Datasheet

Three-phase Motor Controller MCU
FU6881Q1

Fortior Technology (Shenzhen) Co., Ltd.

Contents

Explanation of Symbols

- The symbol "[ ]" following a register indicates a bit in the register. For example, ABCD[XY] indicates the XY bit in ABCD register
- The symbol "x" in a register name indicates similar registers. For example, TIMx_CR0 indicates TIM3_CR0 and TIM4_CR0.
- [m:n] indicates a range of bits. For example, [3:0] means the bits from bit3 to bit0.
- Pm.n indicates the n^{th} port of Portm. For example, P0.0 indicates the 0^{th} 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 invalid or uncertainty 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
EEPROM	Electrically Erasable Programmable Read Only Memory
FG	Frequency Generator
FICE	Fortior Interactive Connectivity Establishment
FOC	Field Oriented Control
FOSC	Fast Oscillator
GPIO	General Purpose Input Output
IC	Integrated Circuit
I^2C	Internal Integrated Circuit
IDE	Integrated Development Environment
IRAM	Internal RAM
LDO	Low Dropout Regulator
LIN	Local Interconnect Network
LPF	Low Pass Filter
LSB	Least Significant Bit
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
OCP	Over-current Protection
OVP	Over-voltage Protection
PGA	Programmable Gain Amplifier
PI/PID	Proportional Integral/Proportional Integral Derivative
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 6.5V ~ 20V is connected to VCC pin, and internal LDO supplies VDD5 voltage
- Dual core: 8051 core and ME core
- An instruction cycle mostly takes 1 or 2 system clock cycle(s)
- 32kB Flash ROM with CRC, self-program and code protection
- 256 bytes IRAM and 3.75k bytes XRAM
- Built-in 2k bits EEPROM, supporting data read and write via I²C
- Protection features, including TSD, OCP and OVP
- ME: Core integrating PID module, FOC module, MDU auxiliary computing module and LPF module
- 16 interrupt sources with 4 configurable priority levels
- Number of GPIOs: 16
- Timer:
  - Timer1: Timer supporting square wave drive timing control, automatic commutation, adaptive diode freewheeling masking, cycle-by-cycle current limiting and Hall/BEMF-based position sensing
  - Timer2: Timer supporting PWM output, measurement of duty cycle and period of input PWM wave, measurement of the time of set PWM wave numbers, QEP decoding, tailwind/headwind detection (RSD), rotation direction and speed detection of step motor
  - Timer3/Timer4: Timers supporting PWM output, and measurement of duty cycle and period of input PWM wave. Timer4 supports FG generation and Timer3 supports up to 40MHz input
  - Systick Timer
  - RTC
- Communication interface:
  - 1*SPI
  - 1*I²C
  - 1*UART, supporting single-wire mode
  - 1*LIN, integrating LIN transceiver to implement single-wire mode, fixed baud rate mode, automatic baud rate mode and auto-addressing mode
  - Dual-channel DMA: supporting data transmission via I²C/SPI/UART/LIN
- Analogue peripherals:
  - 12-bit ADC: Internal VREF or external VREF selectable as reference voltage

- Number of ADC channels: 11
- Internal VREF, with configurable 3V, 4V, 4.5V or VDD5
- Zero-crossing point (ZCP) detection circuit
- 2-channel analog comparators
- DAC: Single-channel 6-bit

- Drive Type
  - Built-in MOSFET, high side + low side = 0.25Ω
  - Drive current: 1A
- FOC module supports single/dual/triple-shunt current sampling
- SVPWM and a three-phase inverter are used to drive the two-phase step motor, and dual-shunt current sampling is available
- System clock
  - Built-in 20MHz fast RC oscillator
  - Built-in 32.8kHz slow RC oscillator
- WDT
- LVD
- Temperature sensor
- Two-wire FICE protocol based in-circuit emulation
- AEC-Q100 Qualified (Grade 1)

1.2 Applications

The chip can be used for the drive of motors in automotive grilles, air conditioner outlets, thermal control valves, seat ventilation system, charging port covers, etc.

1.3 Overview

The high-performance motor drive chip incorporates ME core and 8051 core. ME core integrates FOC, MDU, LPF, PID and SVPWM modules that allow for automatic calculation of FOC or square-wave control by the hardware for sensored/sensorless BLDC/PMSM motors, and implements SVPWM control for the two-phase step motor. 8051 core is used for parameter configuration and routine processing. Most of 8051 core instruction cycle takes 1T or 2T clock cycle(s). The dual cores work in parallel to achieve high-performance motor control. The chip integrates high-speed operational amplifiers, comparators, high-speed ADC, CRC, SPI, I^2C, UART, LIN and Timers. The LIN module supports auto-addressing for multiple slave devices at different baud rates. The built-in MOS driver delivers up to 1A drive current, and the EEPROM module provides an additional 2k bits of memory space. Also, the chip offers built-in high-voltage LDO, making it suitable for FOC or square-wave based BLDC/PMSM motors and SVPWM-based two-phase step motors.

