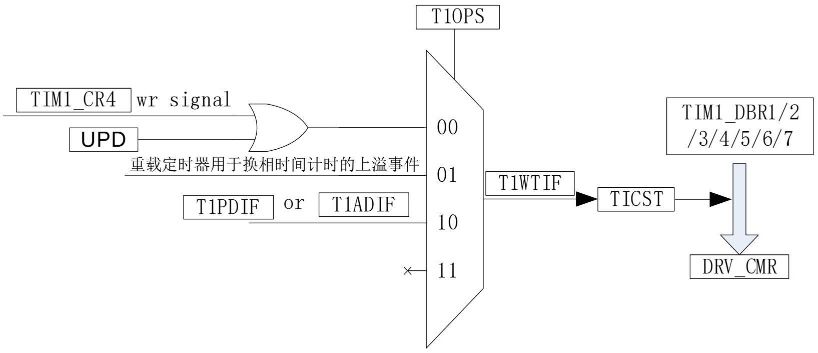


Figure 15-11 Write Timing Block Diagram

The triggered source of write timing interrupt is selected by TIM1_CR0[T1OPS]. After a writing sequence interrupt is generated, writing sequence interrupt flag TIM1_SR[T1WTIF] is set to “1”. If TIM1_CR4[T1CST] ranges in 001 ~ 110, TIM1_CR4[T1CST] adds 1 automatically and the value held by TIM1_DBR1/2/3/4/5/6/7 is updated to DRV_CMR.

15.1.4 Timer1 Interrupt

Timer1 supports 6 interrupt sources:

- Base Timer overflow interrupt
- Reload Timer overflow interrupt
- Writing sequence interrupt
- Position Detected Interrupt
- Diode Freewheeling End Interrupt
- ADC Position Detected Interrupt

Configuring enable bit of TIM1_IER enables or disables the corresponding interrupt request.

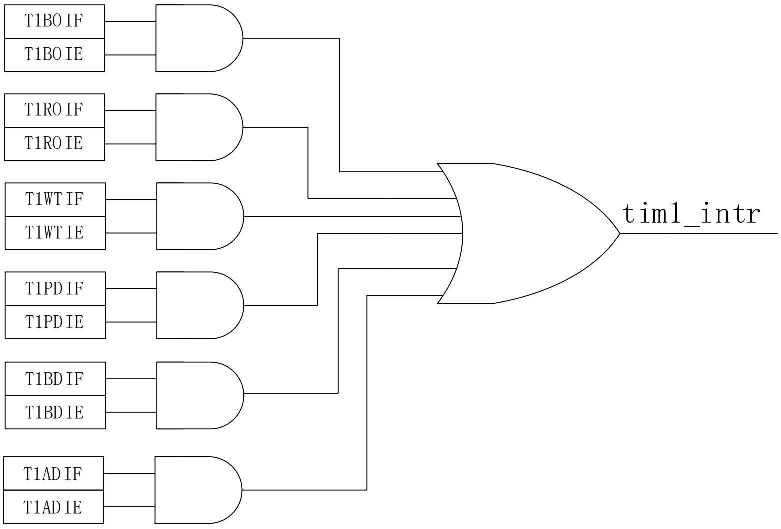


Figure 15-12 Timer1 Interrupt Sources

15.2 Square-wave Control Based BLDC Motor

For BLDC motor square-wave control application, Timer1 works with comparators 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 comparator sampling 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 automatically
- Comparator signals can be sampled during PWM ON/OFF, and the signals can be filtered as well
- Take over DRV_CMR register to control three-phase 6 PWM outputs automatically

Square-wave control is mainly used for sensorless applications of BLDC motors, where 30-degree commutation delay is implemented after BEMF zero-crossing points. The following example is based on this principle (TIM1_CR0[T1OPS] = 01).

15.2.1 Six-Step Phase Commutation of Square Wave Control

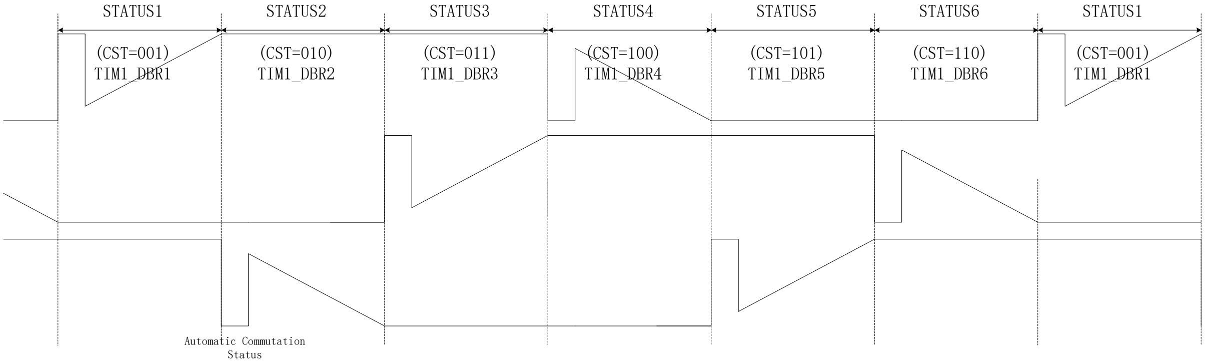


Figure 15-13 Diagram of Six-step Phase Commutation of Square Wave Control

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.

15.2.2 Working Principle of BLDC Motor

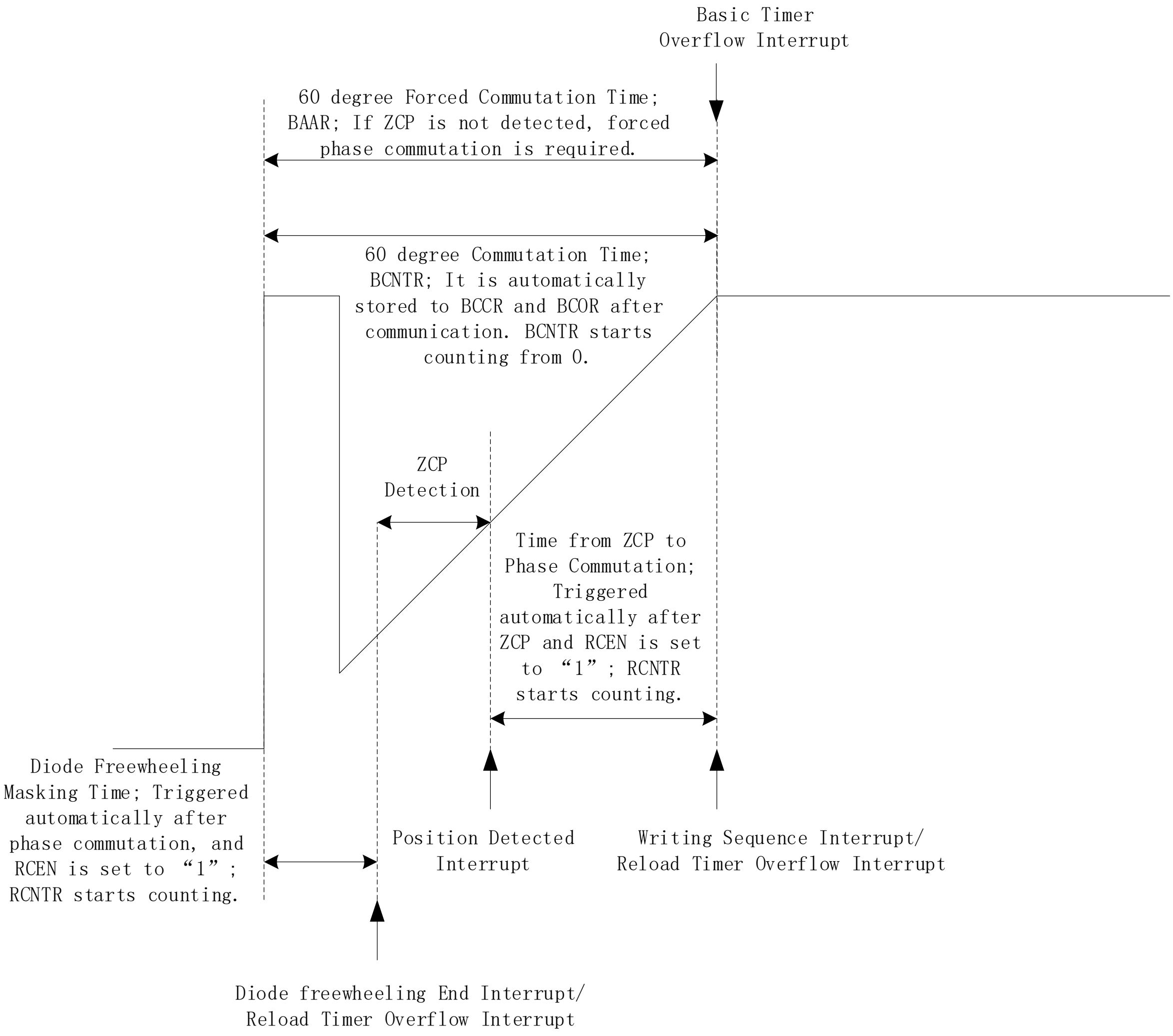

Basic Timer
Overflow Interrupt
60 degree Forced Commutation Time;
BAAR; If ZCP is not detected, forced
phase commutation is required.
60 degree Commutation Time;
BCNTR; It is automatically
stored to BCCR and BCOR after
communication. BCNTR starts
counting from 0.
ZCP
Detection
Time from ZCP to
Phase Commutation;
Triggered
automatically after
ZCP and RCEN is set
to "1"; RCNTR
starts counting.
Diode Freewheeling
Masking Time; Triggered
automatically after
phase commutation, and
RCEN is set to "1";
RCNTR starts counting.
Position Detected
Interrupt
Writing Sequence Interrupt/
Reload Timer Overflow Interrupt
Diode freewheeling End Interrupt/
Reload Timer Overflow Interrupt

Figure 15-14 Working Principle of BLDC Motor

15.2.2.1 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] to "1" 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[T1CFLT] can select the last 1/2/4/8 TIM1__BCCR averaged to obtain TIM1_BCOR.

The diode freewheeling masking time, the time from ZCP to phase commutation, and the time to forced commutation are determined by the 60 degree base time TIM1_BCOR.

15.2.2.2 Phase Commutation

TIM1_CR1[T1OPS] decides phase commutation mode or writing timing event. Configuring TIM1_CR1[T1OPS] = 00 enables sensorless control, TIM1_CR1[T1OPS] = 01 enables sensorless automatic commutation, or TIM1_CR1[T1OPS] = 10 enables sensored commutation

After phase commutation, Timer1 automatically performs the following operations:

1. Store the data held by TIM1_BCNTR to TIM1_BCCR for filtering, and then sends it to TIM1_BCOR as the 60 degree base time;
2. TIM1__BCNTR starts counting from “0”;
3. Start freewheeling mask, write the mask angle into TIM1_RARR and set TIM1_CR0[T1RCEN] to “1”. TIM1_RCNTR begins to count;
4. If TIM1_CR4[TICST] is in the state 1 to 6, it automatically switches to the next state;
5. Write timing interrupt TIM1_SR[T1WTIF] and reload timer overflow interrupt TIM1_SR[T1ROIF] are generated.

15.2.2.3 Forced Commutation at 60°

When the motor rotates smoothly, ZCP is generally detected after 30 degrees of rotation after phase commutation. If ZCP is not detected in 60 degree after the phase commutation, a forced phase commutation is required. In this case, TIM1_CR0[T1FORC] is set to “1” to enable the forced commutation feature. If no ZCP is detected in 60 degree after commutation, TIM1_SR[T1BOIF] (overflow interrupt flag of the Basic Timer) is set to “1” for forced phase commutation (Note: 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[T1FORC] = 0) and TIM1__BCNTR > TIM1__BARR, the interrupt flag TIM1_SR[T1BOIF] is set to “1” and TIM1__BCNTR continues. Phase commutation can be performed manually by Basic Timer overflow interrupt flag TIM1_SR[T1BOIF] and the position detected interrupt flag TIM1_SR[T1PDIF].

15.2.2.4 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. By masking comparator signal during freewheeling time, 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.

Timer1 holds the last latched level value of the comparator during freewheeling masking, and samples the level value after freewheeling masking. If the time for freewheeling masking is shorter than that for freewheeling, a false zero-crossing trigger point is generated. Therefore, it is necessary to adjust freewheeling masking time according to motor characteristics and ensure it is longer than freewheeling time.

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

15.2.2.5 Angle of ZCP to Phase Commutation (Delayed Commutation)

A zero-crossing detection is performed after diode freewheeling masking. If no ZCP is detected, the detection continues until phase commutation. It is triggered by a position detected interrupt, and only the first position detected interrupt is the effective trigger source. If a ZCP is detected, zero-crossing detection is completed and other subsequent trigger sources become invalid. Therefore, parameters related to filtering and sampling must be properly configured to ensure the first trigger source is a correct zero-crossing point.

After commutation, a ZCP is detected (generating a position detected interrupt) and the hardware starts TIM1_RCNTR for 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 writing sequence interrupt flag TIM1_SR[T1WTIF].

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

15.2.2.6 Cycle-by-cycle Current Limiting

See section 29.1.1.2.

15.2.3 BLDC Motor Debugging

The chip supports the following debugging methods.

1. GP07 displays comparator signals in real time
2. GP01 displays the status of Timer1 in real time
3. SPI debugger displays Timer1 related registers on the oscilloscope

15.2.3.1 Comparator Debugging

Configure CMP_CR3[CMPSEL] to output CMP0/1/2_OUT via GP07. CMP0/1/2_OUT is the filtered comparator signal.

Configure CMP_CR3[DBGSEL] = 11 to output comparator signals via GP01. See section 15.1.2.1.1 for waveforms. Select sampling delay by CMP_CR3[SAMSEL], as shown in the below table.

Table 15-3 Relation between CMP_CR3[SAMSEL] and GP01 Output

SAMSEL	Sampling Delay	GP01 Output
00	Sampling at both PWM ON and OFF modes without time delay	Constant high level
01	Sampling at PWM OFF, with time delay according to CMP_SAMR	Sampling at PWM OFF
10	Sampling at PWM ON, with time delay according to CMP_SAMR	Sampling at PWM ON
11	Sampling at both PWM ON and OFF, with time delay according to CMP_SAMR	Sampling at both PWM ON and OFF

Display comparator sampling, one of the signals of CMP0/1/2_OUT and the corresponding phase signal of phase-U/V/W output on the oscilloscope, adjust CMP_SAMR register to fall the comparator sampling waveform within the PWM waveform and then observe whether CMP0/1/2_OUT meets the requirements.

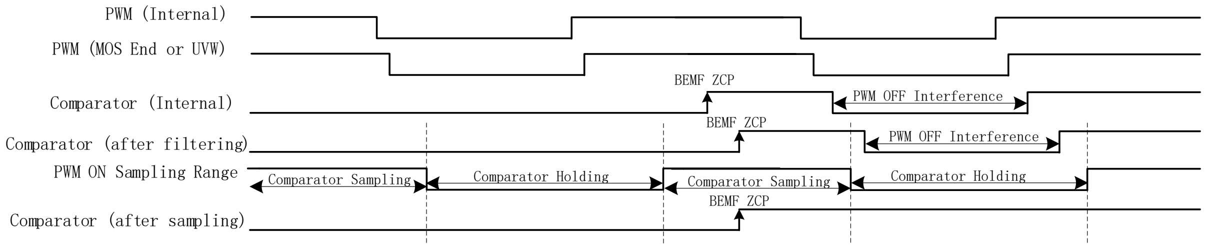


Figure 15-15 Comparator Debugging

15.2.3.2 Debugging on Diode Freewheeling Masking and Phase Commutation

The reload timer RCNTR is used for freewheeling masking and delayed commutation, so SPI debugger can be used to display its waveform on the oscilloscope.

After the real-time status of Timer1 is displayed via GP01, waveforms of the freewheeling masking, ZCP and delayed commutation are obtained through phase-U/V/W commutation points.

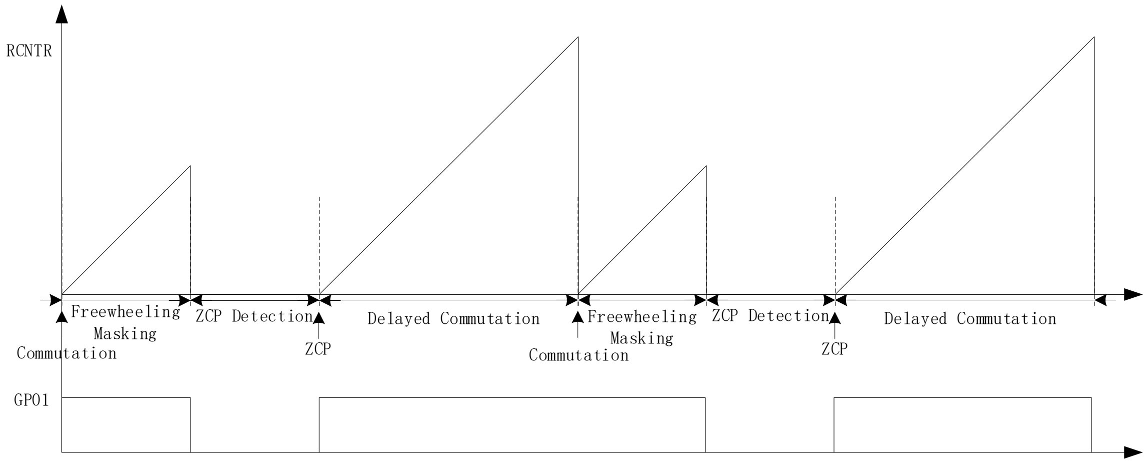


Figure 15-16 RCNTR/GP01 Waveform at 15° Freewheeling Masking and 30° Delayed Commutation

15.3 Phase Commutation Signals by ADC

Formula K*A-B, where K refers to coefficient, A to sampled voltage of active phase and B to sampled voltage of floating phase. When the symbol of the formula changes, phase commutation signal is generated.

Table 15-4 Relation between TIM1_DBR112/3/4/5/6/7[T1CPE] and K/A/B

CPE	Description
000	No sampling
001	A = W, B = U, K = TIM1_KR
010	A = U, B = W, K = TIM1_KF
011	A = U, B = V, K = TIM1_KR
100	A = V, B = U, K = TIM1_KF

CPE	Description
101	A = V, B = W, K = TIM1_KR
110	A = W, B = V, K = TIM1_KF
111	Reserved

Similar to the comparator, ADC can also samples the volage at PWM ON/OFF. To reduce interference, filtering counts for ADC sampled voltage can be set as 2 or 4 by TIM1_CR3[T1AFL].

ADC has basically the same configurations for phase commutation as the comparator. But ADC has more flexible calculation formula, including different coefficients K, symbol flip with different position points, etc. For example, K=TIM1_KR=TIM1_KF=0.5, the symbol flip point is zcp.

15.4 Timer1 Registers

15.4.1 TIM1_CR0 (0x4068)

Bit	7	6	5	4	3	2	1	0
Name	T1RWEN	T1CFLT		T1FORC	T1OPS		T1BCEN	T1RCEN
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]	T1RWEN	When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] must be set to “1” to enable TIM1_CR0[T1RCEN]. This bit can be read to “0” only.
[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 If no ZCP is detected after phase commutation, the hardware implements forced phase commutation. But if a ZCP is detected, forced phase commutation will not be implemented even if TIM1__BCNTR exceeds TIM1__BARR. Note: If TIM1_CR0[T1FORC] = 0, TIM1__BCNTR continues (does not restart from “0”) and forced phase commutation will not be implemented even if TIM1__BCNTR exceeds TIM1__BARR. 0: Disable 1: Enable
[3:2]	T1OPS	Data Transfer Mode Selection The bit selects the trigger signal for TIM1_DBRx to transfer data to DRV_CMR, i.e., write timing even or phase commutation. 00: The transfer is triggered upon a write of “1” to TIM1_IER[T1UPD] in software or on a write to TIM1_CR4[T1CST] (sensorless square-wave control) 01: The transfer is triggered upon an overflow interrupt of 16-bit reload timer during phase commutation (sensorless square-wave control) 10: The transfer is triggered upon a Position Detected Interrupt (sensored square-wave control) or ADC calculation result 11: The transfer is triggered upon an overflow interrupt of 16-bit reload timer during phase commutation or ADC calculation result
[1]	T1BCEN	Base Timer Enable 0: Disable 1: Enable
[0]	T1RCEN	Reload Timer Enable

		When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] must be set to “1” to enable TIM1_CR0[T1RCEN]. TIM1_CR0[T1RCEN] is automatically enabled upon a Position Detected Interrupt and a Write Timing Interrupt. TIM1_CR0[T1RCEN] is cleared to “0” by hardware upon a Reload Timer Overflow Interrupt. 0: Disable 1: Enable

15.4.2 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 Register Auto-load Enable With this bit enabled, TIM1__BCOR is written to TIM1__BARR when Basic Timer is reset due to a Position Detected Interrupt or a Write Timing Interrupt. 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 The bit is used to configure the angle (time) of diode freewheeling masking after phase commutation. Position is not detected during diode freewheeling masking. Equation: Diode freewheeling masking angle = TIM1_CR1[BSEL]/128*60° Note: This bit is invalid in Manual mode.

15.4.3 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 Select This bit is invalid in Manual mode (TIM1_IER[T1MAME] = 1). TIM1__BCNTR can only be cleared by a BCNTR Overflow Interrupt. 0: Write Timing Reset 1: Position Detected Interrupt Reset
[6:0]	CSEL	Phase Commutation Angle Select After a position detected event, phase commutation is implemented after the degree configured by TIM1_CR2[CSEL]. Equation: Commutation angle = TIM1_CR2[CSEL]/128*60° Note: TIM1_CR2[CSEL] shall be set as “1” if the phase commutation angle is required at 0.

15.4.4 TIM1_CR3 (0x406B)

Bit	7	6	5	4	3	2	1	0
Name	T1AFL	T1PSC			T1TIS		T1INM	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bit	Name	Description
[7]	T1AFL	ADC Sampled Voltage Calculation Filtering Counts 0: 4 1: 2
[6:4]	T1PSC	Timer Clock Source Frequency Selection These bits are configured to divide the system clock as the clock source for Base 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 Timer1 filters, samples and generates a position detection on the selected input source. This bit affects CMP0/1/2_OUT and CMP0/1/2_IF results of the comparator module CMP_SR. 00: GPIO as the input (select P1.4/P1.6/P2.1 according to CMP_CR1[7]). The results of CMP_SR is generated through the GPIO. 01: Output signal of CMP0/CMP1/CMP2 as the input. The results of CMP_SR is generated through the comparator. 1x: Reserved
[1:0]	T1INM	Noise Pulse Width Selection for TI0/TI1/TI2 When pulse width of the noise is less than the set value, it is filtered as noise. Assuming that MCU clock runs at 24MHz(41.67ns): 00: Disable 01: 8 system clock cycles, 8 x 41.67ns 10: 32 system clock cycles, 32 x 41.67ns 11: 64 system clock cycles, 64 x 41.67ns

15.4.5 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	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 the TIM1_DBRx[T1CPE]. When TIM1_CR4[T1CST] reads 001~110, Timer1 automatically adds by “1” each cycle upon a Write Timing Interrupt.

Table 15-5 Mapping between TIM1_CR4[T1CST] and TIM1_DBRx

T1CST	TIM1_DBRx	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

15.4.6 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 Write only, and automatically 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. Details: TIM1__BCNTR of the Basic Timer is cleared by a Base Timer 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 controlled 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 controlled 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	Write Timing Interrupt Enable 0: Disable 1: Enable
[1]	T1PDIE	Position Detected Interrupt Enable 0: Disable 1: Enable
[0]	T1BDIE	Diode Freewheeling Masking Interrupt Enable 0: Disable 1: Enable

15.4.7 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/W0	R/W0	R/W0
Reset	-	-	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	T1ADIF	ADC Position Detected Interrupt Flag This bit is set to "1" by hardware when a Position Detected Interrupt is generated. It is cleared to "0" by software. 0: No Interrupt Pending 1: Interrupt Pending
[4]	T1BOIF	Base Timer Overflow Interrupt Flag An overflow event occurs when Basic Timer counts up and TIM1__BCNTR matches with TIM1__BARR. If TIM1_CR0[T1FORC] = 1, TIM1__BCNTR is cleared to "0", otherwise, TIM1__BCNTR continues. This bit is set to "1" by hardware, and cleared

		to “0” by software. Note: If TIM1__BCNTR is to be cleared during an interrupt, UPD or TIM1_CR4 is written when TIM1_CR2[T1BRS] = 0. 0: No Interrupt Pending 1: Interrupt Pending
[3]	T1ROIF	Reload Timer Overflow Interrupt Flag An overflow event occurs and TIM1__RCNTR is cleared to “0” when TIM1__RCNTR matches TIM1__RARR. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[2]	T1WTIF	Writing Sequence Interrupt Flag Writing Sequence Interrupt is generated when TIM1_DBRH/TIM1_DBRL is transferred to TIM1_DRH/TIM1_DRL. This bit is set to “1” by hardware, and cleared to “0” by software. Note: When OPS = 00, a write of “1” to this bit generates a writing sequence interrupt. 0: No Interrupt Pending 1: Interrupt Pending
[1]	T1PDIF	Position Detected Interrupt Flag A position detected interrupt is generated when Position Detection matches TIM1_DBRx[T1CPE]. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[0]	T1BDIF	Diode Freewheeling Masking End Interrupt Flag Diode freewheeling masking starts after phase commutation. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending

15.4.8 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 hods the filtered count value, i.e.60 Degree Base Time.
		Note: TIM1_BCCR and TIM1_BCOR shall be initialized when the 60 Degree Base Time is initialized. For TIM1__BCCR, the 60 Degree Base Time can be directly written. For TIM1_BCOR, please note the followings:
		T1CFLT=00，60 Degree Base Time;
		T1CFLT=01，60 Degree Base Time /2;
		T1CFLT=10，60 Degree Base Time /4;
		T1CFLT=11，60 Degree Base Time /8.

15.4.9 TIM1_DBRx (x=1 ~ 7)(0x4074+2*x, 0x4075+2*x)

TIM1_DBRx(x=1~7) corresponds to the data when CST=1/2/3/4/5/6, respectively. TIM1_DBR1 is taken as an example to introduce TIM1_DBRx registers.

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	TI0/TI1/TI2 Input Edge Polarity and Comparator Enable Select This bit is used to define the edge of Position Detection Input and enable or disable the associated comparators. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt. See section 15.1.2.1 and Table 15-2.
[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 Note: The high-side and low-side outputs of phase-W are complementary and deadtime is automatically added when WLE and WHE are set to “1”.
[4]	T1WLE	Low-side Output Enable of W-phase 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-W are complementary and deadtime is automatically added when WLE and WHE are set to “1”.
[3]	T1VHE	High-side Output Enable of V-phase 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-V are complementary and deadtime is automatically added when VLE and VHE are set to “1”.

[2]	T1VLE	Low-side Output Enable of V-phase 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-V are complementary and deadtime is automatically added when VLE and VHE are set to “1”.
[1]	T1UHE	High-side Output Enable of U-phase 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-U are complementary and deadtime is automatically added when ULE and UHE are set to “1”.
[0]	T1ULE	Low-side Output Enable of U-phase 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-U are complementary and deadtime is automatically added when ULE and UHE are set to “1”.

15.4.10 TIM1__BCNTR (0x4082, 0x4083)

TIM1__BCNTRH(0x4082)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__BCNTRH[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__BCNTRL(0x4083)[7:0]								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__BCNTRL							
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	This bit holds count values of the Base Timer and is used for clocking commutation at 60°. Note: TIM1__BCNTR selects the reset source according to TIM1_CR2[T1BRS], and TIM1__BCNTR does not restart when TIM1__BCNTR overflow interrupt is generated.