1.4 Functional Block Diagram

1.4.1 FU6881Q1

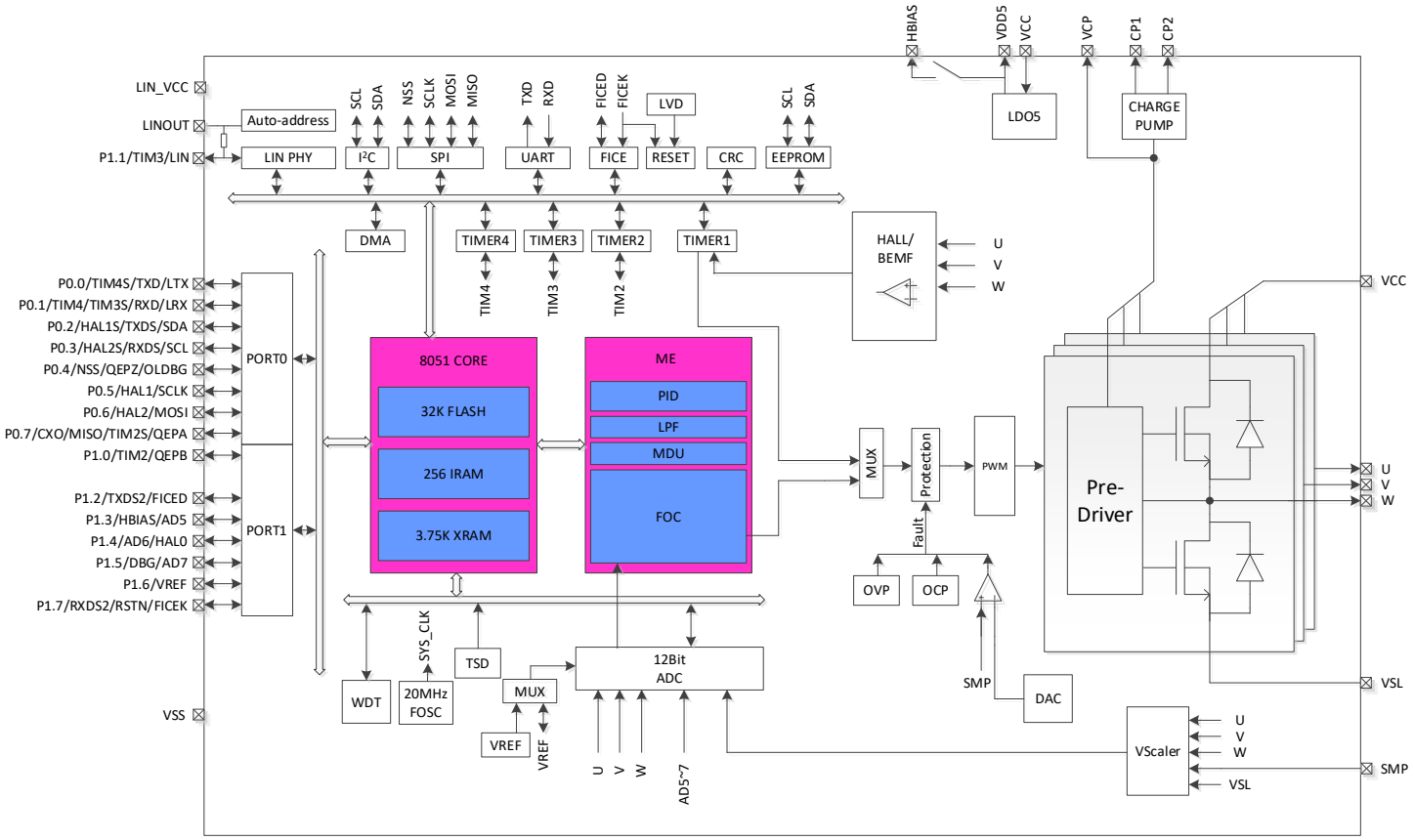


Figure 1-1 Functional Block Diagram of FU6881Q1

1.5 Memory Organization

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

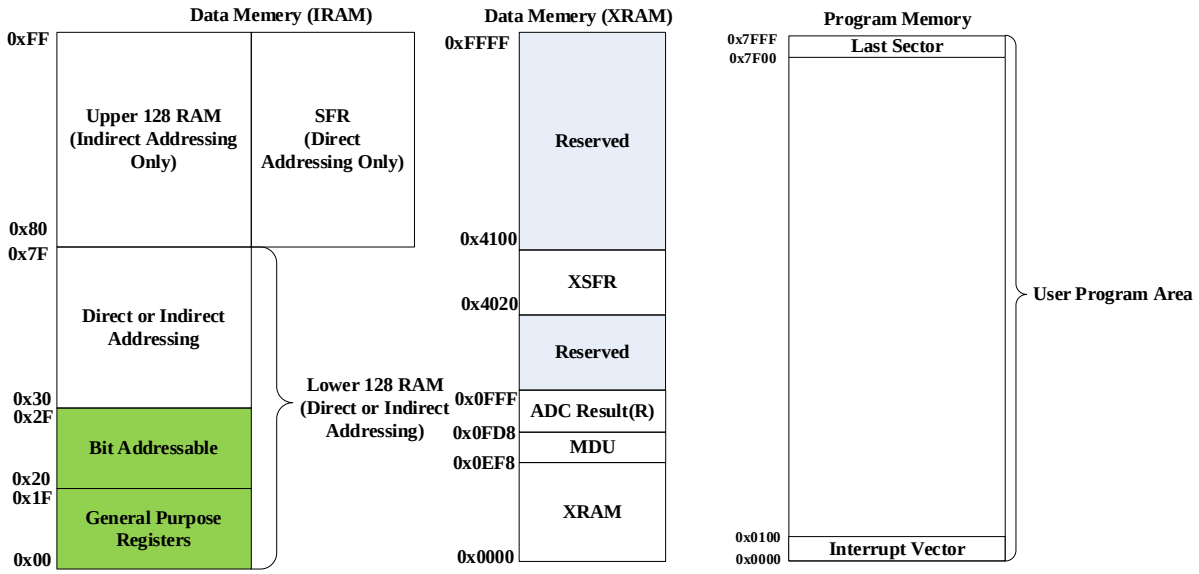


Figure 1-2 Memory Map

1.5.1 Program Memory

The chip implements this program memory as Flash memory with a block from addresses 0x0000 to 0x7FFF to store control programs.

The first sector (0x0000 ~ 0x00FF) is the interrupt vector address area, which is used to store the start address of each interrupt routine. The last sector (0x7F00 ~ 0x7FFF) contains internal control bits of the chip.

1.5.2 Data Memory

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

The External Data Memory is addressed from 0x0000 to 0xFFFF, which can be accessed only with MOVX instructions. It comprises XRAM (0x0000 ~ 0x0EF7), extended control register space (0x4020 ~ 0x40FF), MDU register space (0x0EF8 ~ 0x0FD7) and ADC result memory area (0x0FD8 ~ 0x0FFF).

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

1.5.3 SFR

Table 1-1 SFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xF8	DRV_OUT		INT_SR6		P0_OE	P1_OE		
0xF0	B		INT_SR0	INT_SR1	INT_SR2	INT_SR3	INT_SR4	INT_SR5
0xE8								
0xE0	ACC	CMP_CR4	HALL_CR	OVCP_CR				
0xD8		EVT_FILT	CMP_CR2	LVSR	CMP_CR3			
0xD0	PSW	EXT1_IE	EXT1_IF			CMP_CR0	CMP_CR1	CMP_SR
0xC8		RST_SR						
0xC0		MDU_CR						
0xB8		LIN_CR	LIN_SR	LIN_CSR	LIN_ID	LIN_SIZE	LIN_BAUDL	LIN_BAUDH
0xB0								
0xA8	IE	TIM2_CR1	TIM2__CNTRL	TIM2__CNTRH	TIM2__DRL	TIM2__DRH	TIM2__ARRL	TIM2__ARRH
0xA0		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	CK_CR	TIM4__CNTRL	TIM4__CNTRH	TIM4__DRL	TIM4__DRH	TIM4__ARRL	TIM4__ARRH
0x88	TCON		IP0	IP1	IP2	IP3	CHIPIDL	CHIPIDH
0x80	P0	SP	DPL	DPH	FLA_KEY	FLA_CR		PCON

Notes:

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

1.5.4 XSFR

Table 1-2 XSFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x40E8							DEL_THEH	DEL_THEL
0x40D8	FOC__POWH	FOC__POWL	FOC__IAMAXH	FOC__IAMAXL	FOC__IBMAXH	FOC__IBMAXL	FOC__ICMAXH	FOC__ICMAXL
	FOC_EOMEKLPF							
	ST_VABK							
0x40D0	FOC__EALPH	FOC__EALPL	FOC__EBETH	FOC__EBETL	FOC__EOMEH	FOC__EOMEL		
0x40C8	FOC__IBH	FOC__IBL	FOC__IAH	FOC__IAL	FOC__THETAH	FOC__THETAL	FOC__ETHETAH	FOC__ETHETAL
0x40C0	FOC__IBETH	FOC__IBETL	FOC__VBETH	FOC__VBETL	FOC__VALPH	FOC__VALPL	FOC__ICH	FOC__ICL
			FOC_UDCPSH	FOC_UDCPSL				
0x40B8	FOC__UDH	FOC__UDL	FOC__UQH	FOC__UQL	FOC__IDH	FOC__IDL	FOC__IQH	FOC__IQL
0x40B0	FOC__OMEESTH	FOC__OMEESTL						
	ST__OMEESTH	ST__OMEESTL						
0x40A8			FOC_EOMELPFH	FOC_EOMELPFL			FOC_THECOMPH	FOC_THECOMPL
							ST_BPCOMP	
				ST__ETHETAH		CMP_SAMR	FOC__EMFH	FOC__EMFL
0x40A0	FOC_CR1	FOC_CR2	FOC_TSMIN	FOC_TGLI	FOC_TBLO	FOC_TRGDLY	FOC_CSOH	FOC_CSOL
			ST_DZCOMP					
0x4098	FOC__UDCFLTH	FOC__UDCFLTL				FOC_CR4	FOC_CR3	FOC_CR0
	TIM1__ITRIPH	TIM1__ITRIPL						
0x4090	FOC_IDREFH	FOC_IDREFL	FOC_IQREFH	FOC_IQREFL	FOC_QKPH	FOC_QKPL	FOC_QKIH	FOC_QKIL
	TIM1__URESH	TIM1__URESL	TIM1_KRMAX	TIM1_KFMIN	TIM1_KFH	TIM1_KFL	TIM1_KRH	TIM1_KRL
0x4088	FOC_EK3H	FOC_EK3L	FOC_QMAX	FOC_QMIN	FOC_EK1H	FOC_EK1L	FOC_EK2H	FOC_EK2L
	TIM1__RARRH	TIM1__RARRL	TIM1__RCNTRH	TIM1__RCNTRL	TIM1__UCOPH	TIM1__UCOPL	TIM1__UFLPH	TIM1__UFLPL
0x4080	FOC_FBASEH	FOC_FBASEL	FOC_EFREQACCH	FOC_EFREQACCL	FOC_EFREQMINH	FOC_EFRQMINL	FOC_EFREQHOLDH	FOC_EFREQHOLDL
	TIM1_DBR7H	TIM1_DBR7L	TIM1__BCNTRH	TIM1__BCNTRL	TIM1__BCCRH	TIM1__BCCRL	TIM1__BARRH	TIM1__BARRL
0x4078	FOC_DMAX	FOC_DMIN	FOC_EKLPFMINH	FOC_EKLPFMINL	FOC_DKIH	FOC_DKIL	FOC_OMEKLPFH	FOC_OMEKLPFL

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
	TIM1_DBR3H	TIM1_DBR3L	TIM1_DBR4H	TIM1_DBR4L	TIM1_DBR5H	TIM1_DBR5L	TIM1_DBR6H	TIM1_DBR6L
0x4070	TIM1_BCORH	TIM1_BCORL	TIM1_CR5		FOC_EKPH	FOC_EKPL	FOC_EKIH	FOC_EKIL
	FOC_DKPH	FOC_DKPL			TIM1_DBR1H	TIM1_DBR1L	TIM1_DBR2H	TIM1_DBR2L
0x4068	TIM1_CR0	TIM1_CR1	TIM1_CR2	TIM1_CR3	TIM1_CR4	TIM1_IER		
0x4060	DRV_DTR	DRV_CR1	DRV_CR0		SYST_CR		DRV__CNTRH	DRV__CNTRL
0x4058	DRV_DRH	DRV_DRL	DRV_COMRH	DRV_COMRL	DRV_CMRH	DRV_CMRL	DRV_ARRH	DRV_ARRL
0x4050	P1_AN		PX_PL	P0_PU	P1_PU			
0x4048	TSEN_DR	PH_SEL2	LCP_DR		PH_SEL	PH_SEL1	AMP_CR0	VREF_VHALF_CR
0x4040	DMA1_BAH	DMA1_BAL			CAL_CR0	CAL_CR1		
0x4038	ADC_SCYC2	ADC_CR	DMA0_CR0	DMA1_CR0	DMA0_LEN	DMA1_LEN	DMA0_BAH	DMA0_BAL
0x4030	SPI_CR0	SPI_CR1	SPI_CLK	SPI_DR	AMP_CR1	ADC_SCYC1	ADC_MASKH	ADC_MASKL
0x4028	I2C_CR	I2C_ID	I2C_DR	I2C_SR	RTC_TMH	RTC_TML	RTC_STA	TSD_CR
0x4020		CRC_DIN	CRC_CR	CRC_DR	CRC_BEG	CRC_CNT	WDT_CR	WDT_ARR
0x0FE8	AD8_DRH	AD8_DRL	AD9_DRH	AD9_DRL	AD10_DRH	AD10_DRL		
0x0FE0	AD4_DRH	AD4_DRL	AD5_DRH	AD5_DRL	AD6_DRH	AD6_DRL	AD7_DRH	AD7_DRL
0x0FD8	AD0_DRH	AD0_DRL	AD1_DRH	AD1_DRL	AD2_DRH	AD2_DRL	AD3_DRH	AD3_DRL
0x0FD0	LPF0_K		LPF0_X		LPF0_YH		LPF0_YL	
0x0FC8	LPF1_K		LPF1_X		LPF1_YH		LPF1_YL	
0x0FC0	PI0_UKH		PI0_UKL		PI0_UKMAX		PI0_UKMIN	
0x0FB8	PI0_KP		PI0_EK1		PI0_EK		PI0_KI	
0x0FB0	PI1_UKH		PI1_UKL		PI1_UKMAX		PI1_UKMIN	
0x0FA8	PI1_KP		PI1_EK1		PI1_EK		PI1_KI	
0x0FA0	MUL0_MA		MUL0_MB		MUL0_MCH		MUL0_MCL	
0x0F98	MUL1_MA		MUL1_MB		MUL1_MCH		MUL1_MCL	
0x0F90	DIV0_DB		DIV0_DQH		DIV0_DQL		DIV0_DR	
0x0F88	DIV1_DQL		DIV1_DR		DIV0_DAH		DIV0_DAL	
0x0F80	DIV1_DAH		DIV1_DAL		DIV1_DB		DIV1_DQH	
0x0F78	LPF2_K		LPF2_X		LPF2_YH		LPF2_YL	
0x0F70	LPF3_K		LPF3_X		LPF3_YH		LPF3_YL	

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x0F68	PI2_UKMAX		PI2_UKMIN		PI2_KD		PI2_EK2	
0x0F60	PI2_EK		PI2_KI		PI2_UKH		PI2_UKL	
0x0F58	PI3_KD		PI3_EK2		PI2_KP		PI2_EK1	
0x0F50	PI3_UKH		PI3_UKL		PI3_UKMAX		PI3_UKMIN	
0x0F48	PI3_KP		PI3_EK1		PI3_EK		PI3_KI	
0x0F40	MUL2_MA		MUL2_MB		MUL2_MCH		MUL2_MCL	
0x0F38	MUL3_MA		MUL3_MB		MUL3_MCH		MUL3_MCL	
0x0F30	DIV2_DB		DIV2_DQH		DIV2_DQL		DIV2_DR	
0x0F28	DIV3_DQL		DIV3_DR		DIV2_DAH		DIV2_DAL	
0x0F20	DIV3_DAH		DIV3_DAL		DIV3_DB		DIV3_DQH	
0x0F18	SCAT0_SIN		SCAT0_THE		SCAT0_RES1		SCAT0_RES2	
0x0F10	SCAT1_THE		SCAT1_RES1		SCAT1_RES2		SCAT0_COS	
0x0F08	SCAT2_RES1		SCAT2_RES2		SCAT1_COS		SCAT1_SIN	
0x0F00	SCAT3_RES2		SCAT2_COS		SCAT2_SIN		SCAT2_THE	
0x0EF8	SCAT3_COS		SCAT3_SIN		SCAT3_THE		SCAT3_RES1	

Notes:

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

2 Pin Definitions

The IO types are defined as follows:

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

2.1 FU6881Q1 QFN40 Pins

Table 2-1 FU6881Q1 QFN40 Pins

Pin	FU6881Q1 QFN40	IO Type	Description
P0.0/ TIM4S/ TXD/ LTX	1	DO/ DB/ DO/ DO	GPIO, configurable as INT0 input Timer4 input/output after function switching UART TXD output LIN TXD output
P0.1/ TIM4/ TIM3S/ RXD/ LRX	2	DB/ DB/ DB/ DB/ DI	GPIO, configurable as INT0/INT1 input Timer4 input/output Timer3 input/output after function switching UART RXD input in two-wire mode or TXD output/RXD input in single-wire mode LIN RXD input
P0.2/ HAL1S/ TXDS/ SDA	3	DB/ DI/ DO/ DB	GPIO, configurable as INT0 input Hall-IC1 logic level input after function switching UART TXD output after function switching I^2C SDA, configurable as open-drain output
P0.3/ HAL2S/ RXDS/ SCL	4	DB/ DI/ DB/ DO	GPIO, configurable as INT0/INT1 input Hall-IC2 logic level input after function switching UART RXD input in two-wire mode or TXD output/RXD input in single-wire mode after function switching I^2C SCL, configurable as open-drain output
P0.4/ NSS/ QEPZ	5	DB/ DB/ DI	GPIO, configurable as INT0 input SPI NSS QEP ZCP detection input
P0.5/ HAL1/ SCLK	6	DB/ DI/ DB	GPIO, configurable as INT1 input Hall-IC1 logic level input SPI SCLK
P0.6/ HAL2/ MOSI	7	DB/ DI/ DB	GPIO, configurable as INT1 input Hall-IC2 logic level input SPI MOSI, master output or slave input
P0.7/ MISO/ TIM2S/ QEPA	8	DB/ DB/ DB/ DI	GPIO, configurable as INT1 input SPI MISO, master input or slave output Timer2 input/output after function switching QEP encode A input
P1.2/ TXDS2/ FICED	9	DB/ DB/ DB	GPIO, configurable as INT1 input UART TXD output after function switching FICE SDA
P1.7/ RXDS2/	10	DI/ DB/	GPIO input only, configurable with pull-up or pull-down resistors UART RXD input in two-wire mode or TXD output/RXD input in single-wire mode after function switching;

Pin	FU6881Q1 QFN40	IO Type	Description
RSTN/ FICEK		DI/ DI	Input of external reset; Built-in pull-up resistor FICE SCL
P1.0/ TIM2/ QEPB	11	DB/ DB/ DI	GPIO, configurable as INT0/INT1 input, open drain Timer2 input/output, open drain QEP encode B input, open drain
P1.1/ TIM3/ LIN	12	DB/ DI/ DB	GPIO, configurable as INT0/INT1 input, open drain Timer3 input before function switching, open drain LIN single-wire RXD/TXD, open drain
LINOUT	13	AB	LIN auto-addressing
LIN_VCC	14	P	LIN input power
CP1	15	AO	Charge pump pin, with a 0.1μF capacitor connected between CP2 and CP1
CP2	16	AO	Charge pump pin, with a 0.1μF capacitor connected between CP2 and CP1
VCP	17	P	Charge pump output, with a 1μF ~ 4.7μF capacitor connected to VCC pin
DVCC	18	P	Driver input power
DVCC	19	P	Driver input power
W	20	DO	W-phase output
W	21	DO	W-phase output
NC	22	-	Not connected
VSL	23	DO	Low-side ground output
VSL	24	DO	Low-side ground output
V	25	DO	V-phase output
V	26	DO	V-phase output
VSL	27	DO	Low-side ground output
VSL	28	DO	Low-side ground output
NC	29	-	Not connected
U	30	DO	U-phase output
U	31	DO	U-phase output
DVCC	32	P	Driver input power
VCC	33	P	Power input; 6.5V ~ 20V; With an external filter capacitor of 1μF or above
GND	34	P	Ground
VDD5	35	P	Internal LDO outputs 5V power supply, with an external capacitor of 1μF or above
P1.6/ VREF	36	DB/ AO	GPIO ADC external VREF input or internal VREF output, with a 1μF ~ 4.7μF external capacitor
SMP	37	AI	Bus current sampling input
P1.3/ HBIAS/ AD5	38	DB/ DO/ AI	GPIO Hall bias power supply, internally connected to VDD5 via switch to achieve large current output Input of ADC channel 5
P1.4/ AD6/ HAL0	39	DB/ AI/ DI	GPIO Input of ADC channel 6 Hall-IC0 logic level input
P1.5/ DBG/ AD7	40	DB/ DO/ AI	GPIO, configurable as INT0 input Debug port Input of ADC channel 7