15.4.11 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	The bit is configured to capture count values held in Base Timer. When the Base Timer is reset on a Position Detected Interrupt or a Write Timing Interrupt, the count values before the reset are stored into TIM1__BCCR.

15.4.12 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	Auto-Reload Value in Base Timer When the count value of the Base Timer equals to TIM1__BARR value, an overflow interrupt is generated and the counter is cleared to “0”.

15.4.13 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 in Reload Timer When count of the Reload Timer is equal to TIM1__RARR, an overflow interrupt is generated and the value of timer is cleared to “0”.

15.4.14 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	0	0	0	0	0	0	0	0
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	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__RCNTR	Count value of the Reload Timer for diode freewheeling masking and ZCP to phase commutation.

15.4.15 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 The hardware automatically samples the bus current and filters it for software application. The default channel is ADC channel 4. Range (0,32767)

15.4.16 TIM1__UCOP (0x408C, 0x408D)

TIM1__UCOPH(0x408C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__UCOPH[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__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	Active Phase Voltage

15.4.17 TIM1__UFLP (0x408E, 0x408F)

TIM1__UFLPH(0x408E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1__UFLP[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(0x408F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1__UFLP[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	Floating Phase Voltage

15.4.18 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	Voltage Calculation Result

15.4.19 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__UIGNL(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	Ignored Voltage of Active Phase When the voltage of active phase is lower than TIM1__UIGN, the voltage is ignored.

15.4.20 TIM1_KF (0x4094, 0x4095)

TIM1_KFH(0x4094)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1_KF[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_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	Coefficient of the falling edge

15.4.21 TIM1_KR (0x4096, 0x4097)

TIM1_KRH(0x4096)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1_KR[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__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	Coefficient of the rising edge

15.4.22 EXT0 (0x40f0)

Bit	7	6	5	4	3	2	1	0
Name	RSV		FAEN	MOEMD2	T1COM_MD	CMPXO_P11	TIM4_CT	EXT0_P11
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]	FAEN	After this feature is enabled, CMP_SAMR is scaled up by 4 times. 0: Disable 1: Enable
[4]	MOEMD2	10μs of MOE Cycle-by-cycle Current Limiting Enable (MOEMD must be configured as 1X) 0: Disable 1: Enable
[3]	T1COM_MD	Freewheeling Masking by ADC in Square-wave Control Mode 0: Disable. Freewheeling masking is not implemented by ADC. 1: Enable. Freewheeling masking is implemented by ADC (TIM1_CR1[BSEL] ≥ 1).
[2]	CMPXO_P11	Comparator Output Switching 0: P0.7 1: P1.1
[1]	TIM4_CT	DBG Function Switching 0: Reserved 1: P0.6
[0]	EXT0_P11	P11 INT0 Enable 0: Disable 1: Enable

16 Timer2

16.1 Timer2 Instructions

Timer2 has the following five working modes:

- Output mode: PWM generation
- Input capture mode: Detect the duration of high and low level of input PWM
- Input counter mode: Detect input time of the set PWM wave numbers
- QEP & RSD mode: Quadrature Encoder Pulse & Rotating State Detection (tailwind/headwind detection) mode
- Step Mode: Detect rotation direction, position and speed of step motor.

Timer2 features:

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

16.1.1 Prescaler

Prescaler divides the system clock frequency and generates clock source for Base 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.

Clock source frequency formula:

$f_{CK\_CNT}=f_{CK\_PSC}/T2PSC$

when MCU clock rate is at 24MHz(41.67ns).

Table 16-1 Mapping between Clock Rate and TIM2_CR0[T2PSC]

TIM2_CR0[T2PSC]	Coefficient (Hexadecimal)	CLK(Hz)
000	0x01	24M
001	0x02	12M
010	0x04	6M
011	0x08	3M
100	0x10	1.5M
101	0x20	750k
110	0x40	375k
111	0x80	187.5k

16.1.2 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 Base Timer shall be disabled before the write operation. When reading TIM2__CNTR, software reads the high-order bits first, and the hardware synchronously caches the low-order bits. When reading the low-order bits, the software reads the cached data.

16.1.3 Output Mode

When TIM2_CR0[T2OCM] = 01, Timer2 works in output mode.

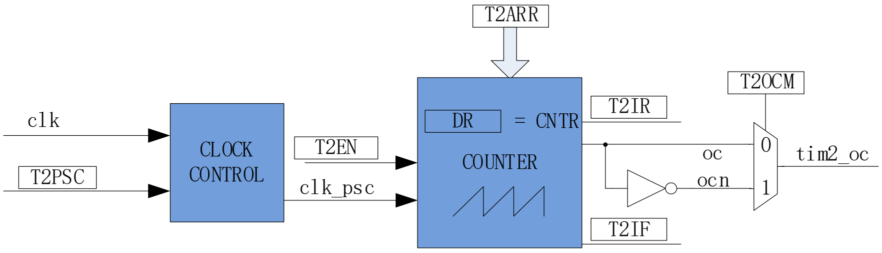


Figure 16-1 Output Mode Block Diagram

The output mode generates output signals according to TIM2_CR0[T2OCM] and the comparison results from the comparator. Meanwhile, corresponding interrupts events are generated.

16.1.3.1 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 Base 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 the preload register is not transferred to the shadow register after the high byte is written or before the low byte is written.

16.1.3.2 High-/Low-level Output Mode

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.

The output signal is always high/low only when TIM2__DR = TIM2__ARR. Configuring TIM2__DR = 0 generates a pulse of one clock cycle.

16.1.3.3 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.

16.1.3.4 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.

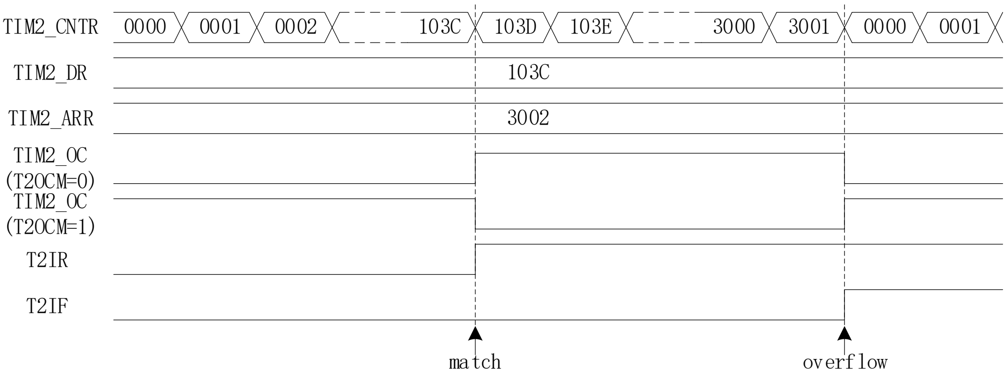


Figure 16-2 Output Mode Waveform

16.1.4 Input Signal Filtering and Edge Detection

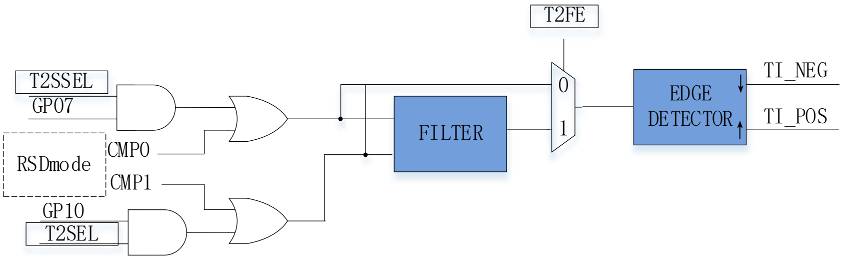


Figure 16-3 Block Diagram of Input Signal Filtering and Edge Detection

The input signal TI of Timer2 comes from GP07 or GP10, set by PH_SEL[T2SEL]. The filter of input signal is optional. The edge detection module detects filtered input signals and records rising edge and falling edge for use by the next module.

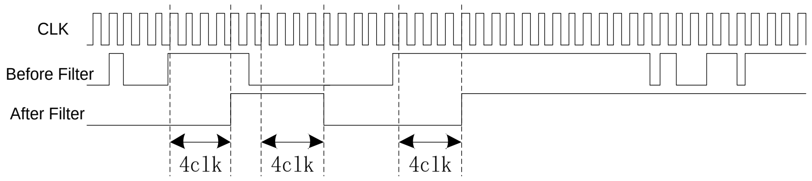


Figure 16-4 Timing Diagram of Filter Module

The filtering circuit removes the input noise with a pulse width of 4 clock cycles. The filtering feature is enabled when TIM2_CR1[T2_FE] is set to “1”. The filtered signal is 4 clock cycles later than the signal before filtering.

16.1.5 Input Capture Mode

When TIM2_CR0[T2MOD] = 00, Timer2 works in input capture mode.

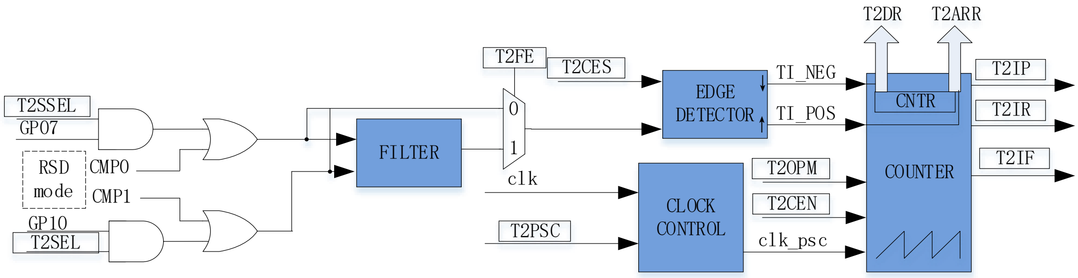


Figure 16-5 Schematic Diagram of Input Capture Mode

The input capture mode detects duty cycle and period of the PWM signal. When TIM2_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 TIM2_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 filter of input signal is optional.

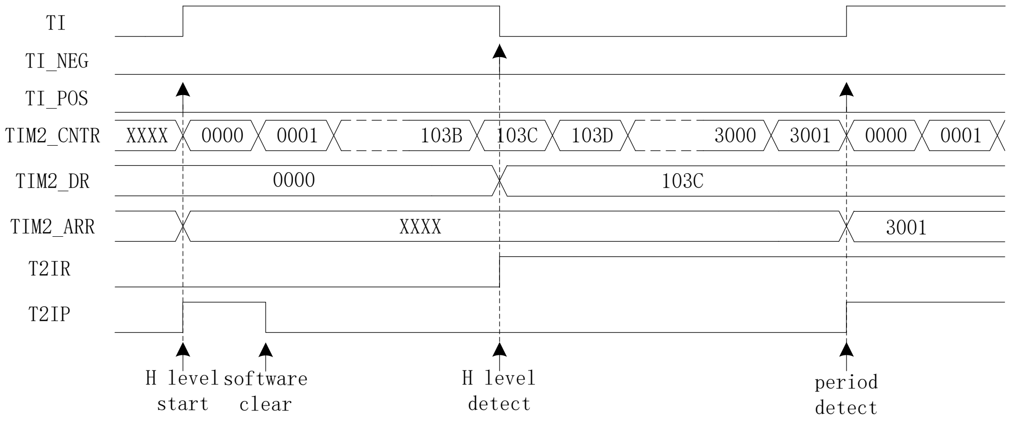


Figure 16-6 Timing Diagram of Input Capture Mode (TIM2_CR0[T2CES] = 0)

For example, when TIM2_CR0[T2CES] = 0, TIM2_CR1[T2CEN] is set to “1” to enable the Base Timer. 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, and 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 TIM2_CR1[T2IF] is set to “1”, and TIM2__CNTR is cleared to “0” and restarts.

16.1.6 Input Counter Mode

When TIM2_CR0[T2MOD] = 10, Timer2 works in input counter mode.

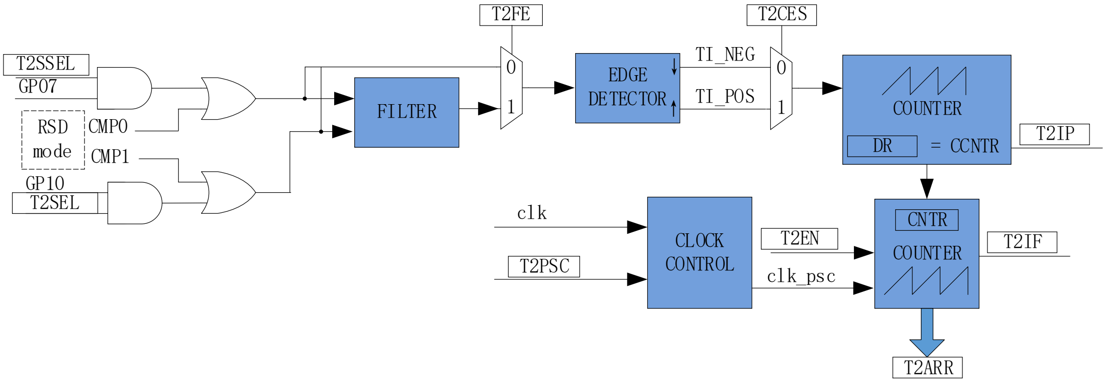


Figure 16-7 Schematic Diagram of Input Counter Mode

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 then sent to the shadow register in case of compare match event (TIM2_CR1[T2IP] = 1), overflow event (TIM2_CR1[T2IF] = 1) or timer 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. In this mode, TIM2__CNTR of the Base Timer is stored in TIM2__ARR. The filter of input signal is optional. When TIM2_CR0[T2CES] is set to “1”, the rising edge of the input PWM signal serves as the active counting edge of the special timer; when TIM2_CR0[T2CES] is set to “0”, the falling edge of the input signal as the active edge.

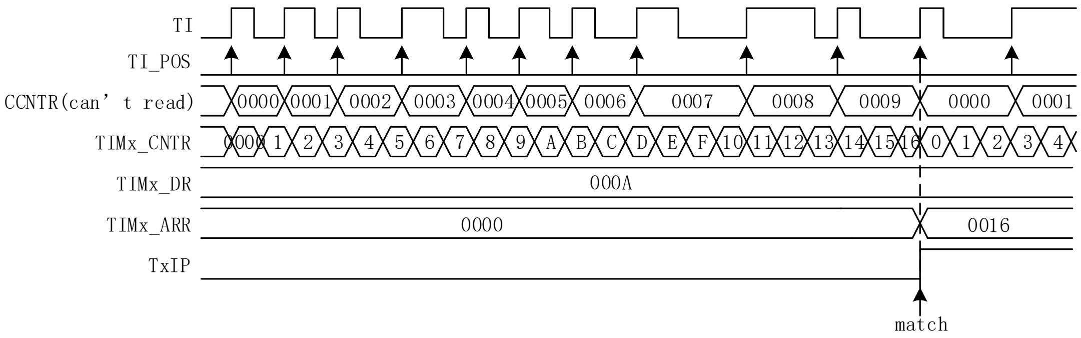


Figure 16-8 Timing Diagram of Input Counter Mode

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.

Whenever active edge of the input signal arrives, one is added to the count value of the special timer CCNTR. When the count value reaches TIM2__DR, TIM2__CNTR is stored in TIM2__ARR. When TIM2_CR1[T2IP] is set to “1”, TIM2__CNTR and CCNTR are cleared to “0” and restart.

When the number of input PWM does not reach 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.

16.1.7 QEP&RSD Mode

When TIM2_CR0[T2MOD] = 11, Timer2 works in QEP & RSD mode.

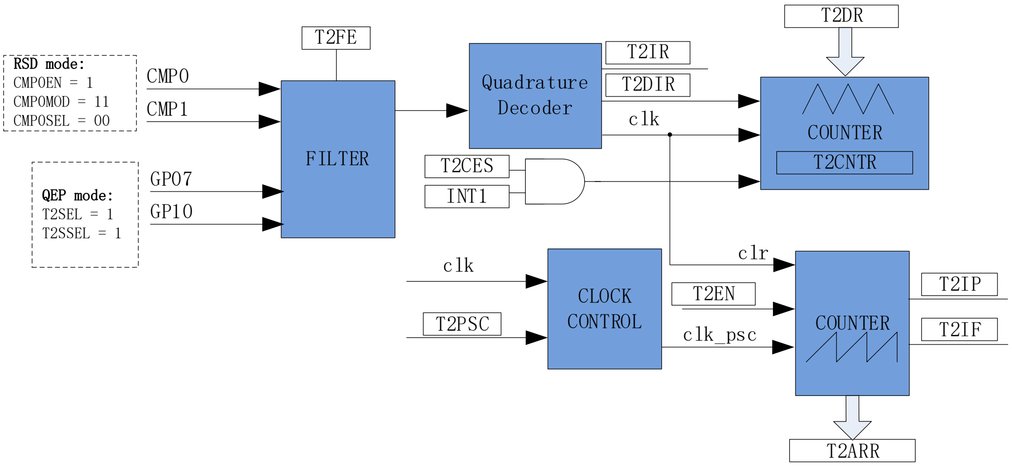


Figure 16-9 Schematic Diagram of QEP&RSD Mode

QEP & RSD mode obtains relative position, direction and speed of the motor by detecting orthogonal signals on two channels. GP07 and GP10 (QEP mode. Two inputs generally differ in phase by 90 degrees) or CMP0 and CMP1 (RSD mode. Two inputs generally differ in phase by 60 degrees) 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]). TIM2_CR1[T2IF] interrupt flag is generated when the direction changes.

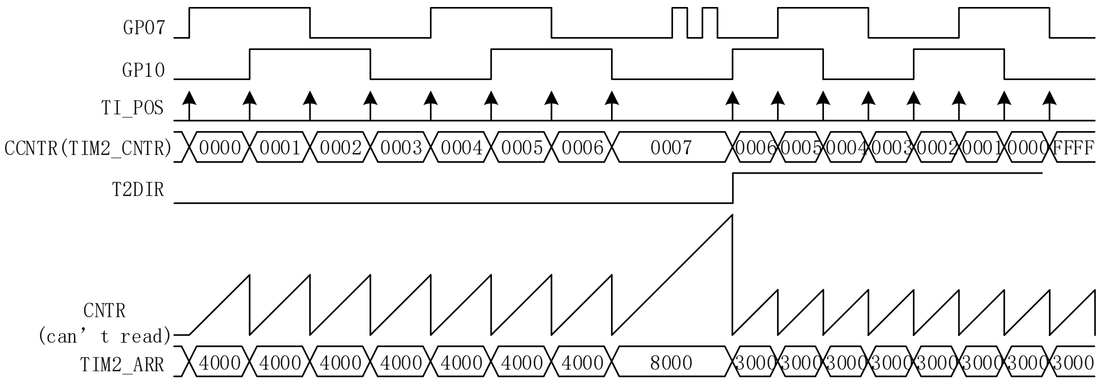


Figure 16-10 Timing Diagram of QEP Mode

(Phase of Orthogonal Input in RSD Mode Different from That in QEP Mode)

The special timer is an up/down timer, and the signal source is the active edge from orthogonal decoding module. If TIM2_CR1[T2DIR] = 0, the direction is positive, and special timer counts upward. When the active edge arrives, the timer increases by one. If TIM2_CR1[T2DIR] = 1, the direction is reverse and special timer counts down. When the active edge arrives, the timer decreases by one. The special timer can be cleared by external interrupt INT1 and the value held by the timer is stored in TIM2__DR, after mechanical zero of the encoder is connected with any port of INT1, INT1 interrupt is enabled and TIM2_CR0[T2CES] = 1. If count value of the special timer 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 timer.

The Base Timer is an up counter, which uses the output of prescaler as the clock source to record the time between two active counting edges. When active counting edge arrives, the value of Basic Timer is stored in TIM2__ARR and then cleared to “0”, and TIM2_CR1[T2IP] interrupt flag bit is set to “1”. When Base Timer counts to 0xFFFF, the count overflows and (TIM2_CR1[T2IF]) interrupt flag is generated.

16.1.7.1 RSD Comparator Sampling

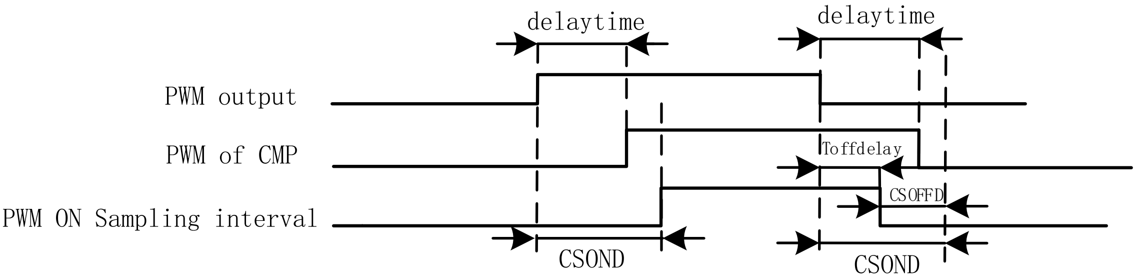


Figure 16-11 PWM ON Sampling Mode

There is a delay from the PWM output to the output of the comparator, which is mainly affected by the following factors: resistance value of drive resistor, MOS on-off speed, and input delay and hysteresis settings of the comparator. As shown above, the delaytime is from the chip output to the comparator output. When high-level sampling is performed, the sampling interval shall be enveloped by actual high-level output of the comparator. First, the sampling ON-delayed time CMP_SAMR[CSOND] is set to overcome output delay and oscillation due to MOS on/off. 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 (PWM of CMP). The sampling OFF-lead time CMP_SAMR[CSOFFD] is set to stop sampling Toffdelay after PWM output falling edge, where Toffdelay = CMP_SAMR[CSOND] – CMP_SAMR[CSOFFD].

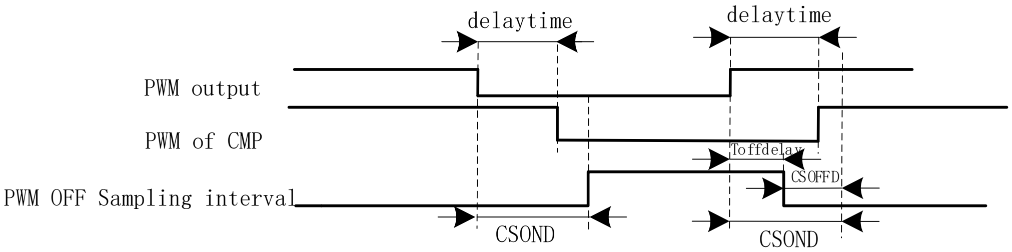


Figure 16-12 PWM OFF Sampling Mode

Similarly, when low-level sampling is performed, sampling ON-delayed time CMP_SAMR[CSOND] and sampling OFF-lead 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. 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.

16.1.8 Step Mode

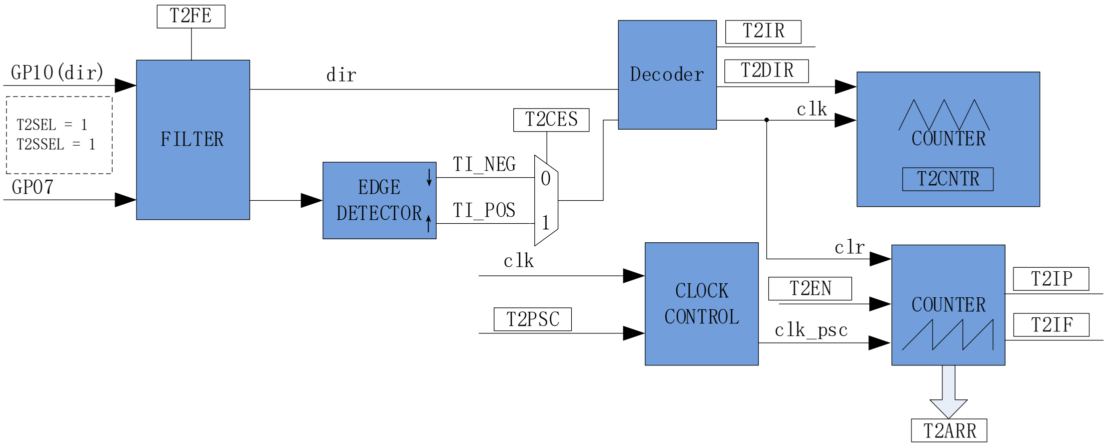


Figure 16-13 Step Mode Schematic Diagram

In step mode, relative position, direction and speed of the step motor are obtained by detecting inputs of the two channel. P1.0 is direction input, and P0.7 is pulse input. Setting TIM2_CR0[T2CES] to select the rising edge or falling edge as the active edge. The input signals are sent to decoding module from the filtering module to obtain the active edge and direction TIM2_CR1[T2DIR]. TIM2_CR1[T2IF] interrupt flag is generated when the direction changes. Note: TIM2_CR1[T2DIR] and TIM2_CR1[T2IR] will not change unless transition occurs at GP10 and active edge is detected at P0.7. To generate an interrupt immediately after P1.0 changes, use INT1.

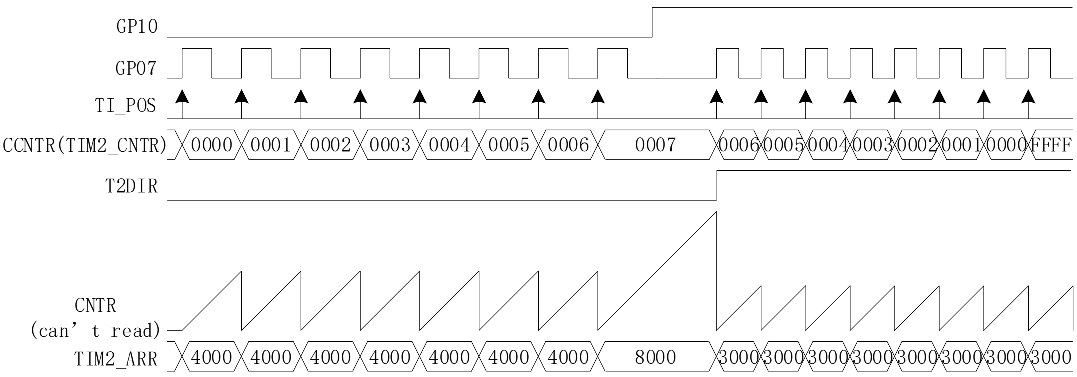


Figure 16-14 Timing Diagram of Step Mode

The special timer is an up/down-counter, and the signal source is active edge of the encoding module. When P1.0 = 0 and TIM2_CR1[T2DIR] = 0, the direction is forward. If active edge of P0.7 arrives, the special timer increases by 1. When P1.0 = 1 and TIM2_CR1[T2DIR] = 1, the direction is reverse. If active edge of P0.7 arrives, the special timer decreases by 1. If count value of the special timer 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 timer.

The Base Timer is an up counter, which uses the output of prescaler as the clock source to record the time between two active counting edges. When active counting edge arrives, the value of Basic Timer is stored in TIM2__ARR and then cleared to “0”, and TIM2_CR1[T2IP] interrupt flag bit is set to “1”. When Base Timer counts to 0xFFFF, the count overflows and (TIM2_CR1[T2IF]) interrupt flag is generated.