2.2 FU6881Q1 QFN40 Pinout Diagram

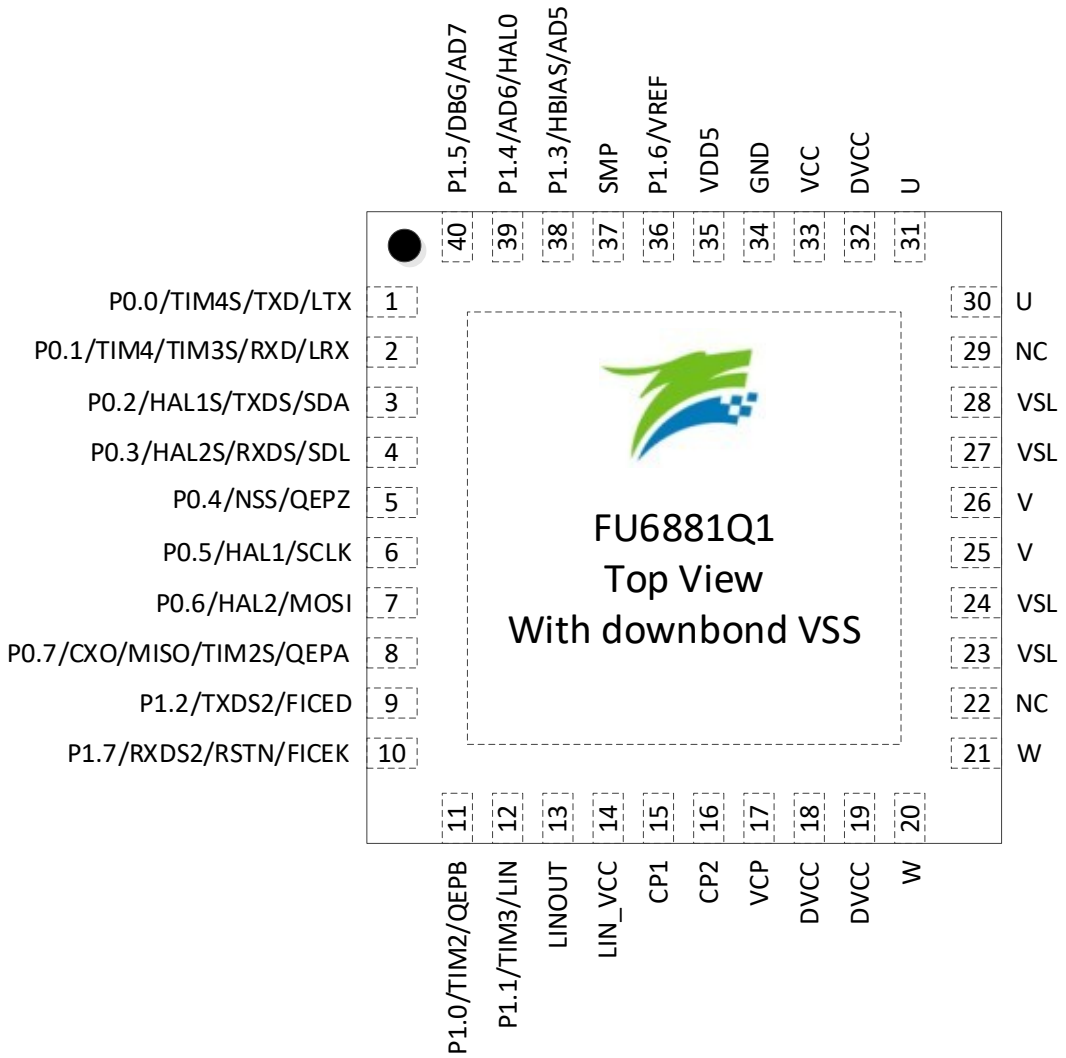
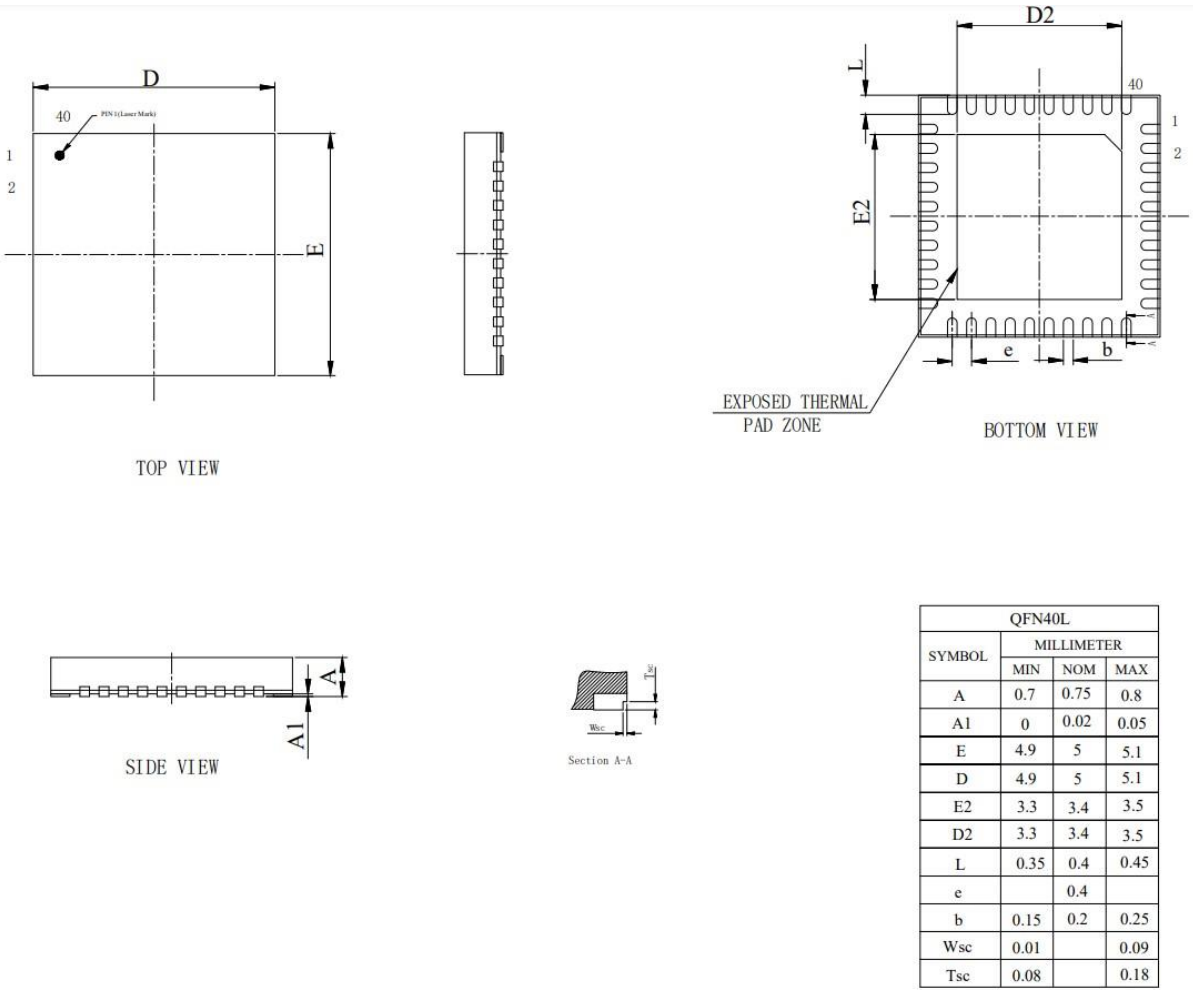


Figure 2-1 FU6881Q1 QFN40 Pinout Diagram

3 Package Information

3.1 FU6881Q1 QFN40_5X5



QFN40L			
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.7	0.75	0.8
A1	0	0.02	0.05
E	4.9	5	5.1
D	4.9	5	5.1
E2	3.3	3.4	3.5
D2	3.3	3.4	3.5
L	0.35	0.4	0.45
e		0.4	
b	0.15	0.2	0.25
Wsc	0.01		0.09
Tsc	0.08		0.18

Figure 3-1 FU6881Q1 QFN40_5X5 Package Drawings and Dimensions

4 Ordering Information

Table 4-1 Model Selections

Model	Supply Voltage (V)	Rdson (H+L) (Ω)	Clock Frequency (MHz)	Flash (kByte)	XRAM (kByte)	Clock Circuits		Drive Type			I²C/UART/SPI/LIN	DMA	GPIO	Timer	Analogue Peripherals							Lead-free	Package
						Internal Fast Clock	Internal Slow Clock	Square-wave	FOC	Step					ADC			DAC		VREF	Comparator		
															Number	Channel	Bits	Number	Bits				
FU6881Q1	6.5~20	0.25	20	32	3.75	√	√	√	√	√	√	√	16	6	1	11	12	1	6	√	2	√	QFN40 (5x5mm)

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

Table 5-1 Absolute Maximum Ratings

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Operating Junction Temperature T_J		-40	-	150	℃
Storage Temperature T_{STG}		-55	-	150	℃
VCC to VSS Spike Voltage	t < 60s	-0.3	-	36	V
VCC to VSS Voltage	Phase-U/V/W output is turned off during chip operation	-0.3	-	28	V
VCC to VSS Voltage	Phase-U/V/W output is turned on	-0.3	-	22	V
U/V/W/CP1 to VSS Voltage		-0.3	-	VCC + 0.3	V
VCP/CP2 to VCC Voltage		-0.3	-	VCC + 6.0	V
VSL to VSS Voltage		-0.3	-	0.5	V
VDD5 to VSS Voltage		-0.3	5	6.5	V
P10		-0.3	-	VCC + 0.3	V
P11/LINOUT	t < 500ms	-22	-	40	V
RSTN/SMP/GPIO to VSS Voltage	Except P10, P11 and LINOUT	-0.3	-	VDD5 + 0.3	V

Note: Stress values greater than the "Absolute Maximum Ratings" listed in Table 5-1 may cause irremediable damages to the device. These are stress ratings only, and it is NOT recommended to use your device in conditions that go beyond these stress ratings. Exposure to "Absolute Maximum Ratings" for extended periods may affect device reliability.