16.2 Timer2 Registers

16.2.1 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 Prescaler Selection It is configured to divide the system clock frequency and generate the clock source for Base Timer. When MCU clock runs at 24MHz(41.67ns), the prescaled clock rates are configured as follows: 000:0x1 (24MHz) 001:0x2 (12MHz) 010:0x4 (6MHz) 011:0x8 (3MHz) 100:0x10 (1.5MHz) 101:0x20 (750kHz) 110:0x40 (375kHz) 111:0x80 (187.5kHz)
[4]	T2OCM	Output Mode: Output Mode Selection 0: “0” is output when TIM2__CNTR≤TIM2__DR; “1” is output when TIM2__CNTR ＞ TIM2__DR. 1: “1” is output when TIM2__CNTR≤TIM2__DR; “0” is output when TIM2__CNTR ＞ TIM2__DR. Input Counter Mode: No effect Input Capture Mode: No effect QEP&RSD Mode and Step Mode Selection 0: QEP&RSD Mode 1: Step Mode
[3]	T2IRE	Output Mode: Match Interrupt Enable Input Capture Mode: Pulse Width Detection Interrupt Enable Input Counter Mode: No effect QEP&RSD Mode and Step Mode: Direction Change Interrupt Enable 0: Disable 1: Enable
[2]	T2CES	Input Capture Mode: Counting 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). Input Counter Mode and Step Mode: Active Edge Selection

		0: Falling Edge Count 1: Raising Edge Count QEP&RSD Mode: Pulse Counter Cleared Enable upon Z Signal Interrupt INT1 0: Disable 1: Enable
[1:0]	T2MOD	Mode Selection 00: Input Capture Mode 01: Output Mode 10: Input Counter Mode 11: QEP&RSD Mode or Step Mode

16.2.2 TIM2_CR1 (0xA9)

Bit	7	6	5	4	3	2	1	0
Name	T2IR	T2IP	T2IF	T2IPE	T2IFE	T2FE	T2DIR	T2EN
Type	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	T2IR	Output Mode: Match Interrupt Flag The bit is set to “1” by the hardware when TIM2__CNTR matches TIM2__DR. It is cleared to “0” by software. Input Capture Mode: Pulse Width Detection Interrupt Flag This bit is set to “1” when an input pulse width is detected (Setting TIM2_CR0[T2CES] to select the rising edge or falling edge as the active edge). It is cleared to “0” by software. Input Counter Mode: No effect QEP&RSD Mode and Step Mode: Direction Change Flag Bit 0: No Interrupt Pending 1: Interrupt Pending
[6]	T2IP	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag This bit is set to “1” when an input PWM cycle is detected (Setting TIM2_CR0[T2CES] to select the rising edge or falling edge as the active edge). It is cleared to “0” by software. Input Counter Mode: PWM Input Counter Match Interrupt Flag This bit is set to “1” by hardware when the number of input PWM reaches TIM2__DR. It is cleared to “0” by software. QEP&RSD Mode and Step Mode: Active Edge Detection Interrupt Flag This bit is set to “1” by hardware when the input edge is detected as an active edge. It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[5]	T2IF	Output Mode: Timer Overflow Interrupt Flag. This bit is set to “1” and TIM2__CNTR is cleared to “0” when TIM2__CNTR matches TIM2__ARR. It is cleared to “0” by software. Input Capture Mode: Timer Overflow Interrupt Flag Setting TIM2_CR0[T2CES] to select the rising edge or falling edge as the active edge. This bit is set to “1” and TIM2__CNTR is cleared to “0” when the Timer has not detected an input PWM cycle but the timer TIM2__CNTR reaches 0xFFFF (overflow occurs). It is cleared to “0” by software. Input Counter Mode: Base Timer Overflow Interrupt Flag This bit is set to “1” and TIM2__CNTR is cleared to “0” when the input PWM cycle has not reached the preset TIM2__DR value but Base Timer TIM2__CNTR value reaches 0xFFFF (overflow occurs). It is cleared to “0” by software. QEP&RSD Mode and Step Mode: Base Timer Overflow Interrupt Flag This bit is set to “1” and Basic Timer is cleared to “0” when Basic Timer reaches to 0xFFFF (overflow occurs). It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending

[4]	T2IPE	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Enable Input Counter Mode: PWM Input Counter Match Interrupt Enable QEP&RSD Mode and Step Mode: Active Edge Detection Interrupt Enable 0: Disable 1: Enable
[3]	T2IFE	Output Mode: Timer Overflow Interrupt Enable Input Capture Mode: Timer Overflow Interrupt Enable Input Counter Mode: Base Timer Overflow Interrupt Enable QEP&RSD Mode and Step Mode: Base Timer Overflow Interrupt Enable 0: Disable 1: Enable
[2]	T2FE	Input Signal Filter Selection When input signals are filtered out as noise if the pulse width is less than 4 clock cycle. Assuming that MCU clock runs at 24MHz (41.67ns) , then the pulse width for filtering is 166.67ns. 0: Disable 1: Enable
[1]	T2DIR	QEP&RSD Mode and Steo Mode: Indicator of Motor Rotation Direction 0: Forward 1: Backward
[0]	T2EN	Timer Enable 0: Disable 1: Enable

16.2.3 PI_LPF_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2SS	RSV				PIRANGE	PISTA	LPFSTA
Type	R/W	-	-	-	-	R/W	R/W	R/W
Reset	0	-	-	-	-	0	0	0

Bit	Name	Description
[7]	T2SS	Input Mode Selection of TIM2 Stepper Motor 0: P1.0 is direction input and P0.7 is pulse input 1: P1.0 is reverse pulse input and P0.7 is the forward pulse input
[6:3]	RSV	Reserved
[2:0]		See section 11.3.1.

16.2.4 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 values held in the Base Timer QEP&RSD Mode/Step Mode: Count values held in the special timer When TIM2__CNTR is 0xFFFF, this bit is automatically cleared to “0”.

16.2.5 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 value (written by software) Input Capture Mode: Count value of the detected input pulse width based on TIM2_CR0[T2CES] (written by hardware) Input Counter Mode: PWM cycles to be counted (written by software) QEP&RSD: Value of the special timer when TIM2_CR0[T2CES] = 1 and INT1 (zero point) is detected. Step Mode: No effect

16.2.6 STIM2__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: Overload value (written by software) Input Capture Mode: Count value of a PWM cycle (written by hardware) Input Counter Mode: Count value held in Base Timer when the input PWM count matches (written by hardware) QEP&RSD Mode and Step Mode: Count value held in Base Timer when the input signal is detected as an active edge (written by hardware)

17 Timer3/Timer4 (Timer4 supports timing only and does not support input and output)

17.1 Timer3 Instructions

Timer3 support output and input modes:

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

Timer3 includes:

- 3-bit programmable prescaler divides system clock as the clock source for Base Timer
- 16-bit up-counting Base Timer; The output of the prescaler serves as the counting clock source
- Input filtering module
- Edge detection Module
- Output module generates PWM signal and outputs single compare results
- Interrupt event

17.2 Timer4 Instructions

Timer4 includes:

- 3-bit programmable prescaler divides system clock as the clock source for Base Timer
- Interrupt event

17.2.1 Prescaler

Prescaler divides the system clock frequency and generates counter clock source for Base 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:

$f_{CK\_CNT}=f_{CK\_PSC}/TxPSC$

when MCU clock runs at 24MHz(41.67ns).

Table 17-1 Mapping between Clock Rate and TIMx_CR0[TxPSC]

TIMx_CR0[TxPSC]	Coefficient (Hexadecimal)	CLK(Hz)
000	0x01	24M
001	0x02	12M
010	0x04	6M
011	0x08	3M
100	0x10	1.5M
101	0x20	750K
110	0x40	375K
111	0x80	187.5K

Note: In Input Capture Mode of Timer3, the clock rate is 48MHz when TIM234_CTRL[0] is set to “1”.

17.2.2 Reading, Writing and Counting of TIMx__CNTR

TIMx__CNTR starts when TIMx_CR1[TxEN] = 1. The write operation to TIMx__CNTR directly changes the value of the register, so it is required to disable the timer before the write operation. When reading TIMx__CNTR, the software reads high-order bits first and then low-order bits, and the hardware caches the low-order bits simultaneously. When reading the low-order bits, the software reads the cached data.

17.2.3 Output Mode

When TIMx_CR0[TxMOD] = 1, Timer3 works in output mode.

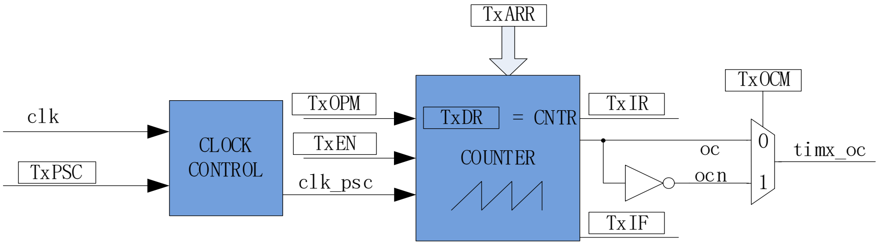


Figure 17-1 Output Mode Block Diagram

The output mode generate output signals according to TIMx_CR0[TxOCM] and the comparison results from the comparator. Meanwhile, corresponding interrupts is generated.

17.2.3.1 High-/Low-level Output Mode

When TIMx_CR0[TxOCM] = 0 and TIMx__DR = TIMx__ARR, the output signals are always low. When TIMx_CR0[TxOCM] = 1 and TIMx__DR = TIMx__ARR, the output signals are always high.

The output signal is always high/low only when TIMx__DR = TIMx__ARR. Configuring TIMx__DR = 0 generates a pulse of one clock cycle.

17.2.3.2 PWM Generation

In PWM generation mode, TIMx__ARR determines PWM cycle, and TIMx__DR determines the duty cycle, and duty cycle = TIMx__DR/TIMx__ARR*100%. If TIMx_CR0[TxOCM] = 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[TxOCM] = 1, the high level is output when TIMx__CNTR ≤ TIMx__DR, and low level is output when TIMx__CNTR ＞ TIMx__DR.

17.2.3.3 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.

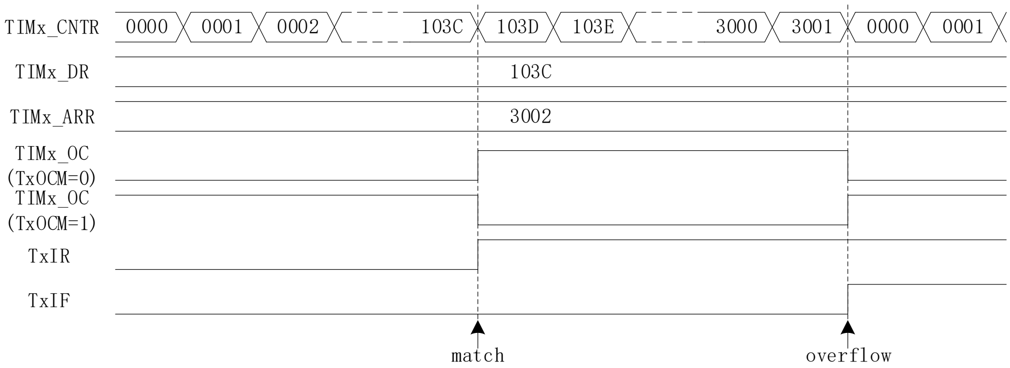


Figure 17-2 Output Waveform of Output Mode

17.2.4 Input Signal Filtering and Edge Detection

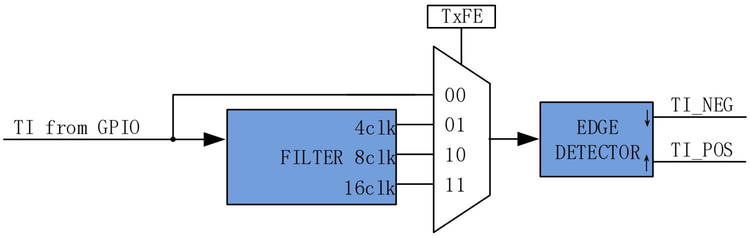


Figure 17-3 Block Diagram of Input Signal Filtering and Edge Detection

The input signal TI of Timer3 comes from P1.1/P0.1. The filter of input signal is optional. The edge detection module detects filtered input signals and records rising edge and falling edge for use by the next module.

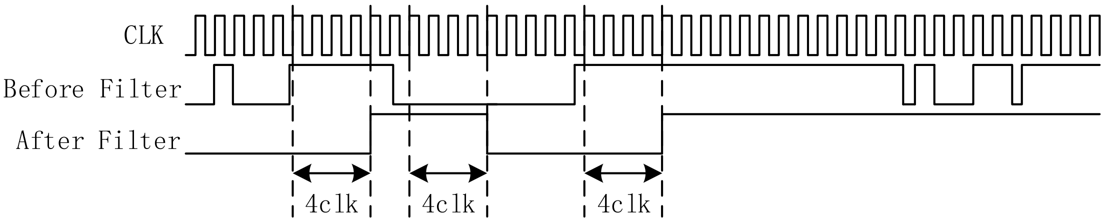


Figure 17-4 4clk Timing Diagram of Filter Module

The filtering circuit removes the input noise with a pulse width of 4/8/16 clock cycles. The filtering feature is enabled when TIMx_CR1[TxFE] = 01/10/11. The filtered signal is 4 ~ 5/8 ~ 9/16 ~ 17 clock cycles later than the signal before filtering.

17.2.5 Input Capture Mode

When TIMx_CR0[TxMOD] = 0, Timer3 works in input capture mode.

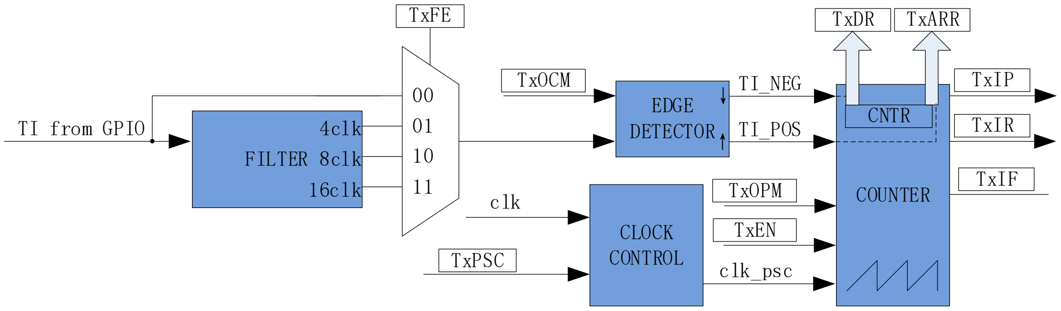


Figure 17-5 Schematic Diagram of Input Capture Mode

The Input Capture Mode detects pulse width and waveform 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.

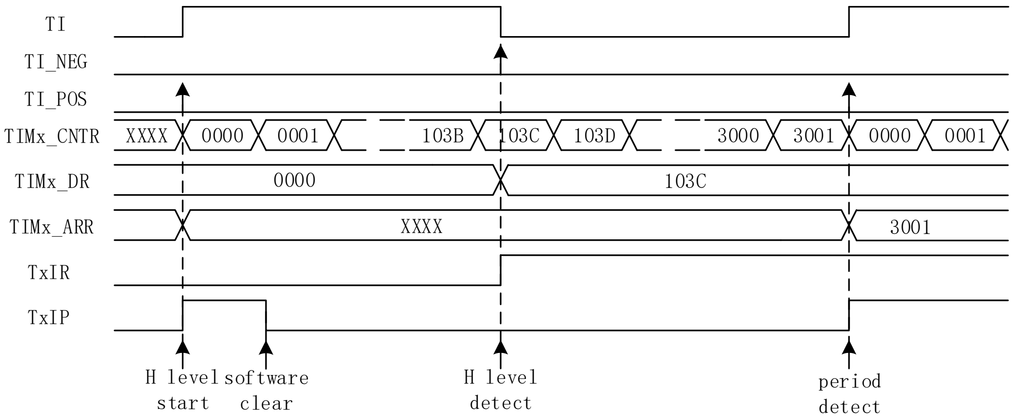


Figure 17-6 Timing Diagram of Input Capture Mode (TIMx_CR0[TxOCM] = 0)

For example, when TIMx_CR0[TxOCM]= 0, TIMx_CR1[TxEN] is set to “1” to enable the timer.

When the falling edge is detected, the value of TIMx__CNTR is stored into TIMx__DR. Meanwhile, the interrupt flag 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 TIMx_CR1[TxIP] is set to “1” and TIMx__CNTR is cleared to “0”. TIMx_CR0[TxOPM] determines whether the timer restarts. If TIMx_CR0[TxOPM] = 1, the timer stops; and if TIMx_CR0[TxOPM] = 0, it restarts.

An overflow event occurs if Timer3/Timer4 does not detect the second rising edge of the input and TIMx__CNTR reaches 0xFFFF. In this case, the interrupt flag bit TIMx_CR1[TxIF] is set to “1”, and TIMx__CNTR is cleared to “0”. TIMx_CR0[TxOPM] determines whether the timer restarts. If TIMx_CR0[TxOCM]= 1, the timer stops counting, and if TIMx_CR0[TxOPM] = 0, it restarts.

17.3 Timer3/Timer4 Registers

17.3.1 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	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description

[7:5]	TxPSC	Base Timer Clock Prescaler Selection
		It is configured to divide the system clock frequency and generate the clock source for Base Timer. When MCU clock runs at 24MHz(41.67ns), the prescaled clock rates are configured as follows:
		000:0x1(24MHz) 001:0x2(12MHz)
		010:0x4(6MHz) 011:0x8(3MHz)
		100:0x10(1.5MHz) 101:0x20(750kHz)
		110:0x40(375kHz) 111:0x80(187.5kHz)

[4]	TxOCM	Output Mode: Output Mode Selection 0: “0” is output when TIMx__CNTR≤TIMx__DR; “1” is output when TIMx__CNTR>TIMx__DR. 1: “1” is output when TIMx__CNTR≤TIMx__DR; “0” is output when TIMx__CNTR>TIMx__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 Detection Interrupt Enable 0: Disable 1: Enable
[2]	RSV	Reserved
[1]	TxOPM	Single Mode The timer stops in any of the following events: Output Mode: Timer overflow event Input Capture Mode: PWM Cycle Detection or Timer Overflow Event 0: The Timer does not stop 1: The Timer stops (TIMx_CR1[TxEN] is reset to “0”)
[0]	TxMOD	Working Mode Selection 0: Input Capture Mode 1: Output Mode

17.3.2 TIM3_CR1 (0x9D/0x9F)

Bit	7	6	5	4	3	2	1	0
Name	TxIR	TxIP	TxIF	TxIDE	TxIFE	TxFE		TxEN
Type	R/W0	R/W0	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]	TxIR	Output Mode: Match Interrupt Flag

		The bit is set to “1” by the hardware when TIM3__CNTR matches TIM3__DR. It is cleared to “0” by software. Input Capture Mode: Pulse Width Detection Interrupt Flag This bit is set to “1” when an input pulse width is detected (Setting TIM3_CR0[T2CES] to select the rising edge or falling edge as the active edge). This bit is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[6]	TxIP	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag This bit is set to “1” when an input PWM cycle is detected (Setting TIMx_CR0[T2CES] to select the rising edge or falling edge as the active edge). It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[5]	TxIF	Output Mode: Timer Overflow Interrupt Flag. This bit is set to “1” and TIMx__CNTR is cleared to “0” when TIMx__CNTR matches TIMx__ARR. It is cleared to “0” by software. Input Capture Mode: Timer Overflow Interrupt Flag This bit is set to “1” and TIMx__CNTR is cleared to “0” when the timer does has not detected an input PWM cycle but the timer TIMx__CNTR reaches 0xFFFF (overflow occurs). It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[4]	TxIPE	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Enable 0: Disable 1: Enable
[3]	TxIFE	Output Mode: Timer Overflow Interrupt Input Capture Mode: Timer Overflow Interrupt Enable 0: Disable 1: Enable
[2:1]	TxFE	Input Signal Filtering Pulse Width Selection Input signals are filtered as noise if pulse width is less than the defined value. Assuming that MCU clock runs at 24MHz(41.67ns): 00: Disable 01: 4 system clock cycles, 4 x 41.67ns 10: 8 system clock cycles, 8 x 41.67ns 11: 16 system clock cycles, 16 x 41.67ns
[0]	TxEN	Base Timer Enable 0: Disable 1: Enable

17.3.3 TIM3__CNTR (0xA2,0xA3/0x92,0x93)

TIM3__CNTRH(0xA3/0x93)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM3__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
TIM3__CNTRL(0xA2/0x92)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM3__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]	TIM3__CNTR	Count values held in the Base Timer When TIM3__CNTR reaches 0xFFFF, this bit is automatically cleared to “0”.

17.3.4 TIM3__DR (0xA4,0xA5/0x94,0x95)

TIM3__DRH(0xA5/0x95)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM3__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
TIM3__DRL(0xA4/0x94)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM3__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]	TIM3__DR	Output Mode: Compare match values (written by software) Input Capture Mode: Count value of the detected input pulse width based on TIMx_CR0[TxOCM] (written by hardware)

17.3.5 TIM3__ARR (0xA6,0xA7/0x96,0x97)

TIM3__ARRH(0xA7/0x97)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM3__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
TIM3__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]	TIM3__ARR	Output Mode: Overload value (written by software) Input Capture Mode: Count value of a PWM cycle based on TIMx_CR0[TxOCM] (written by hardware)

18 Systick

18.1 Systick Instructions

The chip can generate Systick interrupts at a fixed interval, 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 (multiplexed with TIM4 interrupt input).

18.2 Systick Registers

18.2.1 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	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	SYSTIF	Systick Interrupt Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[6]	SYSTIE	Systick Interrupt Enable 0: Disable 1: Enable
[5:0]		See section 19.2.2.

18.2.2 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 Cycle This bit determines the cycle at which Systick interrupts are generated, which defaults to 1ms. Calculation formula is as follows: Systick rate = 24M/(SYST_ARR + 1) Range (0,65535)

19 Driver

19.1 Driver Instructions

19.1.1 Driver Introduction

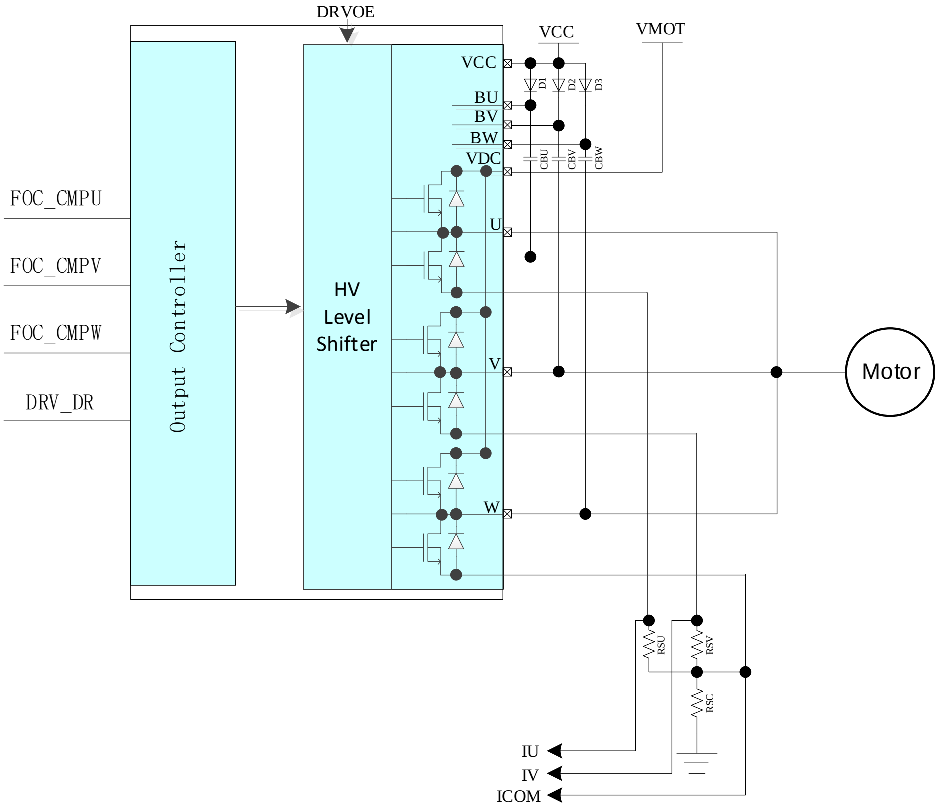


Figure 19-1 Block Diagram of ES9676AH Driver Module

FOC_CMPU/V/W is the three-way comparison value output by FOC module, and DRV_DR is the comparison value set by the software. The above comparison value outputs three sets of level signals U/V/W to HV Level Shifter(ES9676AH) after passing through the output control module. The U/V/W three-way output is applied to BLDC motor control.

DRVOE is the enable signal of HV Level Shifter, and the output of HV Level Shifter is connected with 6 NMOS to drive phase-U/V/W respectively.

Table 19-1 Input and Output of HV Level Shifter

Input		Output	
UH/VH/WH	UL/VL/WL	HU/HV/HW	LU/LV/LW
L	L	L	L
L	H	L	H
H	L	H	L
H	H	L	L

19.1.2 Output Control Module

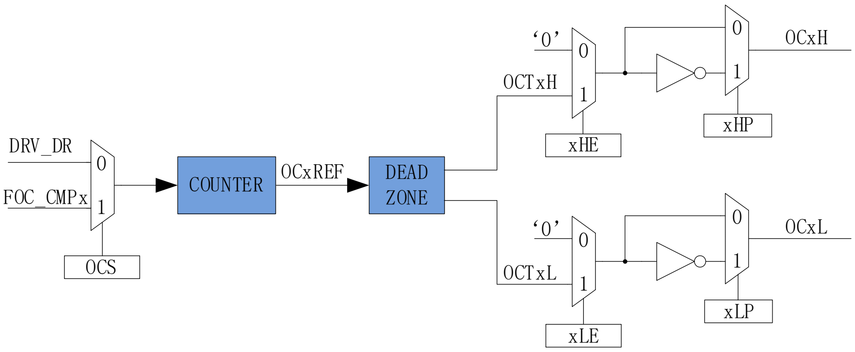


Figure 19-2 Block Diagram of Output Control Module

Before Driver module works, DRV_CR[MESEL] is set to “1” to select FOC/SVPWM/SPWM mode or to “0” to select square-wave control mode.

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

19.1.2.1 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 a 4-way 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.

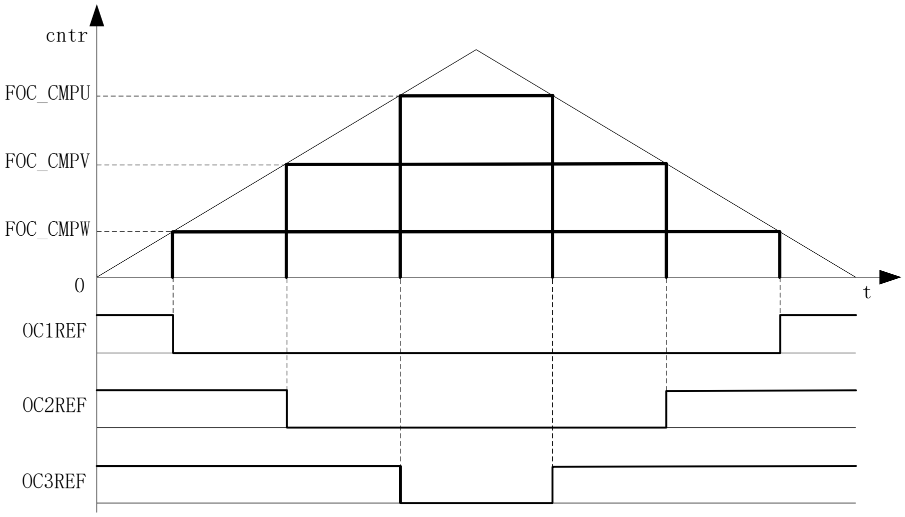


Figure 19-3 PMW Generation

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% (For example, if DRV_ARR=750 and DRV_DR = 375, then duty cycle = 50%)

19.1.2.2 Deadtime Module

OCxREF supports deadtime insertion. For complementary outputs, the deadtime insertion is enabled when DRV_DTR is not “0”. Each channel has an 8-bit deadtime generator, and four channels have the same deadtime, which is set by DRV_DTR. When rising edge signals are detected, output high level of OCxL is delayed for a period of time set in DRV_DTR. When falling edge signals are detected, output high level of OcxH is delayed for a period of time set in DRV_DTR.