5.2 Global Electrical Characteristics

Table 5-2 Global Electrical Characteristics

(T_A = -40℃ ~ 125℃ and VCC = 6.5V ~ 20V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VCC Operating Voltage[1][2]	High-voltage Single-power Supply Mode	6.5	-	20	V
I_{VCC} Operating Current[3]	Average Bus Current	-	-	1	A
I_{VCC} Quiescent Current[3]	Turn off All MOS	-	20	-	mA
I_{VCC} Standby Current [3]		-	10	-	mA
I_{VCC} Sleep-mode Current [3]	T_A = 25℃	-	30	50	μA
Rdson (High-side MOS + Low-side MOS)		-	0.22	0.4	Ω

Notes:

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

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

[3] Characteristics may vary with different configurations.

5.3 GPIO Electrical Characteristics

Table 5-3 GPIO Electrical Characteristics

(T_A = 25℃ and VCC = 6.5V ~ 20V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Turn-on Rise Time	50pF load, from 10% to 90%, T_A = 25℃	-	15	-	ns
Turn-off Fall Time	50pF load, from 90% to 10%, T_A = 25℃	-	13	-	ns
V_{OH} High-level Output Voltage	I_{OH} = 4mA T_A = -40℃ ~ 125℃	VDD5 - 0.7	-	-	V
V_{OL} Low-level Output Voltage	I_{OL} = 4mA T_A = -40℃ ~ 125℃	-	-	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
P10 High-level Input Voltage	High-voltage Mode	0.7*VCC	-	-	V
P10 Low-level Input Voltage	High-voltage Mode	-	-	0.2*VCC	V
P11/LINOUT High-level Input Voltage	LIN Mode	0.6*VCC	-	-	V
P11/LINOUT Low-level Input Voltage	LIN Mode	-	-	0.4*VCC	V
Pull-up Resistor[2]		-	33	-	kΩ
Pull-up Resistor[3]		-	5.6	-	kΩ
Pull-down Resistor[4]		-	33	-	kΩ
Pull-down Resistor[5]		-	50	-	kΩ

Notes:

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

[2] P0[4], P0[7], P1[3:2] and P1[7:5]

[3] P0[3:0], P0[6:5] and P1[4]

[4] P1[7]

[5] P0[1]

5.4 ADC Electrical Characteristics

Table 5-4 ADC Electrical Characteristics

(T_A = 25℃ and VCC = 6.5V ~ 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

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
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	ADCLK2[2]

Notes:

[1] ADCLK = 20MHz

[2] ADCLK2 = 10MHz

5.5 VREF Electrical Characteristics

Table 5-5 VREF Electrical Characteristics

(T_A = –40°C ~ 125°C and VCC = 6.5V ~ 20V unless otherwise specified)

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

5.6 Operational Amplifier Electrical Characteristics

Table 5-6 Operational Amplifier Electrical Characteristics

(T_A = 25°C and VCC = 6.5V ~ 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
U_{GBW} Unity-gain Bandwidth	C_L = 40pF	6	10	-	MHz
Slew Rate (SR)	C_L = 40pF	10	15	-	V/μs
Operational Amplifier Gain[1]	AMP_CR1[AMP_GAIN] = 00	7.5	8	8.5	-
	AMP_CR1[AMP_GAIN] = 01	11	12	13	-
	AMP_CR1[AMP_GAIN] = 10	15	16	17	-
	AMP_CR1[AMP_GAIN] = 11	18.5	20	21.5	-

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.

5.7 OSC Electrical Characteristics

Table 5-7 OSC Electrical Characteristics

(T_A = –40°C ~ 125°C and VCC = 6.5V ~ 20V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
System Clock Rate		19.5	20	20.5	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 20MHz and T = 1/SYSCLK.

5.8 Reset Electrical Characteristics

Table 5-8 Reset Electrical Characteristics

(T_A = 25°C and VCC = 6.5V ~ 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

5.9 LDO Electrical Characteristics

Table 5-9 LDO Electrical Characteristics

(T_A = 25°C and VCC = 6.5V ~ 20V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD5 Voltage	T_A = -40°C ~ 125°C	4.7	5	5.3	V

5.10 LIN Electrical Characteristics

Table 5-10 LIN Electrical Characteristics

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{BAT} LIN Controller Operating Voltage		8	-	18	V
$V_{BUS\_MAX\_RATING}$ Max. Rated Bus Voltage		-27	-	40	V
$V_{SUP\_NON\_OP}$ Supply Voltage Within Which LIN P1.1 Is Not Destroyed	No Guarantee of Correct Operations	-0.3	-	40	V
I_{BUS_LIM} Current Limitation for Driver in Dominant State		40	-	200	mA
$I_{BUS\_PAS\_dom}$ Input Leakage Current	V_{BUS} = 0V, V_{BAT} = 12V	-1	-	-	mA
$I_{BUS\_PAS\_rec}$ Current When Driver Is OFF	8V < V_{BAT} < 18V; 8V < V_{BUS} < 18V; $V_{BUS} \geq V_{BAT}$	-	-	20	μA
$I_{BUS\_NO\_GND}$ Current When Control Unit Is Disconnected from Ground	V_{SS} = V_{CC} = 12V; 0V < V_{BUS} < 18V	-1	-	1	mA
$I_{BUS\_NO\_BAT}$ Current When V_{BAT} Is Disconnected	V_{CC} = V_{SS} = 0V; 0 < V_{BUS} < 18V	-	-	100	μA

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{BUS_dom} LIN Bus Voltage for Receiver in Dominant State		-	-	0.4	VCC
V_{BUS_rec} LIN Bus Voltage for Receiver in Recessive State		0.6	-	-	VCC
V_{BUS_CNT} LIN Bus Center-point Voltage	$V_{BUS\_CNT} = (V_{th\_dom} + V_{th\_rec}) / 2$	0.475	0.5	0.525	VCC
V_{HYS} Receiver Hysteresis Voltage	$V_{HYS} = V_{th\_rec} - V_{th\_dom}$	-	-	0.175	VCC
D1 Duty Cycle 1	Transfer Rate = 20.0kbps; V_{SUP} = 7.0V ~ 18V; t_{BIT} = 50μs; $TH_{Rec(max)} = 0.744*V_{SUP}$; $TH_{Dom(max)} = 0.581*V_{SUP}$; $D1 = t_{Bus\_rec(min)}/(2*t_{BIT})$	0.396	-	-	%
D2 Duty Cycle 2	Transfer Rate = 20.0kbps; V_{SUP} = 7.6V ~ 18V; t_{BIT} = 50μs; $TH_{Rec(min)} = 0.422*V_{SUP}$; $TH_{Dom(min)} = 0.284*V_{SUP}$; $D2 = t_{Bus\_rec(max)}/(2*t_{BIT})$	-	0.581	-	%
D3 Duty Cycle 3	Transfer Rate = 10.4kbps; V_{SUP} = 7.0V ~ 18V; t_{BIT} = 96μs; $TH_{Rec(max)} = 0.778*V_{SUP}$; $TH_{Dom(max)} = 0.616*V_{SUP}$; $D3 = t_{Bus\_rec(min)}/(2*t_{BIT})$	0.417	-	-	%
D4 Duty Cycle 4	Transfer Rate = 10.4kbps; V_{SUP} = 7.6V ~ 18V; t_{BIT} = 96μs; $TH_{Rec(min)} = 0.389*V_{SUP}$; $TH_{Dom(min)} = 0.251*V_{SUP}$; $D4 = t_{Bus\_rec(max)}/(2*t_{BIT})$	-	0.59	-	%
t_{rx_pd} Propagation Delay of Receiver		-	-	6	μs
t_{rx_sym} Symmetry Between Rising Edge and Falling Edge for Receiver Propagation Delay		-2	-	2	μs
C_{SLAVE} Capacitor of Slave Node		-	250	-	pF
R_{SLAVE} Pull-up Resistor of Slave Node		20	30	60	kΩ