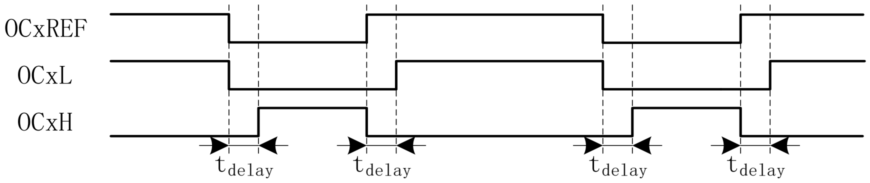


Figure 19-4 Complementary Outputs with Deadtime Insertion

19.1.2.3 Enable and Polarity of Output Signals

DRV_CMR[xHE] and DRV_CMR[xLE] are configured by software to enable high and low sides of the bridge, and DRV_CMR[xHP] and [xLP] to select the polarity of output. For square-wave control, Timer1 automatically controls DRV_CMR to implement phase commutation. Configuring DRV_CR[MESEL] = 0 enables the Square Wave Drive Mode. After Timer1 generates a phase commutation event, the data stored in the corresponding TIM1_DBRx are transferred to DRV_CMR and CMP_CR2[4:3].

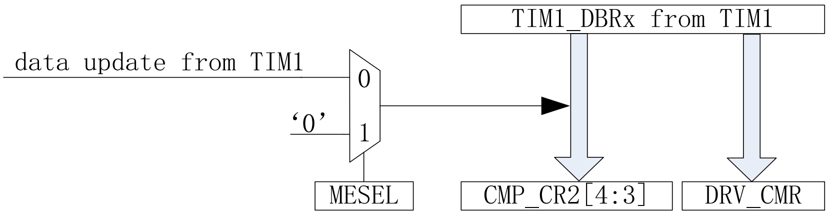


Figure 19-5 Timer1 Automatic Control of DRV_CMR and CMP_CR2[4:3]

With DRV_DR and DRV_ARR registers, DRV_CMR[xHE] and DRV_CMR[xLE] can be configured to implement pre-charging brake, etc. DRV_DR and DRV_ARR control the duty cycle and DRV_CMR[xHE] and DRV_CMR[xLE] control six outputs.

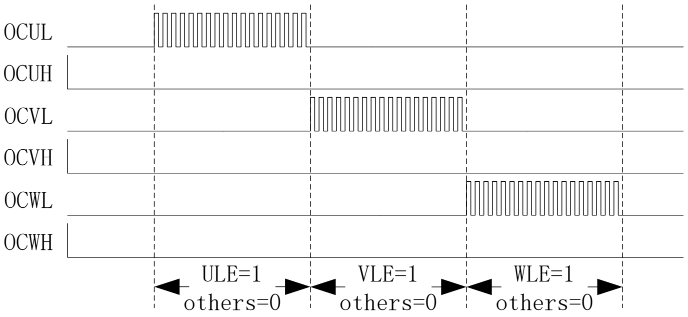


Figure 19-6 Pre-charge Waveform

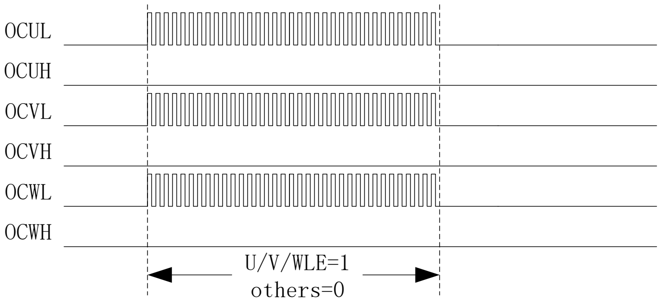


Figure 19-7 Brake Waveform

19.1.2.4 Main Output Enable (MOE)

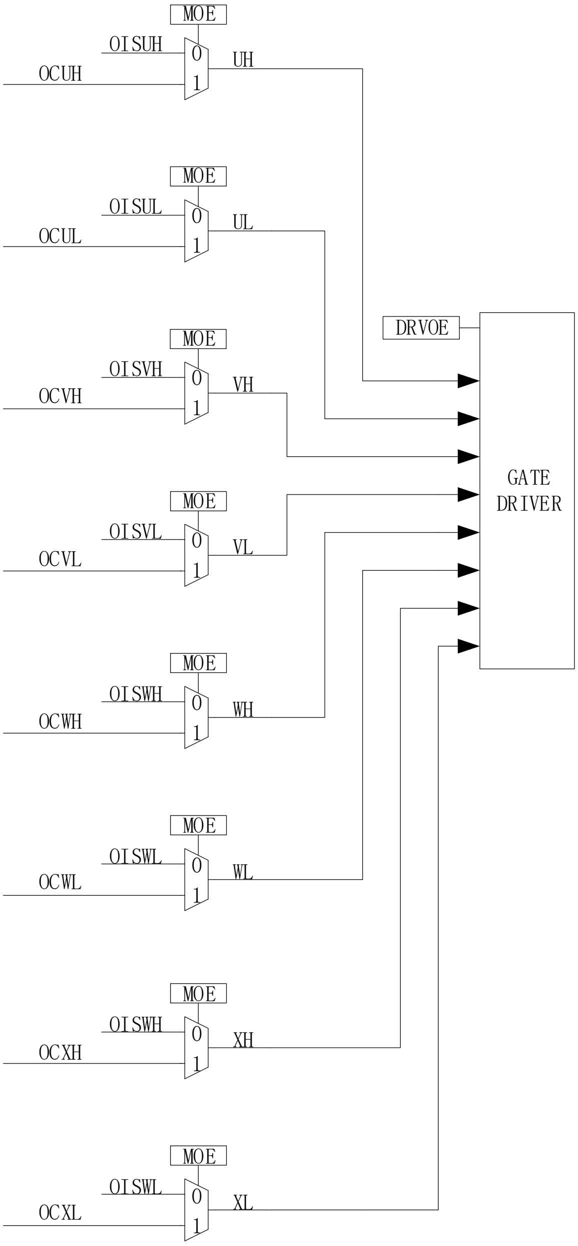


Figure 19-8 Block Diagram of Output Control Module

When DRV_OUT[MOE] is enabled, MOE module uses comparison value of the counter for motor control. When DRV_OUT[MOE] id disabled, the module outputs idle level set by the software to keep the motor at shutdown state.

19.1.2.5 Interrupt

19.1.2.5.1 Compare Match Interrupt

The generation conditions and time for compare match interrupt are configured by DRV_SR[DCIM] and DRV_COMR respectively. When the timer reaches 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. A write of “0” to DRV_SR[DCIF] by software clears the interrupt flag, and a write of “1” has no effect.

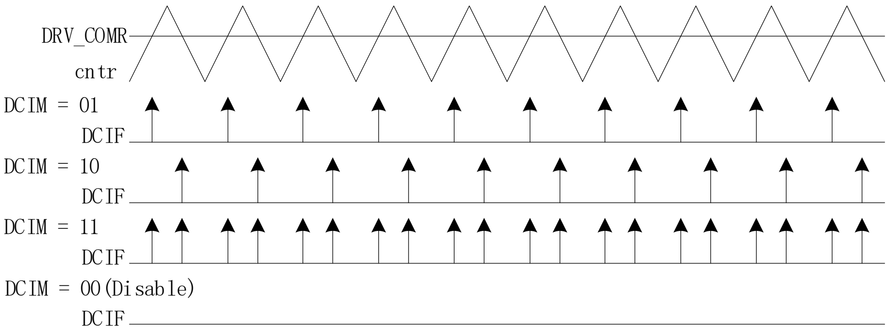


Figure 19-9 Driver Compare Match Interrupt

19.1.2.5.2 FG Interrupt

FG interrupt is enabled when DRV_SR[FGIE] is set to “1”. The motor generates an interrupt for every electrical cycle. After an interrupt event is generated, the interrupt flag is cleared to “0” by software.

19.2 Driver Registers

19.2.1 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/Reverse) This bit controls motor rotation directions and is valid in both square-wave drive and FOC drive modes. 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 Pre-load 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 Drive 1: FOC Drive
[1]	RSV	Reserved
[0]	DRVOE	Driver Enable 0: Disable 1: Enable

19.2.2 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	R/W	R/W	R/W0
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	SYSTIF	SYS TICK Interrupt Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[6]	SYSTIE	SYS TICK Interrupt Enable 0: Disable 1: Enable
[5]	FGIF	FG Interrupt Flag FG interrupt is enabled when DRV_SR[FGIE] is set to “1”.

		The interrupt flag is generated in every electrical cycle under FOC/square-wave control mode. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[4]	DCIF	Driver Match Interrupt Flag When the Driver count value is equal to DRV_COMR, the system decides whether to generate an interrupt according to DRV_SR[DCIM]. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[3]	FGIE	FG Interrupt Enable After the interrupt feature is enabled, an FG Interrupt is generated in each electrical cycle under FOC/square-wave control mode. 0: Disable 1: Enable
[2]	DCIP	Number of PWM cycles to generate an interrupt 0: 1 PWM cycle 1: 2 PWM cycles
[1:0]	DCIM	DRV Compare Match Interrupt Mode Selection When the Driver count value is equal to DRV_COMR, the system decides whether to generate an interrupt according to DRV_SR[DCIM]. 00: No interrupt is generated. 01: An interrupt is generated when the timer counts up. 10: An interrupt is generated when the timer counts down. 11: An interrupt is generated when the timer counts up/down.

19.2.3 DRV_OUT (0xF8)

Bit	7	6	5	4	3	2	1	0
Name	MOE	RSV	OISWXL	OISWXH	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 selects the sources for high and low sides of the bridge of phase-U/V/W/X output signals. It can be set to “1” and “0” by software. When bus current protection occurs, it is automatically cleared to “0” to turn off the output (see section 29.1.1.1). 0: Disable, with output sourced from the idle levels set by DRV_OUT[OISUH]/DRV_OUT[OISVH]/DRV_OUT[OISWH] and DRV_OUT[OISUL]/DRV_OUT[OISVL]/DRV_OUT[OISWL]. 1: Enable, with output sourced from the comparison value of the timer.
[6]	RSV	Reserved
[5]	OISWXL	Output idle level of WL/XL See descriptions on OISUH register
[4]	OISWXH	Output idle level of WH/XH See descriptions on OISUH register
[3]	OISVL	Output idle level of VL See descriptions on OISUH register
[2]	OISVH	Output idle level of VH See descriptions on OISUH register
[1]	OISUL	Output idle level of UL See descriptions on OISUH register
[0]	OISUH	Output idle level of UH This bit sets the UH output in idle state. When DRV_OUT[MOE] = 0, it outputs idle level to disable MOS. 0: Low 1: High

19.2.4 DRV_CMR (0x405C, 0x405D)

Note: For square-wave control, Timer1 automatically controls DRV_CMR register.

DRV_CMRH(0x405C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	XHP	XLP	XHE	XLE	WHP	WLP	VHP	VLP
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
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]	XHP	High-side Polarity Control of phase-X 0: Active High 1: Active Low
[14]	XLP	Low-side Polarity Control of phase-X 0: Active High 1: Active Low
[13]	XHE	High-side Output of phase-X Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-X are complementary and deadtime is automatically added when XLE and XHE are set to “1”. Low-side output is the reference polarity.
[12]	XLE	Low-side Output of phase-X Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-X are complementary and deadtime is automatically added when XLE and XHE are set to “1”. Low-side output is the reference polarity.
[11]	WHP	High-side Polarity Control of W-phase 0: Active High 1: Active Low
[10]	WLP	Low-side Polarity Control of W-phase 0: Active High 1: Active Low
[9]	VHP	High-side Polarity Control of V-phase 0: Active High 1: Active Low
[8]	VLP	Low-side Polarity Control of V-phase 0: Active High 1: Active Low
[7]	UHP	High-side Polarity Control of U-phase 0: Active High 1: Active Low
[6]	ULP	Low-side Polarity Control of U-phase 0: Active High 1: Active Low
[5]	WHE	High-side Output of phase-W Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-W are complementary and deadtime is automatically added when WLE and WHE are set to “1”. Low-side output is the reference polarity.

[4]	WLE	Low-side Output of phase-W Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-W are complementary and deadtime is automatically added when WLE and WHE are set to “1”. Low-side output is the reference polarity.
[3]	VHE	High-side Output of phase-V Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-V are complementary and deadtime is automatically added when VLE and VHE are set to “1”. Low-side output is the reference polarity.
[2]	VLE	Low-side Output of phase-V Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-V are complementary and deadtime is automatically added when VLE and VHE are set to “1”. Low-side output is the reference polarity.
[1]	UHE	High-side Output of phase-U Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-U are complementary and deadtime is automatically added when ULE and UHE are set to “1”. Low-side output is the reference polarity.
[0]	ULE	Low-side Output of phase-U Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-U are complementary and deadtime is automatically added when ULE and UHE are set to “1”. Low-side output is the reference polarity.

19.2.5 DRV_ARR (0x405E, 0x405F)

DRV_ARR(0x405E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DRV_ARR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
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**
[15:0]	DRV_ARR	Timer reload value, which determines PWM frequency (center-aligned) Driver timer up-counts from 0 to DRV_ARR and an overflow event occurs. Then it down-counts to 0. Calculation formula: $f_{Carrier} = f_{mcu}/2/(DRV\_ARR)$ Range (0,4095)

19.2.6 DRV_COMR (0x405A, 0x405B)

DRV_COMRH(0x405A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DRV_COMR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	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**
[15:0]	DRV_COMR	Timer Compare Match Value The compare match interrupt is generated when the count value is equal to DRV_COMR and the conditions set in DRV_SR[DCIM] are met. Range (0, 4095)

19.2.7 DRV_DR (0x4058, 0x4059)

DRV_DRH(0x4058)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	DRV_DR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	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**
[15:0]	DRV_DR	PWM Duty Cycle Setting in Software When DRV_CR[OCS] = 0, DRV_CNTR is compared with DRV_DR to output PWM. “1” is output when DRV_CNTR is smaller than DRV_DR, and “0” is output when DRV_CNTR is larger than DRV_DR. 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. Range (0, 4095)

19.2.8 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 This bit controls deadtime duration between complementary outputs. Assuming that MCU clock runs at 24MHz(41.67ns), DT = (DTR+1) x 41.67ns. Note: When DRV_DTR = 0, deadtime insertion is disabled.

20 WDT

The watchdog timer (WDT) is a timer that works on the internal slow clock (LS_OSC) 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 MCU, and the program starts running from address 0. WDT has to be initialized at regular intervals during operation, and cannot reset the MCU.

20.1 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 timer overflow.

20.2 WDT Operations

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

20.3 WDT Registers

20.3.1 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	0: No effect 1: WDT is initialized.

20.3.2 WDT_REL (0x4027)

Bit	7	6	5	4	3	2	1	0
Name	WDT_REL							
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_REL	This bit sets 8 high-order bits of WDT after initialization.

21 RTC and Clock Calibration

21.1 RTC Functional Block Diagram

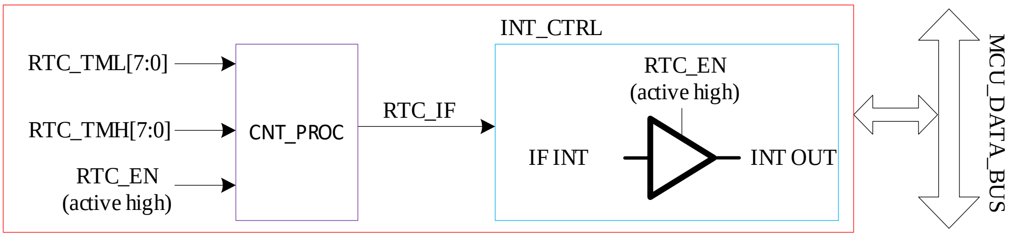


Figure 21-1 RTC Functional Block Diagram

21.2 RTC Operations

A write to RTC_TMH and RTC_TML sets RTC reload value.

RTC is enabled when RTC_STA[RTC_EN] is set to "1".

21.3 RTC Registers

21.3.1 RTC_TM (0x402c, 0x402d)

RTC_TMH(0x402C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RTC0TM[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	RTC0TM[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 Count Register Write: RTC timer up-counts at a rate of 32768Hz from 0 to RTC_TM[15:0] and becomes overflowed . Meanwhile, an interrupt request is generated, causing the timer to be cleared and restart counting. Read: Instantaneous value of the timer

21.3.2 RTC_STA (0x402e)

RTC_STA(0x402E)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RTC_EN	RTCIF	ISOSCSEL	ISOSCEN	RSV			
Type	R/W	R/W	R/W	R/W	-	-	-	-
Reset	0	0	0	0	-	-	-	-

Bit	**Name**	**Description**
7	RTC_EN	RTC Enable 0: Disable 1: Enable
6	RTCIF	RTC Interrupt Flag When IE[RTCIE] is "1", an interrupt event is generated after this bit get overflowed. This bit is cleared to "0" by software. When IE[RTCIE] is "0", an interrupt flag, instead of an interrupt event, is generated after this bit get overflowed. MCU can read and clear the flag.
5	ISOSCSEL	External Source for Clock Calibration 0: Internal Slow Clock 1: P11 Input
4	ISOSCEN	Internal Slow Clock Enable 0: Disable 1: Enable
[3:0]	RSV	Reserved

21.4 Clock Calibration

21.4.1 Introduction

Clock calibration is a feature that uses internal slow clock to calibrate the internal fast clock. Register ISOSCSEL controls the slow clock source, internal slow clock or external clock input. Working principles: A 13-bit timer is used to count the length of 8 slow clock cycles with the fast clock as the clock source.

Calibration operations: Set CAL_CR0[CAL_STA] = 1 in software to start the calibration. Read CAL_CR0[CAL_BUSY] flag bit to check if the calibration process is completed. 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 8 slow clock cycles.

21.4.2 Clock Calibration Register

21.4.2.1 CAL_CR (0x4044, 0X4045)

CAL_CR0, CAL_CR1(0x4040, 0x4041)								
CAL_CR0(0x4040)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	CAL_STA/ CAL_BUSY	RSV		CAL_ARR[12:8]				
Type	R/W	-	-	R	R/W	R/W	R/W	R/W
Reset	1	-	-	0	0	0	0	0
CAL_CR1(0x4041)								
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.
[14:13]	RSV	Reserved
[12:0]	CAL_ARR	Calibration Counts The count values of the fast clock to continuously count eight slow clock cycles When this value is 0, it indicates that no corresponding slow clock input exists, and when this value is 0x1FFF, it indicates that the count overflows (slow clock is too slow or fast clock is too fast).

22 IO

22.1 IO Introduction

ES9676AH has up to 18 GPIO pins, including P0.5 ~ P0.7, P1.0 ~ P1.6, P2.1, P2.4, P2.6 ~ P2.7, P3.0 ~ P3.2 and P3.4 ~ P3.5. Among them, P0.7 and P1.0 share the same pin.

22.2 IO Operations

Each GPIO port pin has relevant configuration 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.

Notes:

- The enable bits of pull-up resistors and pull-down resistors are configured to "1". See 22.3.8 P0_PU (0x4053 ~ 22.3.11 P3_PU (0x4056) for port pins and registers.
- See 5.4 GPIO Electrical Characteristics for the values of pull-up resistors and pull-down resistors.
- Pull-up resistors of P1.3 ~ P1.6 and P2.1 are automatically disabled when the port pins are configured as analog mode.
- The relevant bits of P1_AN, P2_AN and P3_AN registers are configured to "1". See 22.3.5 P1_AN (0x4050) ~ 22.3.7 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.
- IO Priority:
  - GPIO has the lowest priority
  - P0.5: SPI > UART > GPIO
  - P0.6: SPI > UART > GPIO
  - P0.7: TIMER2 > CMP > SPI >GPIO

22.3 IO Registers

22.3.1 P0_OE (0xFC)

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

Bit	Name	Description
[7:5]	P0_OE	P0.7 ~ P0.5 Digital I/O Selection 0: Input 1: Output
[4:0]	RSV	Reserved

22.3.2 P1_OE (0xFD)

Bit	7	6	5	4	3	2	1	0
Name	RSV	P1_OE						
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]	P1_OE	P1.6 ~ P1.0 Digital I/O Selection 0: Input 1: Output

22.3.3 P2_OE (0xFE)

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

Bit	Name	Description
[7:6]	P2_OE	P2.7 ~ P2.6 Digital I/O Selection 0: Input 1: Output
[5]	RSV	Reserved
[4]	P2_OE	P2.4 Digital I/O Selection 0: Input 1: Output
[3: 2]	RSV	Reserved
[1]	P2_OE	P2.1 Digital I/O Selection 0: Input 1: Output
[0]	RSV	Reserved

22.3.4 P3_OE (0xFF)

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

Bit	Name	Description
[7:6]	RSV	Reserved
[5:4]	P3_OE	P3.5 ~ P3.4 Digital I/O Selection 0: Input 1: Output
[3]	RSV	Reserved
[2:0]	P3_OE	P3.2 ~ P3.0 Digital I/O Selection 0: Input 1: Output

22.3.5 P1_AN (0x4050)

Bit	7	6	5	4	3	2	1	0
Name	RSV	P1_AN			HBMOD	RSV		
Type	-	R/W	R/W	R/W	R/W	-	-	-
Reset	-	0	0	0	0	-	-	-

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>[7]</td><td>RSV</td><td>Reserved</td></tr>
<tr><td>[6:4]</td><td>P1_AN</td><td>P1.6 ~ P1.4 Analog Mode Enable
0: Disable
1: Enable</td></tr>
<tr><td>[3]</td><td>HBMOD</td><td>P1.3 mode configuration, which determines the functional mode of P1.3 in combination with P1_OE.3.
<table>
<tr><th>HBMODE</th><th>P1_OE.3</th><th>P1.3 Pin Mode</th></tr>
<tr><td>0</td><td>0</td><td>Digital Input (DI)</td></tr>
<tr><td>0</td><td>1</td><td>Digital Output (DO)</td></tr>
<tr><td>1</td><td>0</td><td>Analog Input (AI)</td></tr>
<tr><td>1</td><td>1</td><td>Digital enhanced drive output mode. The drive mode of high level output provides enhanced drive, and the drive mode of low level output is the same as that of the digital output mode “01”.</td></tr>
</table>
</td></tr>
<tr><td>[2: 0]</td><td>RSV</td><td>Reserved</td></tr>
</table>

22.3.6 P2_AN (0x4051)

Bit	7	6	5	4	3	2	1	0
Name	P2_AN		RSV	P2_AN	RSV		P2_AN	RSV
Type	R/W	R/W	-	R/W	-	-	R/W	-
Reset	0	0	-	0	-	-	0	-

Bit	Name	Description
[7:6]	P2_AN	P2.7 ~ P2.6 Analog Mode Enable 0: Disable 1: Enable
[5]	RSV	Reserved
[4]	P2_AN	P2.4 Analog Mode Enable 0: Disable 1: Enable

[3: 2]	RSV	Reserved
[1]	P2_AN	P2.1 Analog Mode Enable 0: Disable 1: Enable
[0]	RSV	Reserved

22.3.7 P3_AN (0x4052)

Bit	7	6	5	4	3	2	1	0
Name	RSV		P3_AN		RSV	P3_AN		
Type	-	-	R/W	R/W	-	R/W	R/W	R/W
Reset	-	-	0	0	-	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:4]	P3_AN	P3.5 ~ P3.4 Analog Mode Enable 0: Disable 1: Enable
[3]	RSV	Reserved
[2:0]	P3_AN	P3.2 ~ P3.0 Analog Mode Enable 0: Disable 1: Enable

22.3.8 P0_PU (0x4053)

Bit	7	6	5	4	3	2	1	0
Name	P0_PU			RSV				
Type	R/W	R/W	R/W	-	-	-	-	-
Reset	0	0	0	-	-	-	-	-

Bit	Name	Description
[7:5]	P0_PU	P0.7 ~ P0.5 Pull-up Resistor Enable 0: Disable 1: Enable
[4:0]	RSV	Reserved

22.3.9 P1_PU (0x4054)

Bit	7	6	5	4	3	2	1	0
Name	RSV	P1_PU						
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]	P1_PU	P1.6 ~ P1.0 Pull-up Resistor Enable 0: Disable 1: Enable

22.3.10 P2_PU (0x4055)

Bit	7	6	5	4	3	2	1	0
Name	P2_PU		RSV	P2_PU	RSV		P2_PU	RSV
Type	R/W	R/W	-	R/W	-	-	R/W	-
Reset	0	0	-	0	-	-	0	-

Bit	Name	Description

[7:6]	P2_PU	P2.7 ~ P2.6 Pull-up Resistor Enable 0: Disable 1: Enable
[5]	RSV	Reserved
[4]	P2_PU	P2.4 Pull-up Resistor Enable 0: Disable 1: Enable
[3: 2]	RSV	Reserved
[4]	P2_PU	P2.1 Pull-up Resistor Enable 0: Disable 1: Enable
[0]	RSV	Reserved

22.3.11 P3_PU (0x4056)

Bit	7	6	5	4	3	2	1	0
Name	RSV		P3_PU		RSV	P3_PU		
Type	-	-	R/W	R/W	-	R/W	R/W	R/W
Reset	-	-	0	0	-	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:4]	P3_PU	P3.5 ~ P3.4 Pull-up Resistor Enable 0: Disable 1: Enable
[3]	RSV	Reserved
[2:0]	P3_PU	P3.2 ~ P3.0 Pull-up Resistor Enable 0: Disable 1: Enable

22.3.12 PH_SEL (0x404C)

Bit	7	6	5	4	3	2	1	0
Name	SPITMOD	UARTEN	UARTCH	RSV	T3SEL	T2SEL	T2SSEL	RSV
Type	R/W	R/W	R/W	-	R/W	R/W	R/W	-
Reset	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-impedance State
[6]	UARTEN	UART Enable 0: Disable 1: Enable
[5]	UARTCH	UART Function Switching 0: No function switching with P0.6 serving as RXD and P0.5 as TXD. 1: Function switching, with P3.3 serving as RXD and P3.4 as TXD.
[4]	RSV	Reserved
[3]	T3SEL	Port Pins Multiplexed as Timer3 0: P1.1 is multiplexed as GPIO. 1: P1.1 is multiplexed as Timer3 I/O pins.
[2]	T2SEL	Port Pins Multiplexed as Timer2 0: P1.0 is multiplexed as GPIO. 1: P1.0 is multiplexed as Timer2 I/O pins.
[1]	T2SSEL	Port Pins Multiplexed as Timer2 Port 2 0: P0.7 is multiplexed as GPIO. 1: P0.4 is multiplexed as I/O pins ofTimer2 port 2. Note: Timer2 has the highest priority, then the comparator output and SPI MISO.
[0]	RSV	Reserved

22.3.13 P0 (0x80)/P1 (0x90)/P2 (0xA0)/P3 (0xB0) /P4 (0xE8)

Port output register P0/1/2/3/4 supports read and write access. RMW commands are used to access the register value (see Table 22-2 for RMW commands), and other commands are used to access PORT pin.

Table 22-1 P0/P1/P2/P3

Bit	7	6	5	4	3	2	1	0
Name	GPx[7]	GPx[6]	GPx[5]	GPx[4]	GPx[3]	GPx[2]	GPx[1]	GPx[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

Register Name	Function Description	R/W	Initial Value
P0[7:0]	GPIO Register0	R/W	0x00
P1[7:0]	GPIO Register1	R/W	0x00
P2[7:0]	GPIO Register2	R/W	0x00
P3[7:0]	GPIO Register3	R/W	0x00

Table 22-2 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”

23 ADC

23.1 ADC Block Diagram

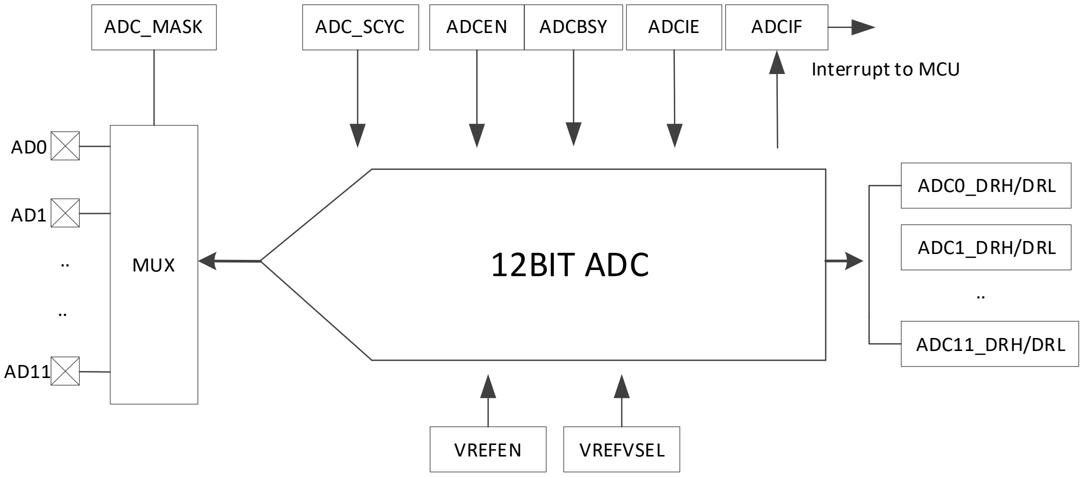


Figure 23-1 ADC Block Diagram

23.2 ADC Operations

ADC module is a 12-bit successive approximation register ADC with 12 channels inside. The sampling mode supports sequential sampling and trigger sampling. The result of sequential sampling are stored in ADCx_DR, and that triggered sampling is sent to FOC module for motor control. If both triggered sampling and sequential sampling are applied at the same time, the triggered sampling is performed first, and ADC automatically stores sequential sampling mode upon completion of triggered sampling.

23.2.1 Sequential Sampling Mode

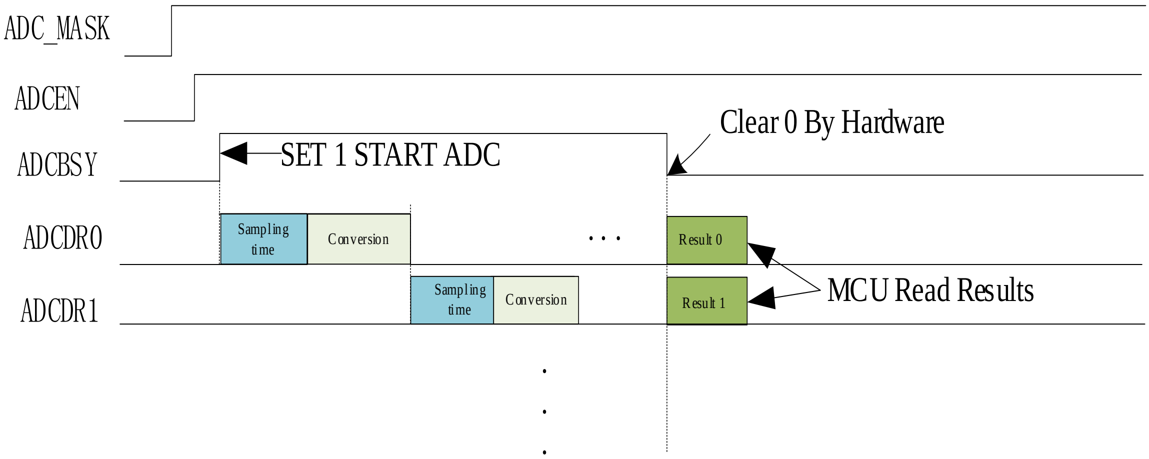


Figure 23-2 ADC Sequential Sampling Timing

ADC operations:

1. Set the appropriate ADC VREF;

2. Configure ADC_MASK to enable the corresponding channel required to sample;
3. Set sampling delay for each channel (minimum value is 3);
4. Configure ADC_CR[ADCEN] = 1 to enable ADC;
5. Configure ADC_CR[ADCBSY] = 1 to start ADC;
6. ADC results are read after ADC_CR[ADCBSY] = 0.
7. The ADC conversion sequence is from low to high based on 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).

The chip integrates a 12-bit ADC with 12 channels. Before using ADC module, ADC_MASK shall be properly configured to enable the corresponding channel required to sample, and sampling delay for each channel (minimum value is 3) shall be defined as well. ADC module starts operating after ADC_CR[ADCEN] and ADC_CR[ADCBSY] are set to “1”.

In addition, ADC module supports ADC triggered sampling, and the priority (triggered by FOC module) is higher than that of sequential sampling (controlled by software).

After FOC module is enabled (DRV_CR[FOCEN] = 1), FOC module automatically starts ADC module and triggers ADC sampling when necessary. The result of triggered sampling is sent to FOC module for motor control.

Note: ADC triggered sampling has a higher priority. If sequential sampling (controlled by software) is being processed and ADC sampling is triggered, ADC module stops sequential sampling to perform ADC triggered sampling, and then continues sequential sampling. If ADC triggered sampling is being processed and sequential sampling is requested, ADC module continues current ADC triggered sampling and performs sequential sampling upon completion of ADC triggered sampling.

23.2.2 ADC Triggered Sampling Mode

After FOC module is enabled, ADC module can initiate ADC triggered sampling when required. ADC triggered sampling and sequential sampling can be requested at the same time. Internal circuit automatically matches the timing for the two different sampling modes, but different ADC channels shall be configured.

After FOC module is enabled (DRV_CR[FOCEN] = 1), FOC module automatically starts ADC module and triggers ADC sampling when necessary. The result of triggered sampling is sent to FOC module for motor control.

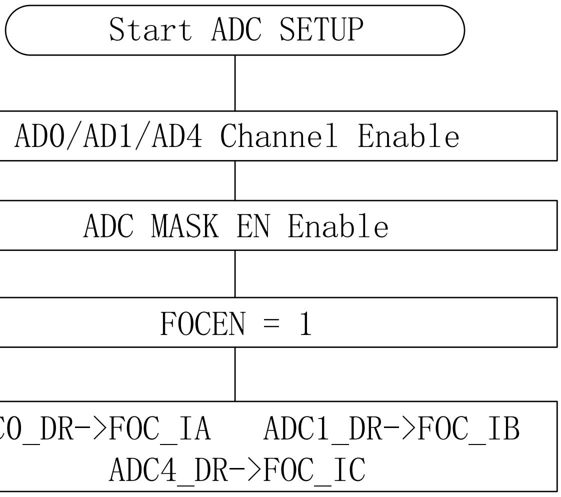


Figure 23-3 ADC Triggered Sampling Process

As shown above, after ADC channel 0/1/4 is enabled and FOC module is enabled, FOC module can trigger ADC triggered sampling and directly read the results.

23.3 ADC Registers

23.3.1 ADC_CR (0x4039)

Bit	7	6	5	4	3	2	1	0
Name	ADCEN	ADCBSY	RSV			ADCALIGN	ADCIE	ADCIF
Type	R/W	R/W1	-	-	-	R/W	R/W	R/W
Reset	0	0	-	-	-	0	0	0

Bit	Name	Description
[7]	ADCEN	ADC Enable 0: Disable 1: Enable
[6]	ADCBSY	ADC Busy Flag ADC_CR[ADCEN] shall be configured first to enable the ADC module, and then ADC_CR[ADCBSY] for ADC conversion. A write of “1” to this bit starts ADC conversion. After ADC conversion, this bit is automatically cleared to “0” by hardware. MCU can read this bit to check the operating status of ADC conversion. When this bit is set to “1”, it has no effect to write “1” to this bit. A write of “0” to this bit has no effect when ADC_MASK=0.
[5:3]	RSV	Reserved
[2]	ADCALIGN	ADC Data Format Selection 0: ADC output is right-aligned 1: ADC output is left-second-high-aligned (invalid in triggered sampling mode)
[1]	ADCIE	ADC Interrupt Enable (excluding triggered sampling mode interrupt) This bit determines whether ADC_CR[ADCIF] sends ADC interrupt to MCU. 0: Disable 1: Enable
[0]	ADCIF	ADC Interrupt Flag After the ADC conversion is completed, if ADC_CR[ADCIE] = 1, an ADC interrupt is generated and sent to MCU. 0: The ADC conversion is not completed. 1: The ADC conversion is completed.

23.3.2 ADC_MASK (0x4036, 0x4037)

ADC_MASKH(0x4036)								
Bit	15	14	13	12	11	10	9	8
Name	ADC_SCYCH[11:8]				CH11EN	CH10EN	CH9EN	CH8EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	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
[11]	CH11EN	ADC Channel 11 Enable See ADC_MASK[CH0EN].
[10]	CH10EN	ADC Channel 10 Enable See ADC_MASK[CH0EN].
[9]	CH9EN	ADC Channel 9 Enable See ADC_MASK[CH0EN].
[8]	CH8EN	ADC Channel 8 Enable See ADC_MASK[CH0EN].

[7]	CH7EN	ADC Channel 7 Enable See ADC_MASK[CH0EN].
[6]	CH6EN	ADC Channel 6 Enable See ADC_MASK[CH0EN].
[5]	CH5EN	ADC Channel 5 Enable See ADC_MASK[CH0EN].
[4]	CH4EN	ADC Channel 4 Enable See ADC_MASK[CH0EN].
[3]	CH3EN	ADC Channel 3 Enable See ADC_MASK[CH0EN].
[2]	CH2EN	ADC Channel 2 Enable See ADC_MASK[CH0EN].
[1]	CH1EN	ADC Channel 1 Enable See ADC_MASK[CH0EN].
[0]	CH0EN	ADC Channel 0 Enable 0: Disable 1: Enable

23.3.3 ADC_MASKH (0x4036)

ADC_MASKH(0x4036)

Bit	7	6	5	4	3	2	1	0
Name	ADC_SCYCH[7:4]				CH11EN	CH10EN	CH9EN	CH8EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	0	0

Bit	Name	Description
[7:4]	ADC_SCYCH[7:4]	ADC Sampling Cycle for ADC Channel 8, 9, 10 and 11 ADC_SCYCH[7] = 0: The sampling cycle is ADC_SCYCH[7:4] ADC clock cycles. ADC_SCYCH[7] = 1: The sampling cycle is (ADC_SCYC [7:4]*8 + 7) ADC clock cycles.
[3]	CH11EN	ADC Channel 11 Enable
[2]	CH10EN	ADC Channel 10 Enable
[1]	CH9EN	ADC Channel 9 Enable
[0]	CH8EN	ADC Channel 8 Enable

23.3.4 ADC_SCYCL (0x4038)

ADC_SCYCL(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 and 7 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, 2 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.

23.3.5 ADC0_DR (0x0300, 0x0301)

ADC0_DRH(0x0300)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC0_DRL(0x0301)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 0 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 0 after ADC conversion

23.3.6 ADC1_DR (0x0302, 0x0303)

ADC1_DRH(0x0302)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV			DH[11:8]				
Type	-	-	-	R	R	R	R	R
Reset	-	-	-	0	0	0	0	0
ADC1_DRL(0x0303)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 1 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 1 after ADC conversion

23.3.7 ADC2_DR (0x0304, 0x0305)

ADC2_DRH(0x0304)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC2_DRL(0x0305)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 2 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 2 after ADC conversion

23.3.8 ADC4_DR (0x0308, 0x0309)

ADC4_DRH(0x0308)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC4_DRL(0x0309)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 4 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 4 after ADC conversion

23.3.9 ADC5_DR (0x030A, 0x030B)

ADC5_DRH(0x030A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC5_DRL(0x030B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 5 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 5 after ADC conversion

23.3.10 ADC7_DR (0x030E, 0x030F)

ADC7_DRH(0x030E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC7_DRL(0x030F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 7 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 7 after ADC conversion

23.3.11 ADC8_DR (0x0310, 0x0311)

ADC8_DRH(0x0310)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC8_DRL(0x0311)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 8 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 8 after ADC conversion

23.3.12 ADC9_DR (0x0312, 0x0313)

ADC9_DRH(0x0312)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC9_DRL(0x0313)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 9 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 9 after ADC conversion

23.3.13 ADC10_DR (0x0314, 0x0315)

ADC10_DRH(0x0314)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC10_DRL(0x0315)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 10 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 10 after ADC conversion

23.3.14 ADC11_DR (0x0316, 0x0317)

ADC11_DRH(0x0316)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC11_DRL(0x0317)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:12]	RSV	Reserved
[11:8]	DH	4 high-order bits of the result by ADC channel 11 after ADC conversion
[7:0]	DL	8 low-order bits of the result by ADC channel 11 after ADC conversion

24 DAC

24.1 DAC Functional Block Diagram

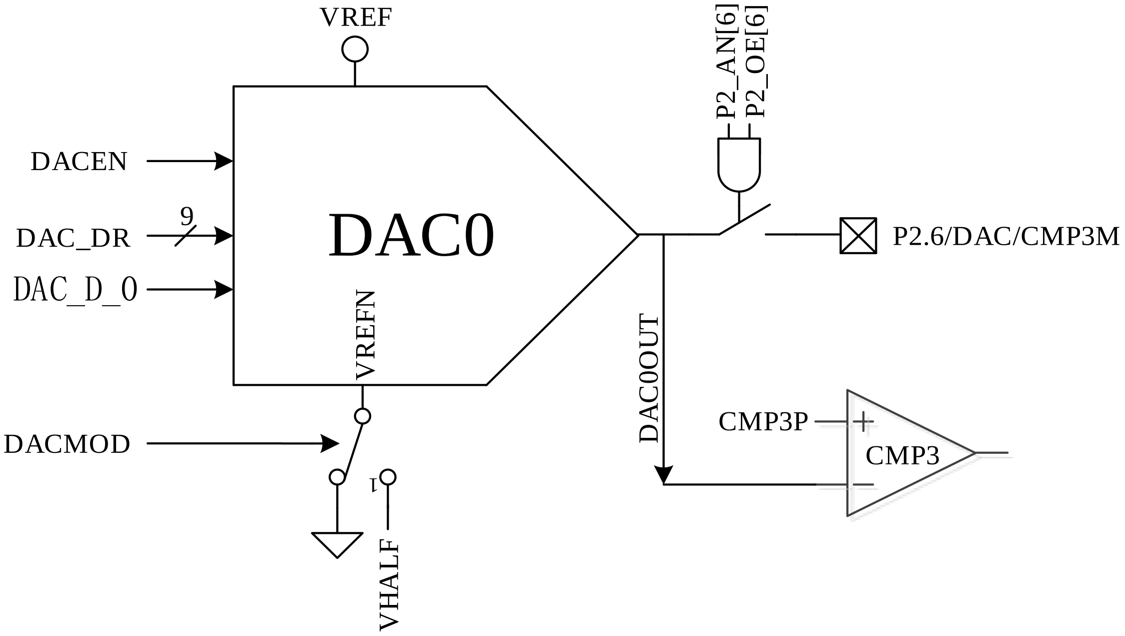


Figure 24-1 DAC Functional Block Diagram

Notes:

- 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.
- Configure P2_AN[6] = 1 and P2_OE[6] = 1, and DAC0 output to P2.6/DAC pin;
- Configure VREF_CR[VREFEN] = 1 and DAC_CR[DACEN] = 1, and VREF is used as DAC0 reference voltage.

24.2 DAC Registers

24.2.1 DAC_CR (0x4035)

Bit	7	6	5	4	3	2	1	0
Name	DACEN	DACMOD	RSV					
Type	R/W	R/W	-	-	-	-	-	-
Reset	0	0	-	-	-	-	-	-

Bit	Name	Description
[7]	DACEN	DAC Enable 0: Disable 1: Enable
[6]	DACMOD	DAC Mode Setting 0: Full-voltage Output Mode. DAC output voltage ranges from 0 to VREF. 1: Half-voltage Output Mode. DAC output voltage ranges from VHALF to VREF.
[5:0]	RSV	Reserved

24.2.2 DAC_DR (0x404B)

Bit	7	6	5	4	3	2	1	0
Name	DAC_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]	DAC_DR	Data Input of DAC Controller

25 DMA

25.1 DMA Instructions

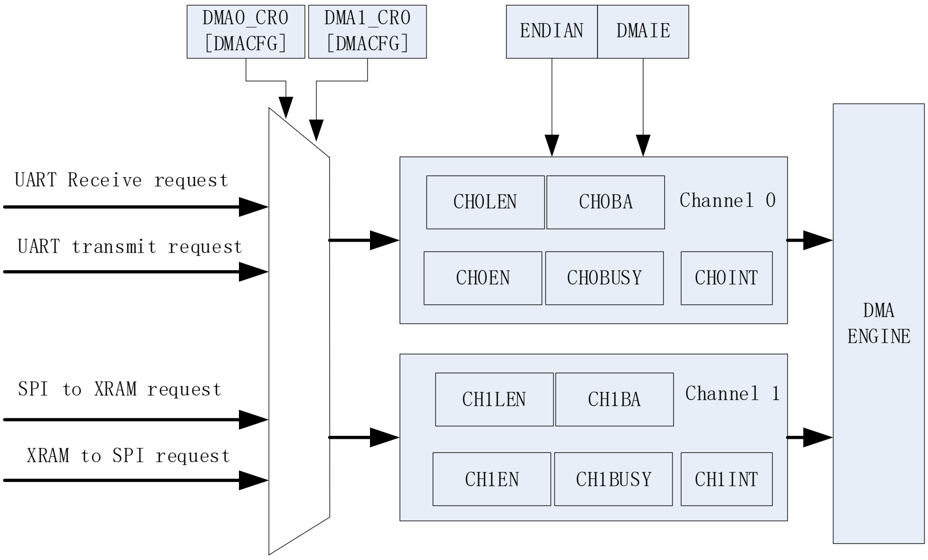


Figure 25-1 DMA Functional Block Diagram

The DMA module is a dual-channel DMA controller, which performs direct data transfer between peripherals (SPI and UART) 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: Configure and enable the peripheral, and set input and output channels taken over by DMA by DMAx_CR0[DMACFG]; and then configure DMA interrupt enable, transfer order, transfer length and XRAM start address. Write “1” to DMAx_CR0[DMAEN] and DMAx_CR0[DMABSY] to start DMA. Clear the interrupt flag bit after data transfer, and set DMAx_CR0[DMABSY] to “1” to start DMA again.

25.2 DMA Registers

25.2.1 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 Start/Busy Flag Read: 0: Channel 0 is idle. 1: Channel 0 is busy. Write: 0: No effect 1: Channel 0 starts for data transfer
[5:3]	DMACFG [2:0]	DMA Channel 0 Peripherals and Transfer Direction Selection 000: From UART to XRAM 001: From XRAM to UART 100: From SPI to XRAM 101: From XRAM to SPI Note: It cannot be configured when Channel 0 is busy.
[2]	DMAIE	DMA Interrupt Request Enable 0: Disable 1: Enable. When the interrupt flag CH0INT or CH1INT is “1”, DMA module sends the interrupt request to MCU.
[1]	ENDIAN	DMA Data Transfer Sequence 0: High bytes are received or sent first 1: Low bytes are received or sent first Note: 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 Event Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending (When INTEN = 1, DMA module sends the interrupt request to MCU. Otherwise, only an interrupt flag is generated.)

25.2.2 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 Start/Busy Flag Read: 0: Channel 1 is idle. 1: Channel 1 is busy. Write: 0: No effect 1: Channel 1 starts for data transfer

[5:3]	DMACFG [2:0]	DMA Channel 1 Peripherals and Transfer Direction Selection 000: From UART to XRAM 001: From XRAM to UART 100: From SPI to XRAM 101: From XRAM to SPI Note: It cannot be configured when Channel 1 is busy.
[2]	DBGSW	Sector Targeted in Debug Mode 0: XSFR as the Debug area (export address space: 0x4020 ~ 0x40FF) 1: XRAM as the Debug area (export address space: 0x0000 ~ 0x0317)
[1]	DBGEN	Debug Mode Enabled 0: Disable 1: Enable DMA module works in Debug mode when DMA1[DMACFG] is set to “101” and DMA1_CR0[DBGEN] to “1”. After SPI is enabled (SPI_CR1[SPIEN] = 1), DMA automatically sends relevant data in the sector defined by DMA1_CR0[DBGSW] via MOSI. CH1BA/CH1LEN defines the start address and range of the relevant data. NSS is automatically pulled low during data transmission. After each transmission, NSS is automatically pulled high. DMA Channel 1 Interrupt is automatically disabled in debug mode.
[0]	DMAIF	DMA Channel 1 Transfer Interrupt Event Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending (When INTEN = 1, DMA module sends the interrupt request to MCU. Otherwise, only an interrupt flag is generated.)

25.2.3 DMA0_CR1 (0x403C)

DMA0_CR1H(0x403C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	CH0LEN						CH0BA[9:8]	
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
DMA0_CR1L(0x403D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CH0BA[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
[15:10]	CH0LEN [5:0]	Read: The number of the byte that is currently transferred by DMA Channel 0 (0 denotes the first byte) Write: XRAM data transfer length of DMA Channel 0 Note: It cannot be configured when Channel 0 is busy. When DMA0_CR0[ENDIAN] = 1 (low bytes are received or transmitted first), it is recommended that CH0LEN be set to an odd number.
[9:0]	CH0BA [9:0]	Start address of XRAM data transfer by DMA Channel 0 It cannot be configured when Channel 0 is busy. Note: XRAM address space for data transfer by Channel 0: CH0BA[9:0] ~ (CH0BA[9:0] + CH0LEN[5:0])

25.2.4 DMA1_CR1 (0x403E)

DMA1_CR1H(0x403E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	CH1LEN						CH1BA[9:8]	
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
DMA1_CR1L(0x403F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CH1BA[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**
[15:10]	CH1LEN [5:0]	Read: The number of the byte that is currently transferred by DMA Channel 1 (0 denotes the first byte) Write: XRAM data transfer length of DMA Channel 1 Note: It cannot be configured when Channel 1 is busy. When DMA_CR0[ENDIAN] = 1, it is recommended that CH1LEN be set to an odd number.
[9:0]	CH1BA [9:0]	Start address of XRAM data transfer by DMA Channel 1 It cannot be configured when Channel 1 is busy. Note: XRAM address space for data transfer by Channel 1: CH1BA[9:0] ~ (CH1BA[9:0] + CH1LEN[5:0]).

26 VREF

26.1 VREF Instructions

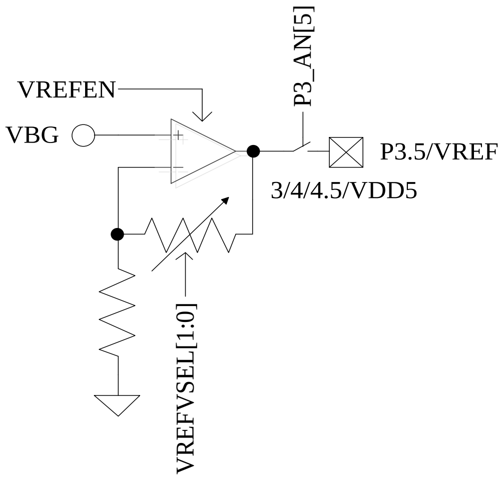


Figure 26-1 I/O Pins of VREF Module

The input and output ports of the VREF module are shown in Figure 26-1. VREF is the voltage reference generation block that provides internal voltage reference to ADC and DAC modules.

VREF is enabled when VREF_VHALF_CR[VREFEN] is set to “1”. The output voltage is selected by configuring VREF_VHALF_CR bit. See section 26.2.1 VREF_VHALF_CR (0x404F) for details. When P3_AN[5] = 1, VREF is output to P3.5 pin.

26.2 VREF Register

26.2.1 VREF_VHALF_CR (0x404F)

Bit	7	6	5	4	3	2	1	0
Name	VREFVSEL		RSV	VREFEN	RSV			VHALF EN
Type	R/W		-	R/W	-	-	-	R/W
Reset	0	0	-	0	-	-	-	0

Bit	Name	Description
[7: 6]	VREFVSEL	VREF Module Output Voltage Selection 01: VDD5 00: 4.5V 11: 4V 10: 3V
[5]	RSV	Reserved
[4]	VREFEN	VREF Module Enable 0: Disable. P3_AN[5] is set to “1”, and external VREF is input from P3.5 pin. 1: Enable. P3_AN[5] is set to “1”, and internal VREF is output to P3.5 pin. A 0.1μF ~ 1μF external capacitor is added to improve the stability of VREF.
[3:1]	RSV	Reserved
[0]	VHALFEN	VHALF Enable 0: Disable 1: Enable

27 VHALF

27.1 VHALF Instructions

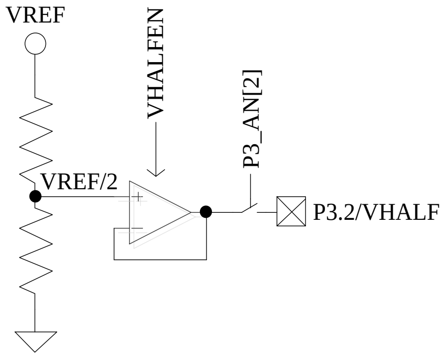


Figure 27-1 I/O Pins of VHALF Module

The input and output ports of VHALF module are shown in Figure 27-1. This module generates the voltage reference.

VHALF is enabled when VREF_VHALF_CR[VHALFEN] is set to “1”. If P3_AN[2] = 1, the voltage is output to P3.2.

27.2 VHALF Register

See section 26.2.1.

28 Operational Amplifier

The chip integrates three high-speed independent operational amplifiers, AMP0, AMP1 and AMP2. Each operational amplifier has a separate enable bit.

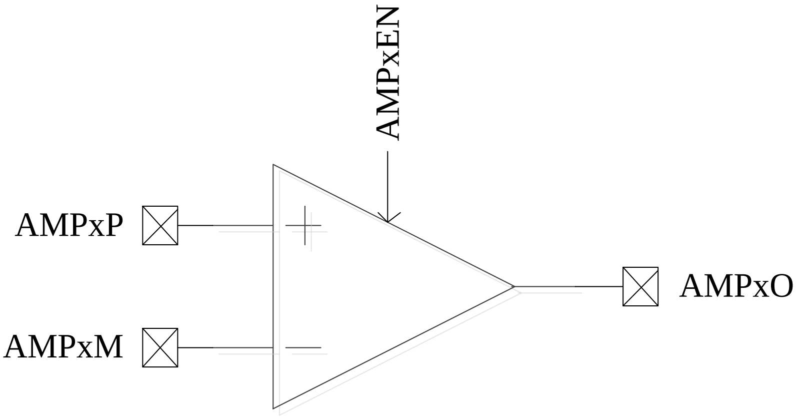


Figure 28-1 Schematic Diagram of Operational Amplifier Module

28.1 Operational Amplifier Instructions

28.1.1 Bus Current Sampling Operational Amplifier (AMP0)

Connection diagram of AMP0 is shown in Figure 28-2.

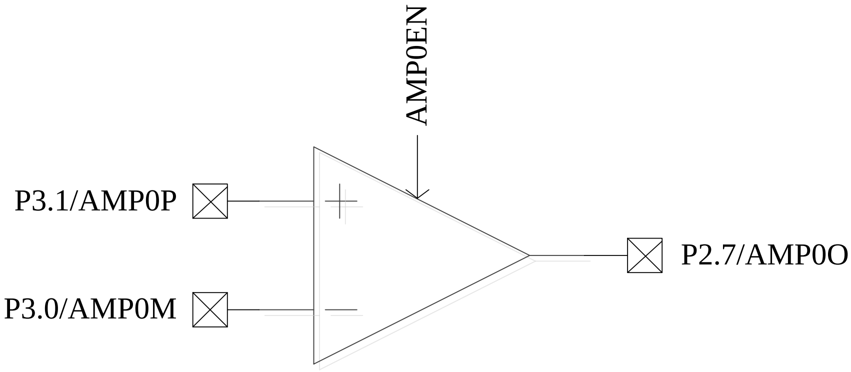


Figure 28-2 AMP0 I/O Pins

AMP0 is enabled when AMP_CR[AMP0EN] = 1.

The I/O pins of AMP0 are shown in Figure 28-2. Before AMP0 is enabled, P2.7, P3.0 and P3.1 shall be automatically configured to analog signal mode, P2_AN[7] to “1” and P3_AN[1:0] to “11”.