5.11 Package Thermal Characteristics

Table 5-11 QFN40 Thermal Characteristics

Parameter	Test Conditions	Min.	Typ.
Junction-to-ambient Thermal Resistance θ_{JA}[1]	JEDEC standard, 2S2P PCB	40	℃/W
	JEDEC standard, 1S0P PCB	60	℃/W
Junction-to-case Thermal Resistance θ_{JC}[1]	JEDEC standard, 2S2P PCB	15	℃/W

Note:

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

6 Reset Control

6.1 Reset Source (RST_SR)

The chip includes a reset circuitry with 7 reset sources:

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

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”. In order to clear a reset flag, you can set RST_SR[RSTCLR] flag to “1” so that RST_SR[7:3]&RST_SR[0] are cleared. After reset, MCU starts the program from address 0x0000.

6.2 Reset Enable

See the corresponding control registers.

6.3 Power-on Reset and External Reset

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

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

6.4 Low Voltage Detection Reset

The chip’s internal circuitry monitors VCC voltage. When VCC voltage drops to a level below the reset threshold voltage, the internal monitor circuitry sends a reset signal to reset the chip.

Configuring corresponding register enables low-voltage monitor circuitry and sets the low voltage threshold .

6.5 Watchdog Timer Reset

After the watchdog timer (WDT) is enabled, a system reset occurs when WDT overflows or “times out”. If system error occurs or system reset fails, the timer generates an output pulse to reset the chip.

6.6 Flash Error Detector Reset

The Flash memory can be programmed by the software using MOVX instruction for read/write/erase operations. A Flash error detector reset (RSTFED) occurs if a Flash erase is attempted targeting the last sector (0x7F00 ~ 0x7FFF) or a Flash write is attempted targeting the last byte (0x7FFF). RSTFED is always enabled and cannot be disabled.

6.7 Debug Reset

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

6.8 Soft Reset

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

6.9 Reset Registers

6.9.1 RST_SR (0xC9)

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

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

7 Interrupt

7.1 Interrupt Introduction

The chip includes an interrupt system with a total of 16 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 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

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", the MCU immediately enters the interrupt subroutine. Therefore, make sure to clear corresponding interrupt flag bit before enabling the interrupt.

7.3 External Interrupts

The external interrupt has two 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 P1.5, P1.1 ~ P1.0 and P0.4 ~ P0.0 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 P0.7 ~ P0.5, P0.1, P0.3 and P1.2 ~ P1.0 can be used to trigger an INT1. EXT1_IF is the interrupt flag bit, and EXT1_IE is the interrupt enable bit. Each trigger source has a corresponding interrupt flag bit and an interrupt enable bit. INT1 can select multiple trigger sources that are recognized by EXT1_IF in the interrupt subroutine. These eight interrupt sources share one interrupt entry and one interrupt enable bit IE[EX1]. To enable INT1, first set IE[EX1] to "1" and then configure the corresponding enable bit. The

interrupt edge is configured by TCON[IT1] bit, and the priority level by IP0[PX1] bit. See 7.5.7 ETX1_IE (0xD1) ~ 7.5.8 EXT1_IF (0xD2) 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 Software?	Enable Bit	Priority Control
Reset	Highest	0x0000	None	N	Always Enabled	Highest
LVW Interrupt TSD	0	0x0003	INT_SR3[0] INT_SR3[6]	Y	CCFG1[6] IE[1]	IP0[1:0]
INT0	1	0x000B	INT_SR4[1]	Y	IE[0]	IP0[3:2]
INT1	2	0x0013	EXT1_IF	Y	IE[2]	IP0[5:4]
DRV CM Interrupt	3	0x001B	INT_SR2[4] INT_SR2[3]	Y	DRV_CR1[3] DRV_CR1[2:0]	IP0[7:6]
Timer2 Interrupt	4	0x0023	INT_SR2[2:0]	Y	TIM2_CR1[4:3] TIM2_CR0[3]	IP1[1:0]
Timer1 Interrupt	5	0x002B	INT_SR0[6:0]	Y	TIM1_CR5[7] TIM1_IER[5:0]	IP1[3:2]
ADC Interrupt	6	0x0033	INT_SR4[0]	Y	ADC_CR[1]	IP1[5:4]
CMP0/1/2 Interrupt Hall Interrupt	7	0x003B	INT_SR3[3:1] INT_SR3[5]	Y	CMP_CR0[5:0] HALL_CR[6]	IP1[7:6]
RTC	8	0x0043	INT_SR2[6]	Y	IE[6]	IP2[1:0]
Timer3 Interrupt	9	0x004B	INT_SR1[2:0]	Y	TIM3_CR1[4:3] TIM3_CR0[3]	IP2[3:2]
Systick Interrupt	10	0x0053	INT_SR2[5]	Y	SYST_CR[1:0]	IP2[5:4]
Timer4 Interrupt	11	0x005B	INT_SR1[5:3]	Y	TIM4_CR1[4:3] TIM4_CR0[3]	IP2[7:6]
CMP3 Interrupt OVP Interrupt OCP Interrupt	12	0x0063	INT_SR3[4] INT_SR4[7] INT_SR4[6]	Y	CMP_CR0[7:6] OVCP_CR[5] OVCP_CR[1]	IP3[1:0]
I^2C Interrupt UART Interrupt	13	0x006B	INT_SR5[0] UT_CR[1:0]	Y	I2C_CR[0] IE[4]	IP3[3:2]
SPI Interrupt LIN Interrupt	14	0x0073	INT_SR5[4:1] INT_SR6[4:0]	Y	IE[3] LIN_CR[3]	IP3[5:4]
DMA Interrupt	15	0x007B	INT_SR4[3:2]	Y	DMA0_CR0[2] DMA1_CR0[0]	IP3[7:6]

Notes:

- UT_CR[RI] and UT_CR[TI] flags can be cleared to “0” or set to “1” by software. When these flags are set to “1”, an interrupt request is generated. Other interrupt flags can only be cleared to “0” by software, and setting them to “1” by software has no meaning.
- For a register containing multiple interrupt flags, you can write a “1” to the active interrupt flags in order to prevent clearing a interrupt flag to “0”. Take INT_SR2 as an example. When INT_SR2 [SYSTIF] is cleared to “0” by software, you can use the statement INT_SR2 = (INT_SR2&0xDF) | 0x08 to prevent the software from clearing INT_SR2 [DCIF] to “0”.