28.1.2 Phase Current Operational Amplifier (AMP1/AMP2)

28.1.2.1 AMP1

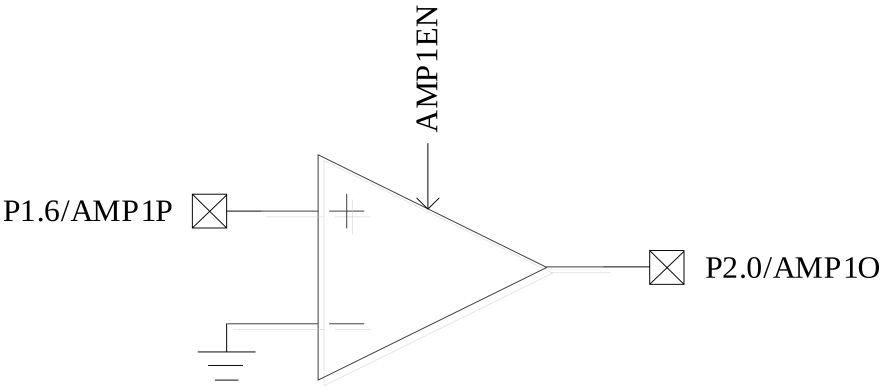


Figure 28-3 AMP1 I/O Pins

AMP1 is enabled when AMP_CR[AMP1EN] = 1.

The I/O pins of AMP1 are shown in Figure 28-3. Before AMP1 is enabled, P1.6 shall be automatically configured to analog signal mode, P1_AN[6] to "11" and P2_AN[0] to "1".

28.1.2.2 AMP2

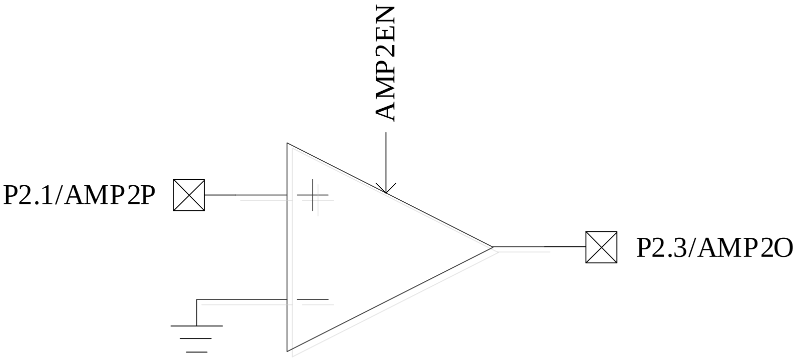


Figure 28-4 AMP2 I/O Pins

AMP2 is enabled when AMP_CR[AMP2EN] = 1.

The I/O pins of AMP2 are shown in Figure 28-4. Before AMP2 is enabled, P2.1 shall be automatically configured to analog signal mode and P2_AN[1] to "111".

28.1.3 PGA Differential Input Mode

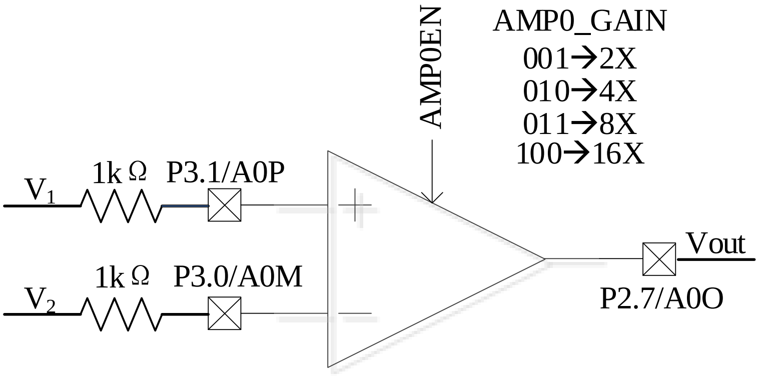


Figure 28-5 AMP0 Operating in PGA Differential Input Mode

In PGA differential input mode, positive and negative inputs of AMP0 are connected with a 1kΩ resistor respectively. The magnification gain can be configured as shown in the figure above.

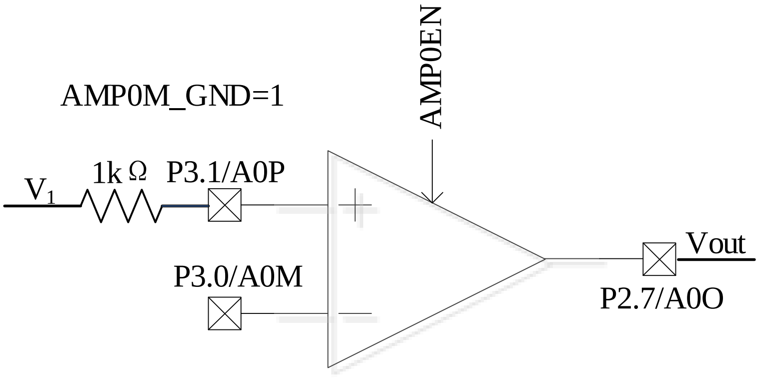


Figure 28-6 AMP0 Operating in PGA Single-ended Input Mode (Negative Input Connected to GND)

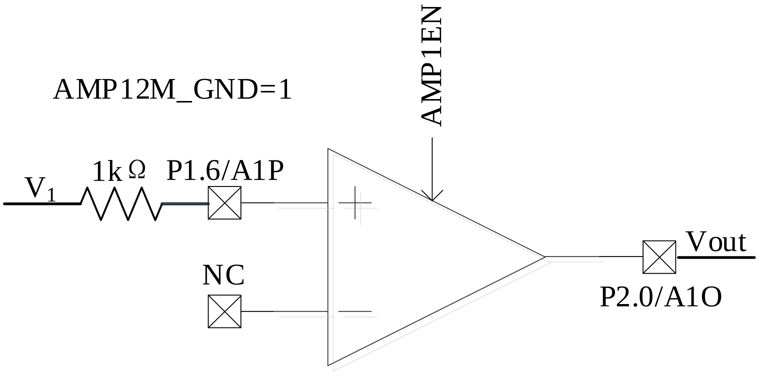


Figure 28-7 AMP1 Operating in PGA Single-ended Input Mode (Negative Input Connected to GND)

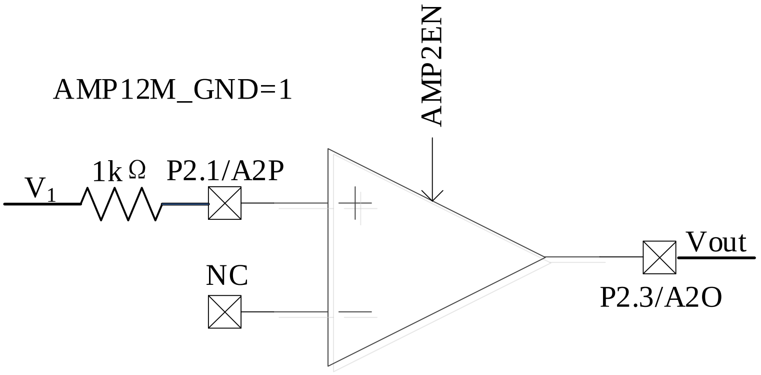


Figure 28-8 AMP2 Operating in PGA Single-ended Input Mode (Negative Input Connected to GND)

When PGA Single-ended Input Mode is selected, negative input of the operational amplifier shall be left floating or connected to GND (after AMP12M_GND/AMP0M_GND is configured in software). In this case, VHALF is equal to 25/64*VREF.

When negative input of the operational amplifier is connected to GND, the output voltage is calculated as follows:

- When AMP_GAIN = 2X, Vout=7/6*V_{half} + 7/3*V1
- When AMP_GAIN = 4X, Vout=6/5*V_{half} + 24/5*V1
- When AMP_GAIN = 8X, Vout=11/9*V_{half} + 88/9*V1
- When AMP_GAIN = 16X, Vout=21/17*V_{half} + 336/17*V1

28.2 Operational Amplifier Registers

28.2.1 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
[1]	AMP1EN	AMP1 Enable
[0]	AMP0EN	AMP0 Enable

28.2.2 CMP_AMP (0x40f2)

Bit	7	6	5	4	3	2	1	0
Name	DAC_D_0	AMP_PH_GAIN[2:0]			AMP0_GAIN[2:0]			CMP3P4M_FS
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]	DAC_D_0	LSB of the 9-bit DAC data
[6:4]	AMP_PH_GAIN[2:0]	Amplification Gain Setting of AMP1&2 (with a 1kΩ external resistor) 000: The gain is configured by external circuit 001: 2X 010: 4X 011: 8X 100: 16X 101: Reserved 110/111: The gain is configured by external circuit
[3:1]	AMP0_GAIN[2:0]	Amplification Gain Setting of AMP0 (with a 1kΩ external resistor) 000: The gain is configured by external circuit 001: 2X 010: 4X 011: 8X 100: 16X 101: Reserved 110/111: The gain is configured by external circuit
[0]	CMP3P4M_FS	Function of CMP3 Positive End/CMP4 Negative End P27 Transferred to P34 0: Disable 1: Enable

28.2.3 TSD_ADJ (0x40f3)

TSD_ADJ(0x40f3)								
Bit	7	6	5	4	3	2	1	0
Name	RSV				AMP12M_GND	AMP0M_GND	TSDADJ3	TSDADJ0_0
Type	-	-	-	-	R/W	R/W	R/W	R/W
Reset	-	-	-	-	0	0	0	0

Bit	Name	Description

<table>
<tr><td>[7:4]</td><td>RSV</td><td>Reserved</td></tr>
<tr><td>[3]</td><td>AMP12M_GND</td><td>After this bit is enabled, negative input P17/P22 of AMP1/AMP2 is connected to GND internally in PGA input mode, and VHALF = 25/64*VREF.
0: Disable
1: Enable</td></tr>
<tr><td>[2]</td><td>AMP0M_GND</td><td>After this bit is enabled, negative input P30 of AMP0 is connected to GND internally in PGA input mode, and VHALF = 25/64*VREF.
0: Disable
1: Enable</td></tr>
<tr><td>[1:0]</td><td>TSDADJ3/
TSDADJ0_0</td><td>After combined with EVT_FILT, TSDADJ3 are the most significant bits, EVT_FILT[6:5] are the medium significant bits and TSDADJ0_0 are the least significant bits.
<table>
<tr><th>TSD_ADJ[3:0]</th><th>Temperature (°C)</th></tr>
<tr><td>1000</td><td>65</td></tr>
<tr><td>1001</td><td>70</td></tr>
<tr><td>1010</td><td>75</td></tr>
<tr><td>1011</td><td>80</td></tr>
<tr><td>1100</td><td>86</td></tr>
<tr><td>1101</td><td>91</td></tr>
<tr><td>1110</td><td>97</td></tr>
<tr><td>1111</td><td>103</td></tr>
<tr><td>0000</td><td>105</td></tr>
<tr><td>0001</td><td>115</td></tr>
<tr><td>0010</td><td>120</td></tr>
<tr><td>0011</td><td>128</td></tr>
<tr><td>0100</td><td>135</td></tr>
<tr><td>0101</td><td>142</td></tr>
<tr><td>0110</td><td>150</td></tr>
<tr><td>0111</td><td>Reserved</td></tr>
</table></td></tr>
</table>

29 Comparator

29.1 Comparator Operations

29.1.1 Comparator CMP3

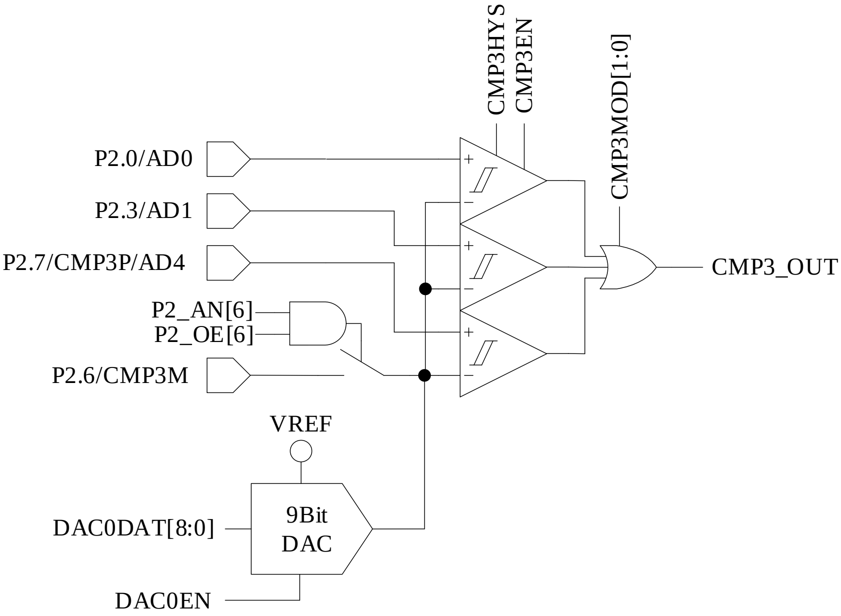


Figure 29-1 CMP3 I/O Pins

The I/O pins of CMP3 are shown in Figure 29-1.

CMP3 configurations:

1. Set P2_AN[6] and P2_OE[6] to "1" to enable CMP3 and VREF on the negative input. The VREF source can be on-chip DAC0 output voltage or external circuit input voltage. Select DAC0 output, and place an external capacitor between P2.6 pin and GND (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 29-2.
   - When CMP_CR1[CMP3MOD] = 01, CMP3 works in dual-comparator input mode. The connection of input and output pins are shown in Figure 29-3.
   - When CMP_CR1[CMP3MOD] = 1X, CMP3 works in triple-comparator input mode. The connection of input and output pins are as shown in Figure 29-4.
3. Configure CMP_CR1[CMP3HYS] to enable or disable hysteresis;
4. Set CMP_CR1[CMP3EN] = 1 to enable CMP3.

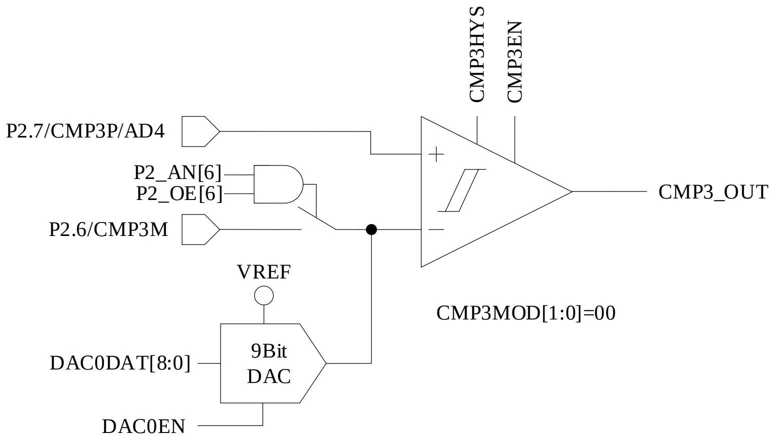


Figure 29-2 Single-comparator Input Mode (CMP_CR1[CMP3MOD] = 00)

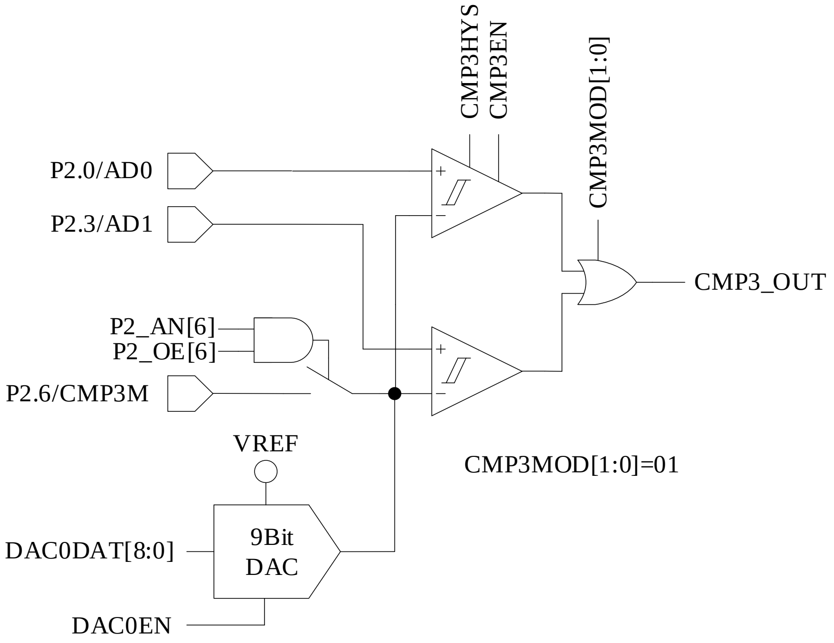


Figure 29-3 Dual-comparator Input Mode (CMP_CR1[CMP3MOD] = 01)

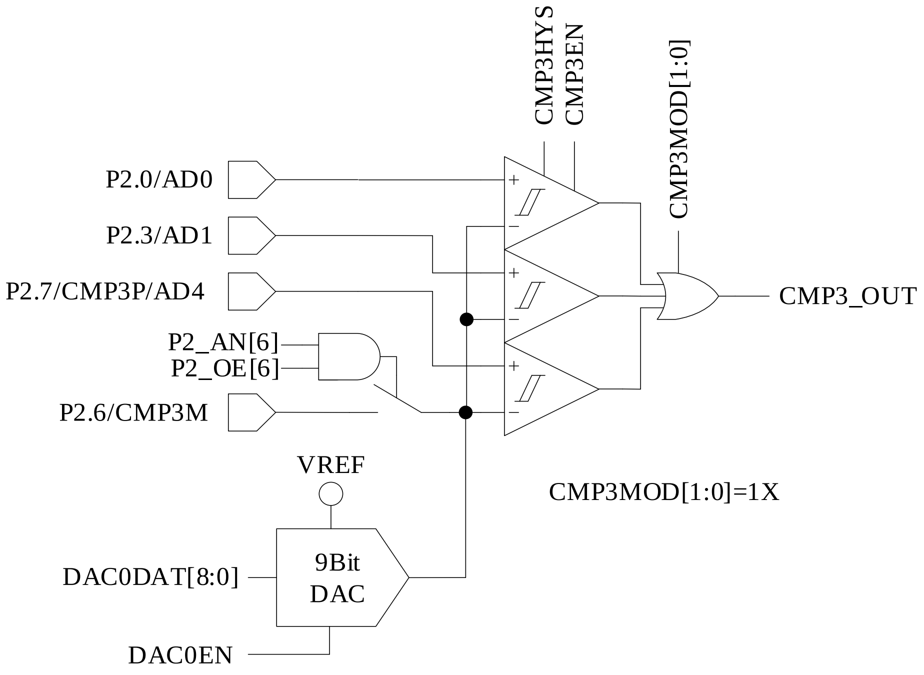


Figure 29-4 Triple-comparator Input Mode (CMP_CR1[CMP3MOD] = 1X)

29.1.1.1 Overcurrent Protection (OCP)

When an overcurrent protection signal is generated, DRV_OUT[MOE] is automatically cleared output idle voltage to stop 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 by configuring EVT_FILT[EFSRC], namely CMP3 interrupt or external interrupt INT0. When EVT_FILT[EFSRC] = 1, 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.

Configuring 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.

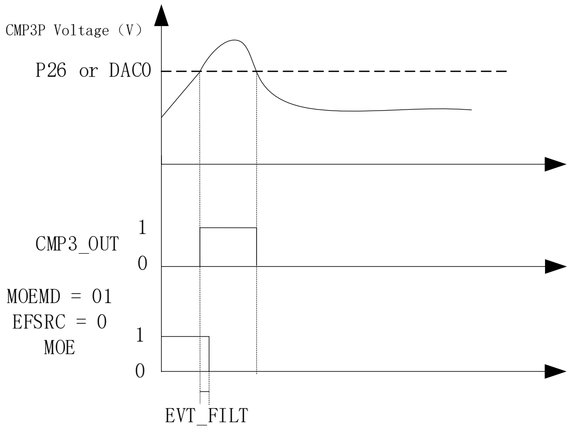


Figure 29-5 MOE Disabled upon Bus Overcurrent Protection

When CMP_CR0[CMP3IM] = 11, the OCP output is generated on the raising edge of CMP3. If EVT_FILT[MOEMD] = 01, the output is automatically turned off when a protection event occurs and a protection interrupt is reported. If EVT_FILT[EFSRC] = 0, the bus current protection signal is generated by CMP3, and sampled voltage on the bus is compared to generate the protection signal. Setting 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 ~ 5/8 ~ 9/16 ~ 17 clock cycles compared to the signal before filtering. As shown in Figure 29-5, when voltage on positive input of the comparator is higher than that on negative input, CMP3_OUT1 is set to “1” to generate CMP3 comparison interrupt for filtering (based on EVT_FILT[EFDIV]). MOE (DRV_OUT[MOE]) is automatically cleared to “0” to turn off six-way outputs to implement overcurrent protection.

29.1.1.2 Cycle-by-cycle Current Limiting

The cycle-by-cycle current limiting feature is applied to square-wave-based drive control of BLDC motors. When EVT_FILT[MOEMD] = 10, MOE is automatically enabled to turn off outputs and DRV_OUT[MOE] is automatically enabled upon DRV timer overflow events. When EVT_FILT[MOEMD] = 11, MOE is automatically enabled to turn off outputs and DRV_OUT[MOE] is automatically enabled upon DRV timer overflow/underflow events or every 5 μs.

CMP_CR0[CMP3IM] must be configured for cycle-by-cycle current limiting, which generates CMP3 interrupt. The priority of CMP3 interrupt can be set to the lowest with an empty process function if the interrupt is not required.

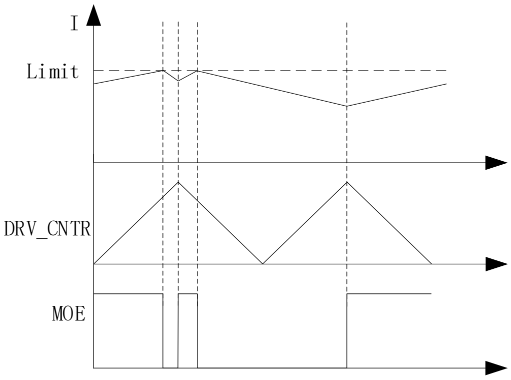


Figure 29-6 Cycle-by-cycle Current Limiting Waveform at EVT_FILT[MOEMD] = 10

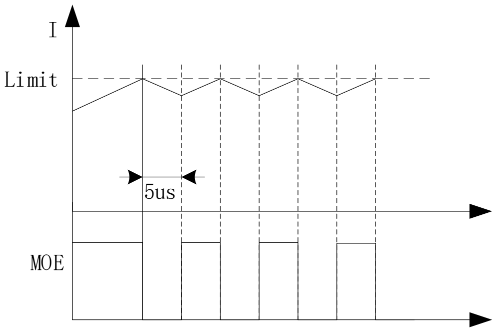


Figure 29-7 Cycle-by-cycle Current Limiting Waveform at EVT_FILT[MOEMD] = 11

29.1.2 Comparator CMP0

CMP0 supports multiple comparison modes for real-time detection on rotor position and motor speed. CMP0, CMP1 and CMP2 are automatically enabled according to the configurations.

Since the three-way output signals CMP0OUT ~ CMP2OUT of CMP0 are first sent to input signal filtering and edge detection modules of Timer1, TIM1_CR3[T1TIS] must be configured as “01” to ensure comparison results held by CMP_SR is generated based on CMP0, instead of Hall signals.

When CMP_CR2[CMP0MOD] = 00, CMP0 works in the mode of three comparators without built-in resistor. The I/O pins are shown in Figure 29-8. It is used for BEMF detection with the external virtual neutral point resistors. The negative inputs of the comparators are connected together to P1.5/CMP0M pin, and the positive inputs are connected to P1.4/CMP0P, P1.6/CMP1P and P2.1/CMP2P respectively. The outputs are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.

Configurations are as follows:

1. Set TIM1_CR3[T1TIS] = 01 to select the comparator as the input;
2. Configure CMP_CR2[CMP0MOD] = 00 to select the mode of three comparators without built-in resistor;
3. Set P1_AN[6:4] = 111 and P2_AN[1] = 1 to configure the port pins as analog mode;
4. Set P1_PU[5:4] = 00 and Reset as 00. Skip this step if no modification is required;

5. Configure CMP_CR1[CMP0HYS] select the hysteresis voltage and Reset as 000;
6. Configure CMP_CR2[CMP0EN] = 1 to enable the comparator;
7. Configure CMP_CR2[CMP0SEL] to select one-way or multiple-way output(s). See Table 29-1;
8. The outputs results are sent to registers CMP2OUT ~ CMP0OUT.

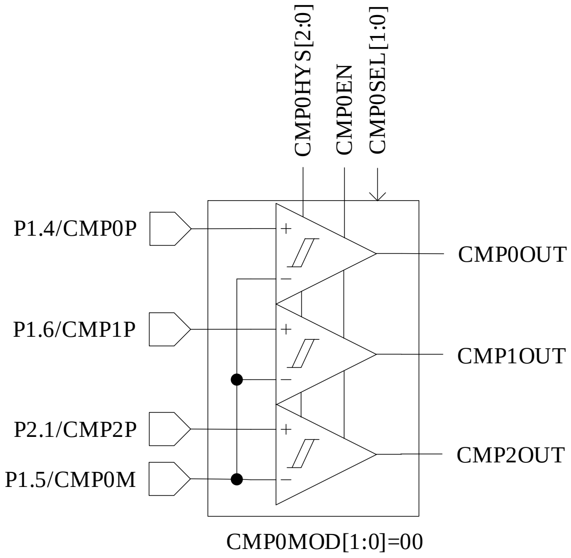


Figure 29-8 CMP0 with Built-in Three Comparators

(without Built-in Resistor)

When CMP_CR2[CMP0MOD] = 01, CMP0 works in the mode of three comparators with built-in resistor. The I/O pins are shown in Figure 29-9. It is used for BEMF detection with the internal virtual neutral point resistors. The negative inputs of the comparators are connected the center point of the built-in resistor, and the positive inputs are connected to P1.4/CMP0P, P1.6/CMP1P and P2.1/CMP2P respectively. The outputs are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.

Configurations are as follows:

1. Set TIM1_CR3[T1TIS] = 01 to select the comparator as the input;
2. Configure CMP_CR2[CMP0MOD] = 01 to select the mode of three comparators with built-in resistor;
3. Set P1_AN[6] = 1, P1_AN[4] = 1 and P2_AN[1] = 1 to configure the port pins as analog mode;
4. Set P1_PU[4] = 0 and Reset as 0. Skip this step if no modification is required;
5. Configure CMP_CR1[CMP0HYS] select the hysteresis voltage and Reset as 000;
6. Configure CMP_CR2[CMP0EN] = 1 to enable the comparator;
7. Configure CMP_CR2[CMP0SEL] to select one-way or multiple-way output(s). See Table 29-1;
8. The outputs results are sent to registers CMP2OUT ~ CMP0OUT.

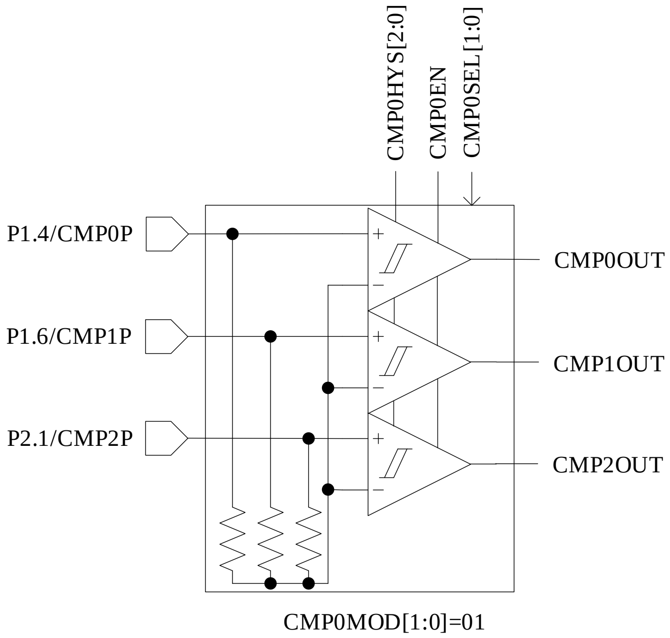


Figure 29-9 CMP0 with Built-in Three Comparators and Resistor

When CMP_CR2[CMP0MOD] = 11, CMP0 works in the mode of dual-comparator. The I/O pins are shown in Figure 29-10. The negative inputs of the comparators are connected together to P1.5/CMP0M, and the positive inputs are connected to P1.4/CMP0P and P1.3/CMP1PS respectively. The outputs are transferred to CMP0OUT and CMP1OUT respectively.

Configurations are as follows:

1. Set TIM1_CR3[T1TIS] = 01 to select the comparator as the input;
2. Configure CMP_CR2[CMP0MOD] = 11 to select dual-comparator mode;
3. Set P1_AN[5:3] = 111 and P1_OE[3] = 0 to configure the port pins as analog mode;
4. Pull-up resistor of P1[5:3] can be enabled (P1_PU[5:3] = 111) or disabled (P1_PU[5:3] = 000) as required, and set Reset as 0. Skip this step if no modification is required. In dual-comparator mode, pull-up resistor of P1[5:3] is generally disabled and used for special applications.
5. Configure CMP_CR1[CMP0HYS] select the hysteresis voltage and Reset as 000;
6. Configure CMP_CR2[CMP0EN] = 1 to enable the comparator;
7. Configure CMP_CR2[CMP0SEL] to select one-way or multiple-way output(s). See Table 29-1;
8. The outputs results are sent to registers CMP1OUT and CMP0OUT.

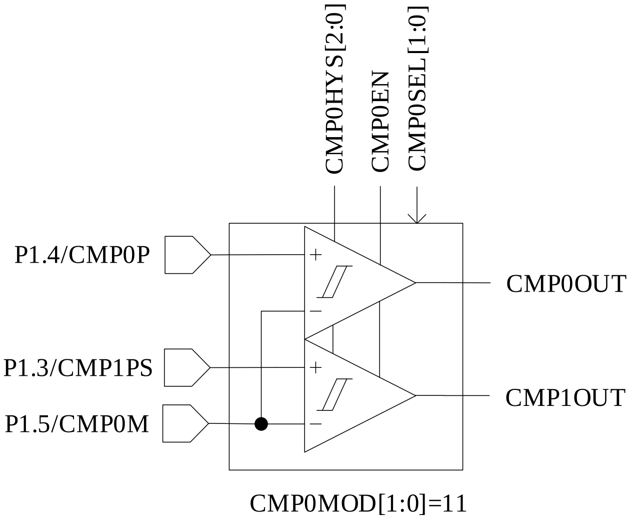


Figure 29-10 CMP0 with Dual-comparator Mode

Hysteresis voltage of CMP0 can be set by register, and shall be properly configured according to actual demand.

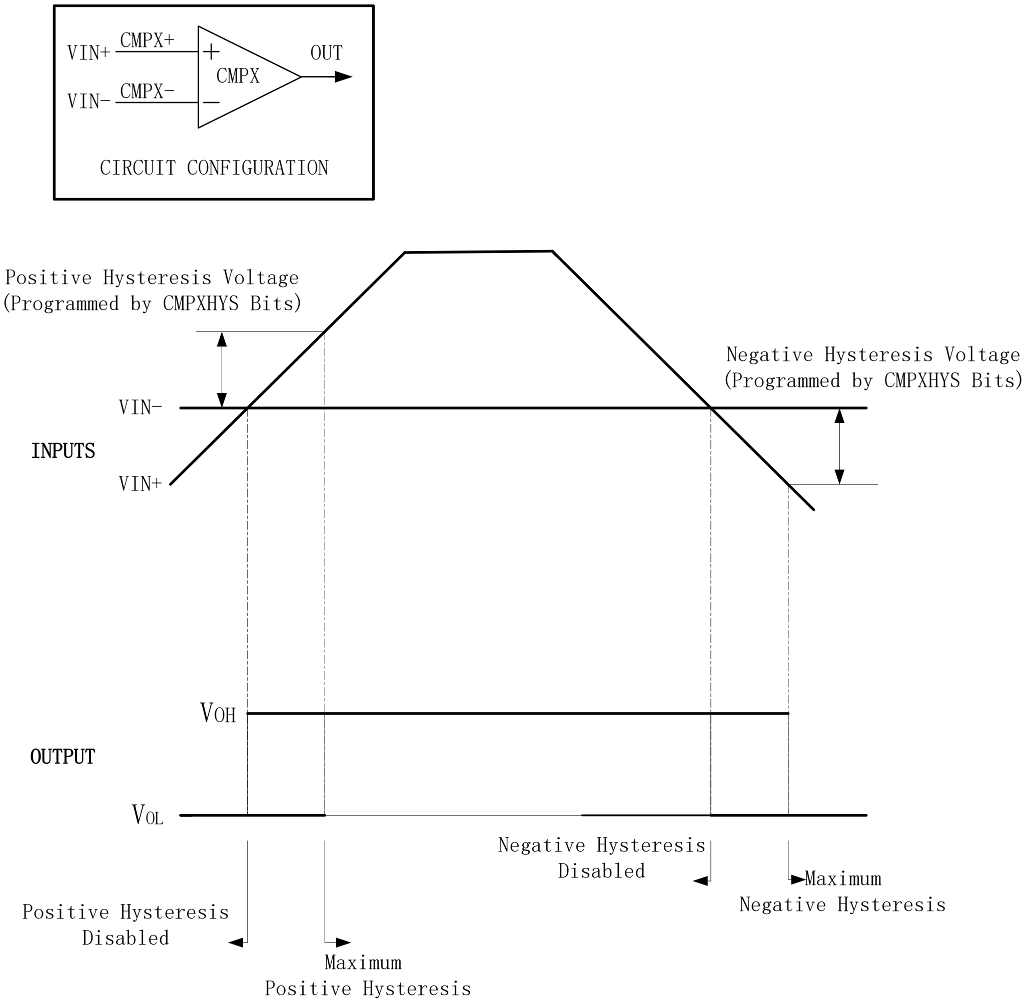


Figure 29-11 CMP0 Input and Output under Positive and Negative Hysteresis Voltage

29.1.3 Comparator Sampling

The comparator sampling feature is mainly used for square-wave control and RSD (tailwind/headwind detection), which eliminates the switching interference from driving circuit. See section Sampling for square-wave control and section RSD Comparator Sampling for RSD.

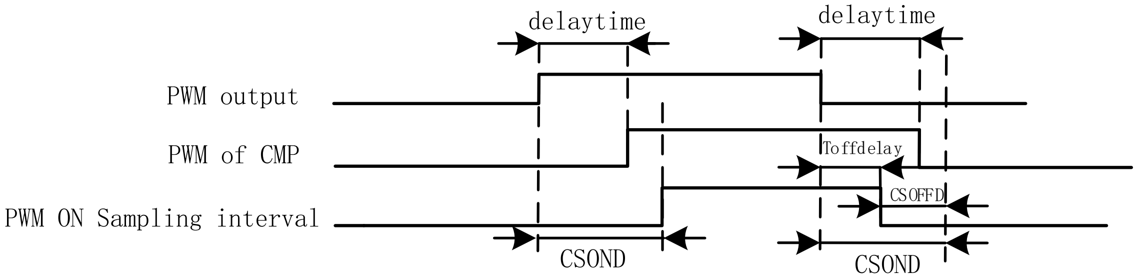


Figure 29-12 PWM ON Sampling Mode

There is a delay from the PWM output to the output of the comparator, which is mainly affected by the following factors: resistance value of drive resistor, MOS on-off speed, and input delay and hysteresis settings of the comparator. As shown in Figure 29-13, the delaytime is from the chip output to the comparator output. When high-level sampling is performed, the sampling interval shall be enveloped by actual high-level output of the comparator. First, the sampling ON-delayed time CMP_SAMR[CSOND] is set to overcome output delay and ringing due to MOS on/off. 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 (PWM of CMP). The sampling OFF-lead time CMP_SAMR[CSOFFD] is set to stop sampling Toffdelay after PWM output falling edge, where Toffdelay = CMP_SAMR[CSOND] – CMP_SAMR[CSOFFD].

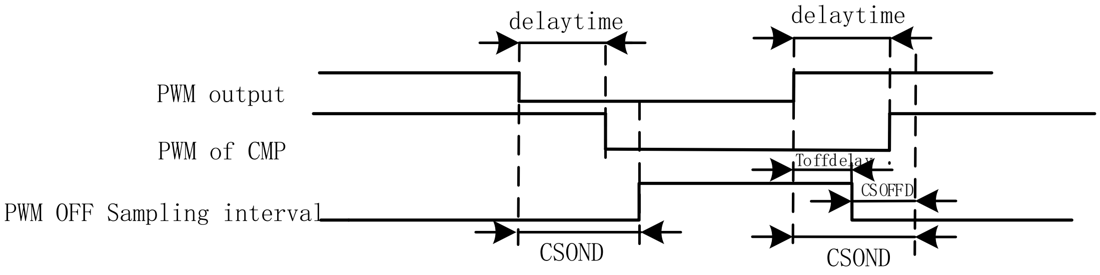


Figure 29-13 PWM OFF Sampling Mode

Similarly, when low-level sampling is performed, ON-delayed time CMP_SAMR[CSOND] and OFF-lead 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. 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.

29.1.4 Comparator Output

CMP_CR3[CMPSEL] is configured to output results of one comparator to a specific pin.

29.2 Comparator Registers

29.2.1 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	
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	CMP3IM	CMP3 Interrupt Mode See descriptions on CMP_CR0[CMP0IM].
[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 is generated. 01: An interrupt is generated upon rising edge. 10: An interrupt is generated upon falling edge. 11: An interrupt is generated upon both rising/falling edges.

29.2.2 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 Input Selection 0: Reserved 1: P1.4/P1.6/P2.1
[6:5]	CMP3MOD	Positive Input of CMP3. See Figure 29-1 for details. The negative input is connected to P2.6 or DAC output. 00: Single-comparator mode, where P2.7 is connected to the positive input, as shown in Figure 29-2. 01: Dual-comparator mode, where P2.0 and P2.3 are connected to the positive input, as shown in Figure 29-3. 1X: Triple-comparator mode, where P2.0, P2.3 and P2.7 are connected to the positive input, as shown in Figure 29-4.
[4]	CMP3EN	CMP3 Enable 0: Disable 1: Enable

[3]	CMP3HYS	CMP3 Hysteresis Voltage Selection 0: No hysteresis 1: Hysteresis voltage is selected
[2:0]	CMP0HYS	CMP0 Hysteresis Voltage Selection: 000: No hysteresis 001: ±2.5mV 010: -5mV 011: +5mV 100: ±5mV 101: -10mV 110: +10mV 111: ±10mV

29.2.3 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
Reset	0	0	0	0	0	-	-	0

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>[7]</td><td>CMP4EN</td><td>CMP4 Enable
0: Disable
1: Enable</td></tr>
<tr><td>[6:5]</td><td>CMP0MOD</td><td>CMP0/1/2 Mode Setting
00: CMP0 with built-in three comparators (without built-in resistor), as shown in Figure 29-8.
01: CMP0 with built-in three comparators and resistors, as shown in Figure 29-9.
11: CMP0 with dual-comparator mode, where only CMP0 and CMP1 work, as shown in Figure 29-10.</td></tr>
<tr><td>[4:3]</td><td>CMP0SEL</td><td>CMP0 Pin Combination Selection, used with CMP_CR2[CMP0MOD] bit. It is set to 00 by default. In square-wave drive application, TIM1_DBRx[T1CPE] automatically controls CMP_CR2[CMP0SEL] to enable or disable each comparator.
Table 29-1 Function Description of CMP0 Port
and CMP_CR2[CMP0MOD] Combination
<table>
<tr><th>CMP0MOD</th><th>CMP0SEL</th><th>Function Description</th></tr>
<tr><td>00</td><td>00</td><td>CMP0/1/2 work simultaneously. The negative input of these comparators are connected to CMP0M. The hardware automatically compares the positive inputs CMP0P, CMP1P and CMP2P with CMP0M, and the output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.</td></tr>
<tr><td></td><td>01</td><td>Only CMP0 works. The positive input is connected to CMP0P, and the negative input to CMP0M. The output results are transferred to CMP0OUT.</td></tr>
<tr><td></td><td>10</td><td>Only CMP1 works. The positive input is connected to CMP1P, and the negative input to CMP0M. The output results are transferred to CMP1OUT.</td></tr>
<tr><td></td><td>11</td><td>Only CMP2 works. The positive input is connected to CMP2P, and the negative input to CMP0M. The output results are transferred to CMP2OUT.</td></tr>
<tr><td>01</td><td>00</td><td>CMP0/1/2 work simultaneously. The negative inputs of these comparators are connected to the center of built-in resistor. The hardware</td></tr>
</table>
</td></tr>
</table>

				automatically compares the positive inputs CMP0P, CMP1P and CMP2P with CMP0M. The output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.
			01	Only CMP0 works. The positive input is connected to CMP0P, and the negative input to the center of BEMF built-in resistor. The output results are transferred to CMP0OUT.
			10	Only CMP1 works. The positive input is connected to CMP1P, and the negative input is connected to the center of BEMF built-in resistor. The output results are transferred to CMP1OUT.
			11	Only CMP2 works. The positive input is connected to CMP2P, and the negative input is connected to the center of BEMF built-in resistor. The output results are transferred to CMP2OUT.
		10	00	CMP0/1/2 work simultaneously. The positive inputs of these comparators are connected to CMP0P, CMP1P and CMP2P respectively, and the negative inputs are connected to CMP0M, CMP1M and CMP2M respectively. The output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.
			01	Only CMP0 works. The positive input is connected to CMP0P, and the negative input to CMP0M, and the output results are transferred to CMP0OUT.
			10	Only CMP1 works. The positive input is connected to CMP1P, and the negative input to CMP1M. The output results are transferred to CMP1OUT.
			11	Only CMP2 works. The positive input is connected to CMP2P, and the negative input to CMP2M. The output results are transferred to CMP2OUT.
		11	00	CMP0/1 work simultaneously. The positive inputs are connected to CMP0P and CMP1PS respectively, and the negative inputs to CMP0M. The outputs results are transferred to CMP0OUT and CMP1OUT respectively.
			01	Only CMP0 works. The positive input is connected to CMP0P, and the negative input to CMP0M. The output results are transferred to CMP0OUT.
			10	Only CMP1 works. The positive input is connected to CMP1PS, and the negative input to CMP0M. The output results are transferred to CMP1OUT.
			11	Reserved
[2:1]	RSV	Reserved		
[0]	CMP0EN	CMP0 Enable 0: Disable 1: Enable		

29.2.4 CMP_CR3 (0xDC)

Bit	7	6	5	4	3	2	1	0
Name	CMPDTEN	DBGSEL		SAMSEL		CMPSEL		
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]	CMPDTEN	Comparator Deadtime Sampling Enable 0: Disable 1: Enable
[6:5]	DBGSEL	Debug Output Selection, connected to P01 pin 00: Debug Output Disable 01: Freewheeling shielding is completed and ZCP signal is detected 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 mode, 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 Signals of one selected comparator is output to the pin. See Comparator Debugging for details. 000: No output 001: CMP0 010: CMP1 011: CMP2 100: CMP3 101: CMP4 110: Reserved 111: ADC result comparison

29.2.5 CMP_SAMR (0x40AD)

Bit	7	6	5	4	3	2	1	0
Name	CMP_SAMR							
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 IC affects input signal of the comparator. In this case, CMP_SAMR[CSOND] is configured to delay the sampling of CMP0/CMP1/CMP2. The delay generated by drive circuit shall be taken into account when CMP_SAMR[CSOND] is calculated. If MCU clock runs at 24MHz(41.67ns) ON-delayed Sampling Time = CSOND x 41.67 x 8ns Notes: ■ CMP_SAMR[CSOND] must be greater than or equal to CMP_SAMR[CSOFFD]. ■ See section Sampling for square-wave control and section RSD Comparator Sampling for RSD.
[3:0]	CSOFFD	CMP0/CMP1/CMP2 OFF-delayed Sampling Time When PWM module switches from OFF to ON or from ON to OFF, turn-on/off of the power IC affects input signal of the comparator. In this case,

		CMP_SAMR[CSOFFD] is configured to reduce comparator interference. If MCU clock runs at 24MHz(41.67ns) OFF-lead Sampling Time = CSOFFD x 41.67 x 8ns Notes: ■ CMP_SAMR[CSOND] must be greater than or equal to CMP_SAMR[CSOFFD]. ■ See section Sampling for square-wave control and section RSD Comparator Sampling for RSD.

29.2.6 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 This bit is set to “1” when CMP3 interrupt is generated. It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[6]	CMP2IF	CMP2 Interrupt Flag This bit is set to “1” when CMP2 interrupt is generated. It is cleared to “0” by software. TIM1_CR3[T1TIS] must be configured to “01”, otherwise, it is the Hall interrupt flag. 0: No Interrupt Pending 1: Interrupt Pending
[5]	CMP1IF	CMP1 Interrupt Flag This bit is set to “1” when CMP1 interrupt is generated. It is cleared to “0” by software. TIM1_CR3[T1TIS] must be configured to “01”, otherwise, it is the Hall interrupt flag. 0: No Interrupt Pending 1: Interrupt Pending
[4]	CMP0IF	CMP0 Interrupt Flag This bit is set to “1” when CMP0 interrupt is generated. It is cleared to “0” by software. TIM1_CR3[T1TIS] must be configured to “01”, otherwise, it is the Hall interrupt flag. 0: No Interrupt Pending 1: Interrupt Pending
[3]	CMP3OUT	CMP3 Comparison Result 0: CMP3 comparison result is 0. 1: CMP3 comparison result is 1.
[2]	CMP2OUT	CMP2 Comparison Result TIM1_CR3[T1TIS] must be configured to “01”, otherwise, the results are generated by Hall signals. 0: CMP2 comparison result is 0. 1: CMP2 comparison result is 1.
[1]	CMP1OUT	CMP1 Comparison Result TIM1_CR3[T1TIS] must be configured to “01”, otherwise, the results are generated by Hall signals. 0: CMP1 comparison result is 0. 1: CMP1 comparison result is 1.
[0]	CMP0OUT	CMP0 comparison result TIM1_CR3[T1TIS] must be configured to “01”, otherwise, the results are generated by Hall signals. 0: CMP0 comparison result is 0. 1: CMP0 comparison result is 1.

29.2.7 EVT_FILT (0xD9)

Bit	7	6	5	4	3	2	1	0
Name	TSDEN	TSDADJ		MOEMD		EFSRC	EFDIV	
Type	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	0	1	1	0	0	0	0	0

Bit	Name	Description
[7]	TSDEN	TSD Enable 0: Disable 1: Enable
[6:5]	TSDADJ	TSD Temperature Selection. See section 0.
[4:3]	MOEMD	MOE Cleared and Enabled by Hardware 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 by hardware upon DRV timer overflow events (for square-wave drive). 11: MOE is automatically cleared and enabled automatically upon DRV timer overflow/underflow events or every 5 μs (for square-wave drive).
[2]	EFSRC	Input Source of Filtering Module Upon Current Protection Event 0: CMP3 interrupt 1: INT0 interrupt
[1:0]	EFDIV	Filter Width for Current Protection 00: Not to filter 01: 4 system clock cycles 10: 8 system clock cycles 11: 16 system clock cycles

30 Power Supply

30.1 LDO

MCU contains two built-in LDO output modules: VDD5 and VDD18.

30.1.1 LDO Operations

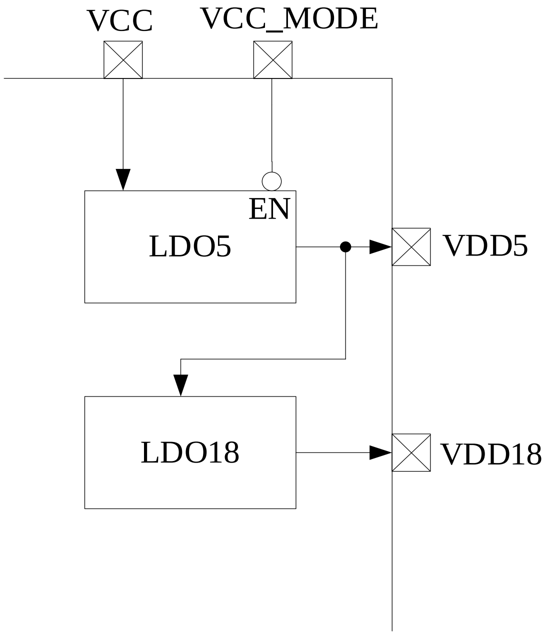


Figure 30-1 Functional Block Diagram of Power Supply

The I/O pins of LDO module is shown in Figure 30-1. The LDO module converts the input supply voltage to 5V VDD5 or 1.8V VDD18 as the power supply for built-in analog and digital modules. Internal LDO5 or external supply for VDD5 is selected by configuring VCC_MODE. In high-voltage single-power supply mode, VCC = 12 ~ 20V as shown in Figure 30-2.

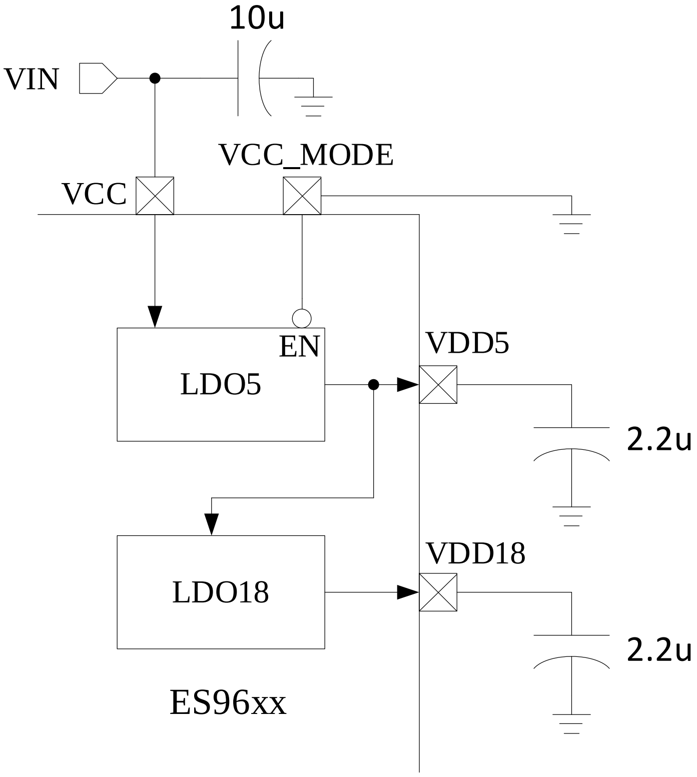


Figure 30-2 High-voltage Single-power Supply Mode

30.2 Low Voltage Detector (LVD)

30.2.1 LVD Introduction

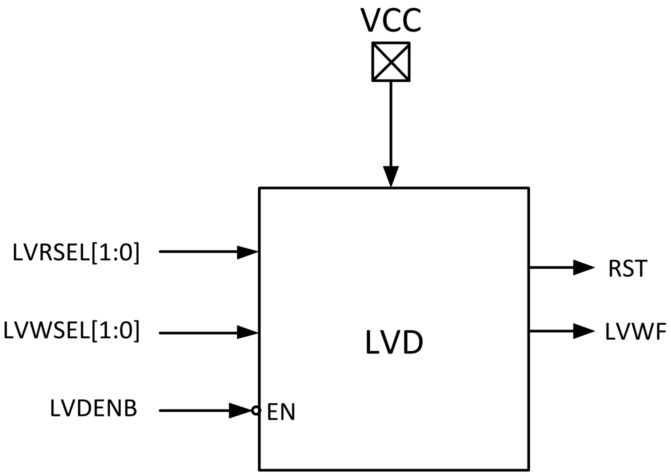


Figure 30-3 LV Detection Module

Configuring CCFG1[LVDENB] = 0 enables LVD module.

30.2.2 LVD Registers

30.2.2.1 CCFG2:RST_MOD (0x401D)

Bit	7	6	5	4	3	2	1	0
Name	LVRSEL		WDTBTEN	WDTRSTEN	RSV		LVWSEL	
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]	LVRSEL	Low Voltage Reset Threshold Voltage. VDD5 is detected for the low voltage reset. 00: Low voltage reset is enabled when VDD5 is lower than 2.8V. 01: Low voltage reset is enabled when VDD5 is lower than 3.0V. 10: Low voltage reset is enabled when VDD5 is lower than 3.5V. 11: Low voltage reset is enabled when VDD5 is lower than 3.8V.
[5:2]		See section 35.1.2.
[1:0]	LVWSEL	Low Voltage Warning Threshold Voltage. VCC is detected for the low voltage warning. 00: Low voltage warning is enabled when VCC is lower than 7V. 01: Low voltage warning is enabled when VCC is lower than 8V. 10: Low voltage warning is enabled when VCC is lower than 9V. 11: Low voltage warning is enabled when VCC is lower than 10V.

30.2.2.2 CCFG1:CK_RST_CFG (0x401E)

Bit	7	6	5	4	3	2	1	0
Name	LVWENB	LVWIE	WDTEN	RSV		FCK_SEL		RSV
Type	R/W	R/W	R/W	-	-	R/W	R/W	-
Reset	0	0	0	-	-	0	0	-

Bit	Name	Description
[7]	LVWENB	Low Voltage Warning Enable 0: Disable 1: Enable
[6]	LVWIE	VCC Low Voltage Warning Interrupt Enable. CCFG2[LVWSEL] decides VCC LV warning voltage threshold. 0: Disable 1: Enable. LV reset detection circuit shall be enabled as well (i.e. CCFG1[LVDENB] = 0).
[5:0]		See section 35.1.1.

30.2.2.3 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/W
Reset	-	-	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:3]	EXT0CFG	INT0 Pin Selection 000: Reserved 001: Reserved 010: Reserved 011: Reserved 100: Reserved 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], and reflects the dynamic over-temperature state.
[1]	LVWF	VCC Low Voltage(LV) Flag This bit indicates whether the chip is in the low voltage state. 0: The chip is not in the LV warning state. 1: The chip is in the LV warning state.
[0]	LVWIF	VCC LV Interrupt Flag This bit indicates whether a low voltage event occurs. It is set to “1” by hardware after low voltage detection interrupt is enabled and an low voltage detection interrupt is detected. A write of “0” in software clears this bit. 0: No Interrupt Pending 1: Interrupt Pending

31 Flash

31.1 Flash Introduction

- 16kB Flash ROM;
- 128 sectors in total, each with a size of 128 bytes
- The first 127 sectors support erasure, write, on-line programing and in-application programming.
- Last sector (the 128th sector) cannot be erased at any time

31.2 Flash Operations

- All interrupts must be disabled before programming to ensure the security of Flash operations and avoid mis-operation of Flash using MOVX instruction during interrupt processing.
- Sector Erasure Operations:
  1. Disable all interrupts by IE[EA];
  2. Write 0x03 to FLA_CR;
  3. Write 5A to FLA_KEY, and then 0x1F to unlock Flash;
  4. Write any value to the Flash sector that is to be erased with MOVX instruction;
  5. Write “1” to FLA_CR[FLAACT] to perform sector erasure. After the command is executed, the Flash is automatically locked.
- Flash Programming Operations:
  1. Disable all interrupts by IE[EA];
  2. Write 0x01 to FLA_CR;
  3. Write 0x5A to FLA_KEY, and then 0x1F to unlock Flash;
  4. Write data to the specified Flash sector with MOVX instruction;
  5. Write “1” to FLA_CR[FLAACT] to program 1-byte data. After this command is executed, data is written and Flash is automatically locked.

     Notes:

     - ➢ All interrupts must be disabled before programming to ensure your program is secured and prevent rom_code is written with MOVX instruction.
     - ➢ It requires quite a long time for the internal circuit to completed Flash operations. For example, it costs about 120ms~150ms to erase a section.
- Each sector has a size of 128 bytes and the last sector (address range: 0x3F80 ~ 0x3FFF) cannot erased at any time. MCU is reset when any instruction in unprotected areas accesses the protected areas (including read, write or erase operations).

31.3 Flash Registers

31.3.1 FLA_CR (0x85)

Bit	7	6	5	4	3	2	1	0
Name	RSV			FLAERR	FLAACT	RSV	FLAERS	FLAWEN
Type	-	-	-	R/W	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 sector erasure succeeds. 1:Programming or sector erasure fails.
[3]	FLAACT	Flash Erase/Write Trigger A write of “0” has no effect and a write of “1” starts Flash operations, including programing, sector erasure, etc.
[2]	RSV	Reserved
[1]	FLAERS	Erase Enable 0: Disable 1: Enable Note: This bit works only when FLA_CR[FLAEN] = 1.
[0]	FLAWEN	Programming Enable 0: Disable 1: Enable

31.3.2 FLA_KEY (0x84)

Bit	7	6	5	4	3	2	1	0
Name	FLA_KEY							
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]	FLA_KEY	Flash Erase/Write Release 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 the FLA_CR register causes the FLA_ KEY to be locked again. Read: The lowest 2 bits reflect internal status, and 6 high-order bits obtains 0x00: 00: Locked 01: Write of 0x5A is done, waiting for 0x1F 10: Frozen 11: Released

32 CRC

32.1 CRC Functional Block Diagram

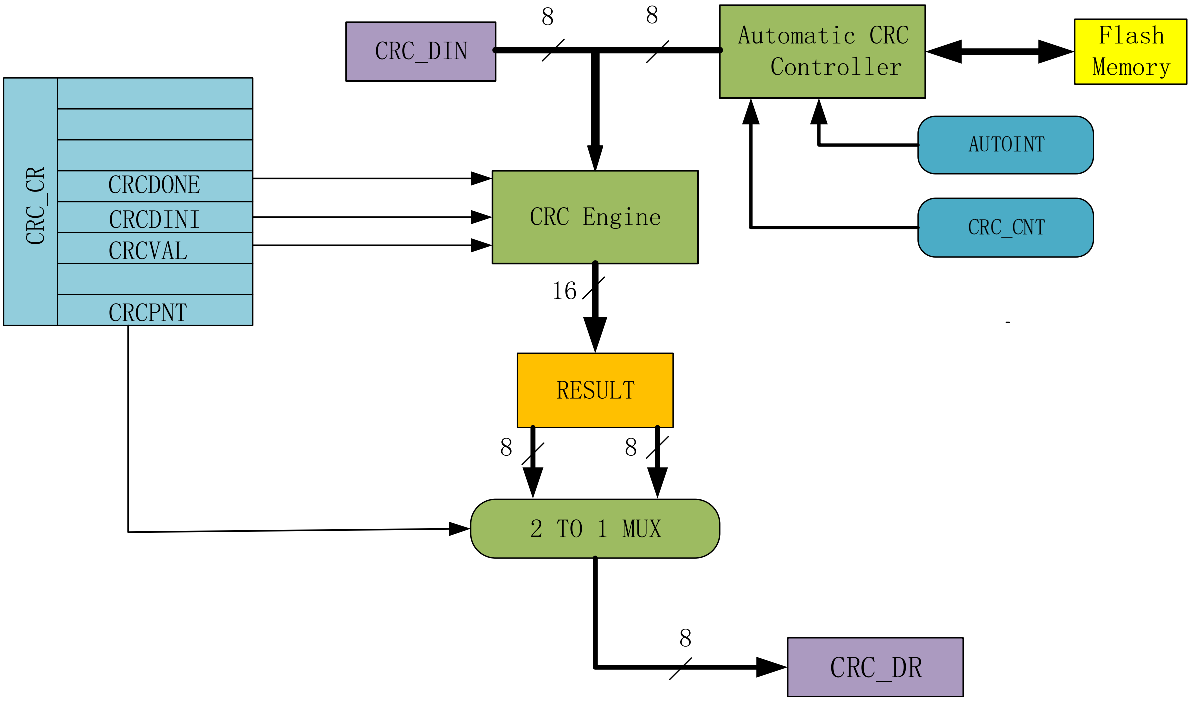


Figure 30-4 CRC Functional Block Diagram

CRC module outputs the result of CRC calculation for any 8-bit data based on a fixed polynomial. As shown in Figure 30-4, 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 30-1 CRC Criteria and Polynomials

S/N.	CRC Criteria	Polynomial	Hexadecimal Representation
1	CRC12	x^12+x^11+x^3+x^2+x+1	80F
2	CRC16	x^16+x^15+x^2+1	8005
3	CRC16-CCITT-FALSE	x^16+x^12+x^5+1	1021
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	04C11DB7

32.2 CRC16 Polynomial

The chip uses CRC16/CCITT-FALSE polynomial: x^16+x^12+x^5+1.

32.3 CRC16 Logic Diagram

The schematic diagram of CRC16 is shown in Figure 30-5. The chip implementation is based on parallel algorithm. For each input byte, MCU calculates the results within one system clock cycle.

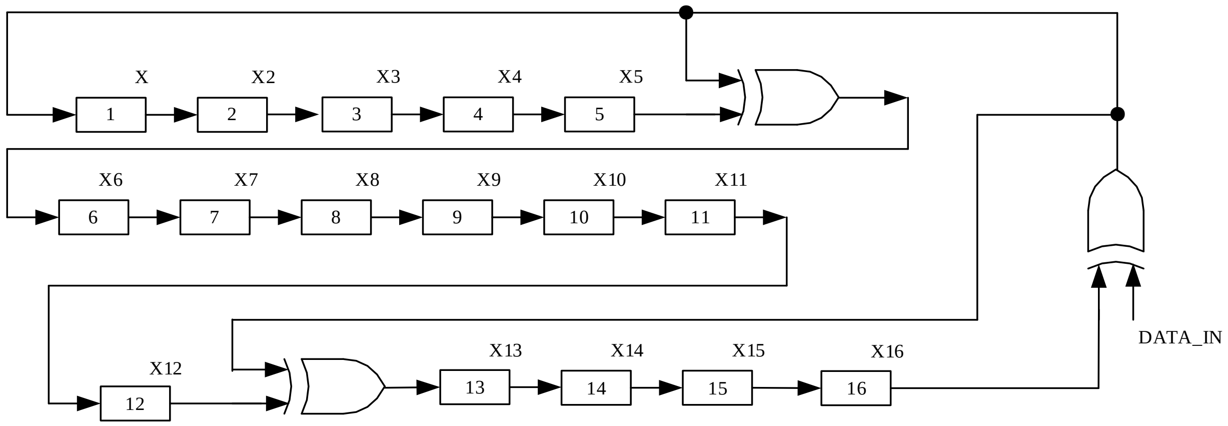


Figure 30-5 CRC16 Schematic Diagram

32.4 CRC Operations

32.4.1 CRC Calculation of 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 (0x63 for example) to CRC_DIN, and the CRC calculation is completed in the next clock cycle;
3. Read CRC results: Configure CRC_CR[CRCPNT] = 1, and read off CRC_DR in software to gets the high bytes. Configure CRC_CR[CRCPNT] = 0, and read off CRC_DR to get the low bytes.

32.4.2 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.

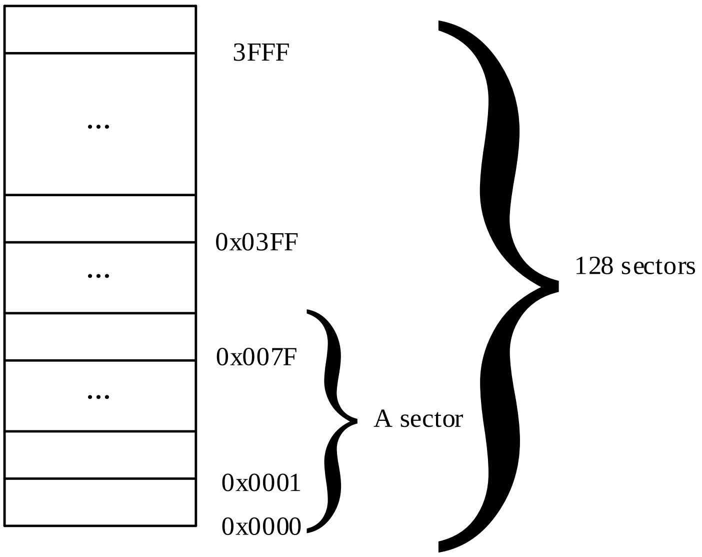


Figure 30-6 ROM Sectors

As shown in Figure 30-6, ROM contains 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 falling between 0x00 and 0x00 ~ 0x7F, including 0x00 and 0x7F. The CRC_CNT (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 CRC_CNT is large (0x01 or larger), CRC controller will automatically limits the number of sectors to be calculated. Finally, CRC module only calculates CRC value of the last sector.

32.5 CRC Registers

32.5.1 CRC_CR (0x4022)

Bit	7	6	5	4	3	2	1	0
Name	RSV			CRCDONE	CRCDINI	CRCVAL	AUTOINT	CRCPNT
Type	-	-	-	R	R/W	R/W	R/W	R/W
Reset	-	-	-	1	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	CRCDONE	CRC Sector Calculation Completion Flag During the calculation, this bit is automatically set to “0” 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.

		When “1” is written to this bit by software, the hardware does not actually write “1” to this bit but synchronously generates a high level pulse of a clock cycle, which is sent to CRC engine for the initialization of CRC results. Therefore, it always returns “0” when it reads, no matter what value is written by the software.
[2]	CRCVAL	CRC Result Initialization Selection 0: CRC result is initialized to 0x0000. 1: CRC result is initialized to 0xFFFF.
[1]	AUTOINT	CRC Sector Calculation Launch A write of “1” to this bit launches CRC batch calculation with CRC_BEG as the start block. A total of CRC_CNT blocks are to be calculated. Note: Other bits shall be configured first. In other words, this bit cannot be configured at the same time as other bits.
[0]	CRCPNT	CRC Result Pointer 0: Read CRC_DR to access 8 low-order bits of the 16-bit CRC result 1: Read CRC_DR to access 8 high-order bits of the 16-bit CRC result

Note: CRC calculation can be performed on single byte or ROM sector. When CRC_CR[AUTOINT] is set to “1”, ROM sector CRC calculation is implemented. When CRC_CR[AUTOINT] is set to “0”, single-byte CRC calculation is implemented.

32.5.2 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: It is a virtual register, so the written data is not saved. 0x00 is returned when the address is accessed.

32.5.3 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, the configuration of CRC_CR[CRCPNT] determines whether to access the high-order or low-order bits of the CRC result.

Note: Because the value of this register can be changed by other signals, this register is placed directly inside the CRC module, instead of register-specific module.

32.5.4 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
Reset	-	-	-	-	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6:0]	CRC_BEG	First ROM Sector Pending Automatic CRC Calculation Example: If CRC_BEG is set to “1”, CRC calculation starts from location 1*128 = 128, or rather from the first byte of sector 2.

32.5.5 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.

33 Sleep Mode

33.1 Introduction

The chip operates in three modes: normal mode, standby mode and sleep mode. 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 30-2.

Table 30-2 Power Consumption Modes

Power Mode	Description	Wakeup 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 other functional modules are either enabled or disabled depending on their control bit setting. WDT stops.	Any interrupt Reset/Debug on external interrupt	Low power performance with flexible performance
Sleep	The analog fast clock circuit is disconnected and software shall ensure that ADC, FOC, and driver modules are disabled before the chip enters the Sleep Mode. WDT is disabled.	External interrupt, RTC interrupt, Reset/Debug on external interrupt	Extremely low power performance with flexible performance

Note: It is recommended to insert 3 null statements in the Sleep mode.

```
PCON = 0x02;
_nop_();
_nop_();
```

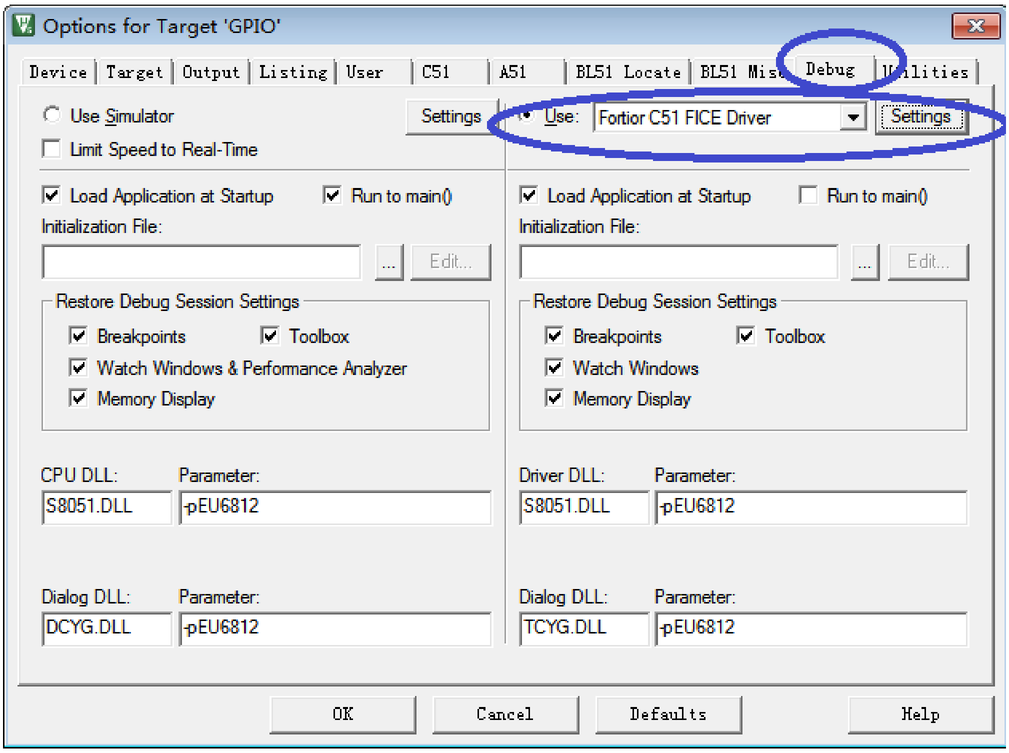
33.2 Sleep Mode Register

33.2.1 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	R/W
Reset	-	-	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
5	GF3	General-purpose flag bit 3
4	GF2	General-purpose flag bit 2
3	GF1	General-purpose flag bit 1
2	RSV	Reserved
1	STOP	A write of “1” makes the chip enter the sleep mode. The bit is automatically cleared to “0” by hardware after wakeup.
0	IDLE	A write of “1” makes the chip enter the sleep mode. The bit is automatically cleared to “0” by hardware after wakeup. Power Consumption Mode: {STOP，IDLE} = 1x, the system is in sleep mode. {STOP，IDLE} = 01, the system is in standby mode. {STOP，IDLE} = 00, the system is operates normally.

34 Code Protection



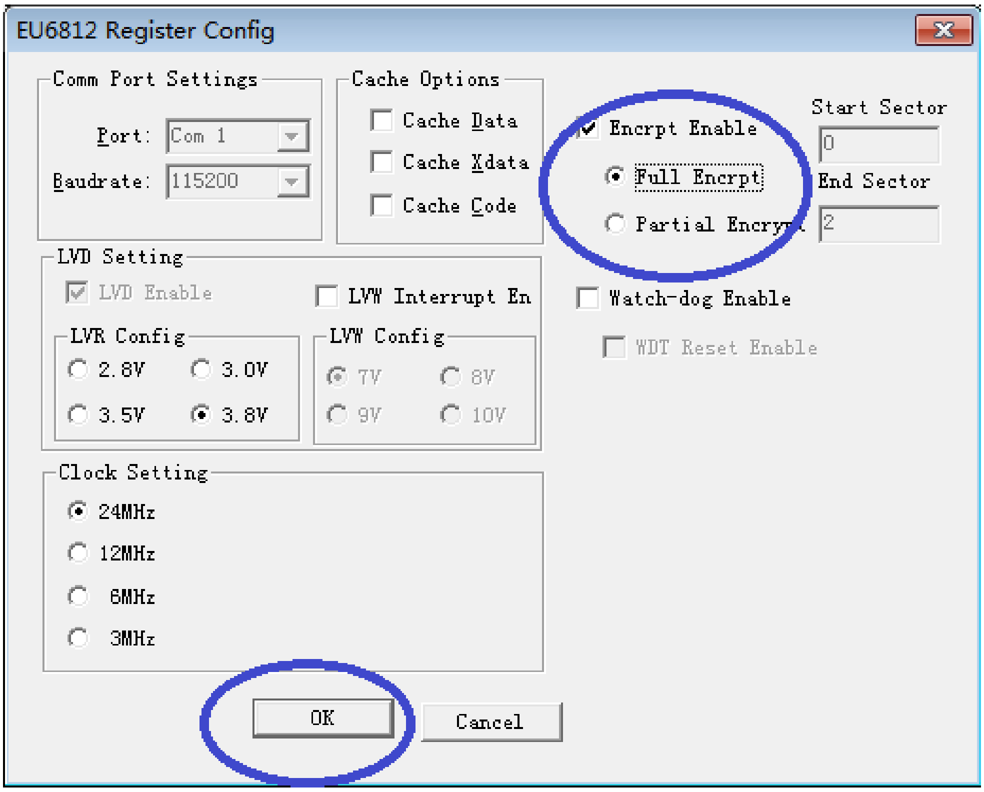
Options for Target 'GPIO'

Device | Target | Output | Listing | User | C51 | A51 | BL51 Locate | BL51 Misc | Debug | Utilities

○ Use Simulator Settings
□ Limit Speed to Real-Time
☑ Load Application at Startup ☑ Run to main()
Initialization File:
Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows & Performance Analyzer
☑ Memory Display
CPU DLL: S8051.DLL Parameter: -pEU6812
Dialog DLL: DCYG.DLL Parameter: -pEU6812

● Use: Fortior C51 FICE Driver Settings
☑ Load Application at Startup □ Run to main()
Initialization File:
Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows
☑ Memory Display
Driver DLL: S8051.DLL Parameter: -pEU6812
Dialog DLL: TCYG.DLL Parameter: -pEU6812

Figure 30-7 Code Protection Configurations



EU6812 Register Config

Comm Port Settings
Port: Com 1
Baudrate: 115200

Cache Options
□ Cache Data
□ Cache Xdata
□ Cache Code

☑ Encrpt Enable
● Full Encrpt
○ Partial Encrypt

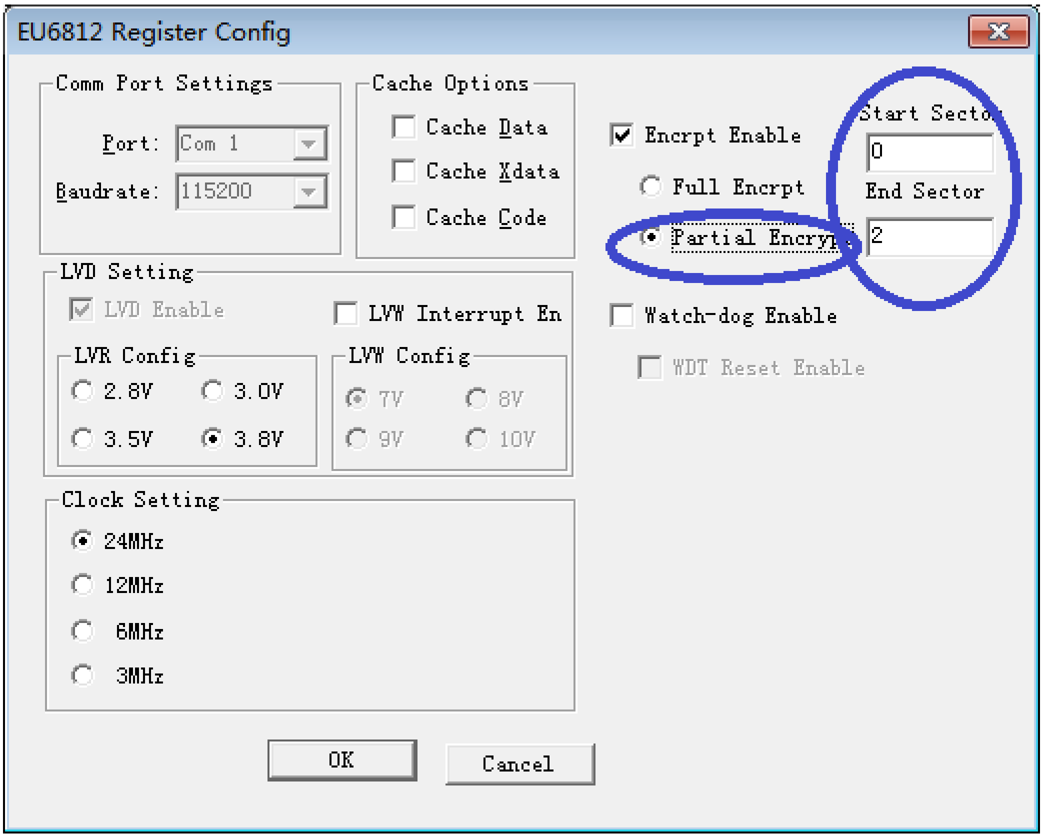
Start Sector: 0
End Sector: 2

LVD Setting
☑ LVD Enable □ LVW Interrupt En
LVR Config: ○ 2.8V ○ 3.0V ○ 3.5V ● 3.8V
LVW Config: ● 7V ○ 8V ○ 9V ○ 10V

□ Watch-dog Enable
□ WDT Reset Enable

Clock Setting
● 24MHz
○ 12MHz
○ 6MHz
○ 3MHz

Figure 30-8 Full Code Protection Mode



EU6812 Register Config

Comm Port Settings

Port: Com 1

Baudrate: 115200

Cache Options

☐ Cache Data

☐ Cache Xdata

☐ Cache Code

☑ Encrpt Enable

○ Full Encrpt

● Partial Encryp

Start Secto: 0

End Sector: 2

☐ Watch-dog Enable

☐ WDT Reset Enable

LVD Setting

☑ LVD Enable

☐ LVW Interrupt En

LVR Config

○ 2.8V ○ 3.0V

○ 3.5V ● 3.8V

LVW Config

● 7V ○ 8V

○ 9V ○ 10V

Clock Setting

● 24MHz

○ 12MHz

○ 6MHz

○ 3MHz

Figure 30-9 Partial Code Protection Mode

The chip supports Flash space encryption to protect your software intellectual property. Operation steps are as follows:

1. Start 8051 IDE, enter Target Options and select Debug tab. As shown in Figure 30-7, click Settings to proceed with the setting;
2. Select the options as shown in Figure 30-8, and click OK. Then compile the project and download it. Get the BIN file and program it to Flash.

The chip support full code protection mode and partial code protection mode. After Full Encrypt is selected as shown in Figure 30-8, all codes in Flash are protected. After Partial Encrypt is selected as shown in Figure 30-9, the codes from sector 0 to END SECTOR are protected. The last sector is always protected in any case.

The size of each sector is 128 bytes.

35 Configuration Registers

35.1 CCFG

35.1.1 CCFG1 (0x401E)

Bit	7	6	5	4	3	2	1	0
Name	LVDENB	LVWIE	WDTEN	RSV		FCK_SEL		RSV
Type	R/W	R/W	R/W	-	-	R/W	R/W	R
Reset	0	0	0	-	-	0	0	0

Bit	Name	Description
[7]	LVDENB	LV Reset Detection Enable 0: Enable. The system resets if VDD5 is lower than the threshold voltage set by CCFG2[LVRSEL]. 1: Disable. Note: The low voltage detector has two main features: low voltage reset and low voltage warning. The chip is reset when VDD5 is lower than the threshold voltage set by CCFG2[LVRSEL], and LV warning interrupt is generated when VCC is lower than the threshold voltage set by CCFG2[LVWSEL]. The above two features work only after LV Reset Detection is enabled (CCFG1[LVDENB]=0) and LVW Detection Interrupt is enabled (CCFG1[LVWIE]=1).
[6]	LVWIE	LVW Detection Interrupt Enable. CCFG2[LVWSEL] decides VCC LV warning voltage threshold. 0: Disable 1: Enable. LV reset detection circuit shall be enabled as well (i.e. CCFG1[LVDENB] = 0).
[5]	WDTEN	WDT Enable 0: Disable 1: Enable
[4:3]	RSV	Reserved
[2:1]	FCK_SEL	System Clock Rate Selection 00: 24MHz 01: 12MHz 10: 6MHz 11: 3MHz
[0]	RSV	Reserved

Note: This register can be directly accessed by software. It is recommended that you set these parameters in the IDE, so that you do not have to configure them again in software.

35.1.2 CCFG2 (0x401D)

This register can be configured by IDE only. The register configurations are compiled and merged with ROM_CODE to produce a BIN file for programming. The register can be read by software only.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LVRSEL		WDTBTEN	WDTRSTEN	RSV		LVWSEL	
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]	LVRSEL	Low Voltage Reset Threshold Voltage. VDD5 is detected for the low voltage reset. 00: Low voltage reset is enabled when VDD5 is lower than 2.8V. 01: Low voltage reset is enabled when VDD5 is lower than 3.0V. 10: Low voltage reset is enabled when VDD5 is lower than 3.5V. 11: Low voltage reset is enabled when VDD5 is lower than 3.8V.
[5]	WDTBTEN	1: BOOT is activated when WDT is reset.
[4]	WDTRSTEN	WDT Reset Enable after Overflow 1: WDT is rest after it is overflowed.
[3:2]	RSV	Reserved
[1:0]	LVWSEL	Low Voltage Warning Threshold Voltage. VCC is detected for the low voltage warning. 00: Low voltage warning is enabled when VCC is lower than 7V. 01: Low voltage warning is enabled when VCC is lower than 8V. 10: Low voltage warning is enabled when VCC is lower than 9V. 11: Low voltage warning is enabled when VCC is lower than 10V.

36 Revision History

Rev.	Description	Date	Prepared By
V0.1	Preliminary datasheet, translated from Chinese version 0.1	2023/07/12	Eric Deng
V1.0	Final datasheet. 1. Added chapters 6 ~ 35; 2. Modified the package name “QHF120120-54L” as “WBQHFBP-45L”; 3. Deleted descriptions on CMP4M and CMP4MS for P2.7 and P3.4; 4. Deleted I^{2}C, LIN and EEPROM from Abbreviations, and deleted descriptions on I^{2}C. 5. Modified the number of interrupt sources as “15” from “16” and the number of ADC channels as “10” from “8” in section 1.1 Features; Added descriptions on SPI in communication interfaces; Modified “3-channel analog comparator” as “2-channel analog comparator”; 6. Modified “PID” as “PI” in sections 1.1 Features, 1.3 Overview, 1.4 Functional Block Diagram; 7. Renamed comparators and operational amplifiers in sections 1.4 Functional Block Diagram and 2 Pin Definitions, and added SPI and deleted INT0 in Figure 1-1 ES9676AH Functional Block Diagram ; 8. Deleted registers I2C_CR, I2C_ID, I2C_DR and I2C_SR with the address 0x4028, and deleted repeated registers with the address 0x0310, 0x0308 and 0x0300 in Table 1-2 XSFR Address Mapping. 9. Updated section 2 Pin Definitions; 10. Added SPI to Table 4-1 Model Selections; Modified the number of ADC channel as “10” from “8” and the number of comparator as “2” from “3”.	2023/11/15	Eric Deng

Copyright Notice

Fortior Technology Co., Ltd.
Customer service:info@fortiortech.com
URL: http://www.fortiortech.com/global/