7.5 Interrupt Registers

7.5.1 IE (0xA8)

Bit	7	6	5	4	3	2	1	0
Name	EA	RTCIE	MCDIE	ES0	SPIIE	EX1	TSDIE	EX0
Type	R/W	R/W	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]	EA	All Interrupts Enable 0: Disable 1: Enable
[6]	RTCIE	RTC Interrupt Enable 0: Disable 1: Enable
[5]	MCDIE	Missing Clock Detector Interrupt Enable 0: Disable 1: Enable
[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 (0x8A)

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

Bit	Name	Description
[7:6]	PDRV	DRV Compare Match Interrupt Priority Setting
[5:4]	PX1	External Interrupt 1 (INT1) Priority Setting
[3:2]	PX0	External Interrupt 0 (INT0) Priority Setting
[1:0]	PLVW_TSD	LVW/TSD Interrupt Priority Setting

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

7.5.3 IP1 (0x8B)

Bit	7	6	5	4	3	2	1	0
Name	PCMP_HALL		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_HALL	CMP0/1/2 and Hall 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 (0x8C)

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

Bit	Name	Description
[7:6]	PTIM4	Timer4 Interrupt Priority Setting
[5:4]	PSYSTICK	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 (0x8D)

Bit	7	6	5	4	3	2	1	0
Name	PDMA		PSPI_LIN		PI2C_UART		PCMP3_OVP_OCP	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	PDMA	DMA Interrupt Priority Setting
[5:4]	PSPI_LIN	SPI/LIN Interrupt Priority Setting
[3:2]	PI2C_UART	I^2C/UART Interrupt Priority Setting
[1:0]	PCMP3_OVP_OCP	CMP3/OVP/OCP 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			IT1		RSV	IT0	
Type	-	-	-	R/W	-	R/W0	R/W	R/W
Reset	-	-	-	0	-	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved

[4:3]	IT1	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]	RSV	Reserved
[1:0]	IT0	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 ETX1_IE (0xD1)

Bit	7	6	5	4	3	2	1	0
Name	P07_IE	P06_IE	P05_IE	P01_IE	P03_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]	P07_IE	P0.7 INT1 Enable 0: Disable 1: Enable
[6]	P06_IE	P0.6 INT1 Enable 0: Disable 1: Enable
[5]	P05_IE	P0.5 INT1 Enable 0: Disable 1: Enable
[4]	P01_IE	P0.1 INT1 Enable 0: Disable 1: Enable
[3]	P03_IE	P0.3 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 EXT1_IF (0xD2)

Bit	7	6	5	4	3	2	1	0
Name	P07_IF	P06_IF	P05_IF	P01_IF	P03_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]	P07_IF	P0.7 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[6]	P06_IF	P0.6 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending

[5]	P05_IF	P0.5 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[4]	P01_IF	P0.1 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[3]	P03_IF	P0.3 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 INT_SR0 (0xF2)

Bit	7	6	5	4	3	2	1	0
Name	RSV	T1ABDIF	T1ADIF	T1BOIF	T1ROIF	T1WTIF	T1PDIF	T1BDIF
Type	-	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Reset	-	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	T1ABDIF	Adaptive Diode Freewheeling Masking Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[5]	T1ADIF	ADC Position Detection Interrupt Flag A position detection interrupt is generated when TIM1_DBRx[T1CPE] matches ACD position detection signal. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	T1BOIF	Base Timer Overflow Interrupt Flag An overflow event occurs when Base Timer counts up and TIM1__BCNTR matches with TIM1__BARR. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

[3]	T1ROIF	Reload Timer Overflow Interrupt Flag An overflow event occurs and TIM1__RCNTR is cleared to “0” when TIM1__RCNTR matches TIM1__RARR. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	T1WTIF	Writing Timing Interrupt Flag Writing timing interrupt is generated when TIM1_DBRx is transferred to DRV_CMR. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	T1PDIF	CMP/GPIO Position Detection Interrupt Flag A position detection interrupt is generated when CMP/GPIO position detection signal matches TIM1_DBRx[T1CPE]. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	T1BDIF	Diode Freewheeling Masking End Interrupt Flag Diode freewheeling masking starts after phase commutation and an interrupt is generated at end. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

7.5.10 INT_SR1 (0xF3)

Bit	7	6	5	4	3	2	1	0
Name	RSV		T3IR	T3IP	RSV	T3IR	T3IP	T3IF
Type	-	-	R/W0	R/W0	-	R/W0	R/W0	R/W0
Reset	-	-	0	0	-	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	T4IR	Timer4 Compare Match/Pulse Width Detection Interrupt Flag Output Mode: Compare Match Interrupt Flag Input Capture Mode: Pulse Width Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

[4]	T4IP	Timer4 PWM Cycle Detection Interrupt Flag Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[3]	RSV	Reserved
[2]	T3IR	Timer3 Compare Match/Pulse Width Detection Interrupt Flag Output Mode: Compare Match Interrupt Flag Input Capture Mode: Pulse Width Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	T3IP	Timer3 PWM Cycle Detection Interrupt Flag Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	T3IF	Timer3 Overflow Interrupt Flag Output Mode: Base Timer Overflow Interrupt Flag, which is set to”1” when TIMx__CNTR matches TIMx__ARR. Input Capture Mode: Base Timer Overflow Interrupt Flag, which is set to “1” when the Timer does not detect an input PWM cycle but TIMx__CNTR reaches 0xFFFF. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

7.5.11 INT_SR2 (0xF4)

Bit	7	6	5	4	3	2	1	0
Name	RSV	RTC_IF	SYSTIF	RSV	DCIF	T2IR	T2IP	T2IF
Type	-	R/W0	R/W0	-	R/W0	R/W0	R/W0	R/W0
Reset	-	0	0	-	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	RTC_IF	RTC Interrupt Flag This bit is set to “1” when the timer value reaches RTC_TM. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

[5]	SYSTIF	Systick Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	RSV	Reserved
[3]	DCIF	Driver Compare Match Interrupt Flag When the Driver counter value is equal to DRV_COMR, the system decides whether to generate an interrupt according to the counting direction set by DRV_SR[DCIM]. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	T2IR	Timer2 Compare Match/Pulse Width Detection/QEP ZCP Detection Interrupt Flag Output Mode: Compare Match Interrupt Flag Input Capture Mode: Pulse Width Detection Interrupt Flag Input Count Mode: No effect QEP Mode: QEP Encoder ZCP Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	T2IP	Timer2 PWM Cycle Detection/PWM Input Count Match/Active Edge Detection Interrupt Flag Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag Input Count Mode: PWM Input Count Match Interrupt Flag QEP&RSD Mode and Step Mode: Active Edge Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	T2IF	Timer2 Overflow Interrupt Flag Output Mode: Base Timer Overflow Interrupt Flag, which is set to “1” when TIM2__CNTR matches TIM2__ARR. Input Capture Mode: Base Timer Overflow Interrupt Flag, which is set to “1” when the Timer has not detected an input PWM cycle but the timer TIM2__CNTR reaches 0xFFFF. Input Count Mode: Special-purpose Timer Overflow Interrupt Flag, which is set to “1” when the input PWM cycle has not reached the preset TIM2__DR value but the Base Timer TIM2__CNTR value reaches 0xFFFF. QEP&RSD Mode and Step Mode: Base Timer Overflow Interrupt Flag, which is set to “1” and Base Timer is cleared to “0” when Base Timer reaches to 0xFFFF. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

7.5.12 INT_SR3 (0xF5)

Bit	7	6	5	4	3	2	1	0
Name	RSV	TSDIF	HALL_IF	CMP3IF	CMP2IF	CMP1IF	CMP0IF	LVWIF
Type	-	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Reset	-	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	TSDIF	TSD Interrupt Flag This bit is set to “1” by hardware when the chip exceeds the specified temperature. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect Note: This bit always works with LVSR[TSDF].
[5]	HALL_IF	Hall Interrupt Flag Read: 0: No Hall edge change is detected 1: Hall edge change is detected Write: 0: This bit is cleared to “0” 1: No effect
[4]	CMP3IF	CMP3 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[3]	CMP2IF	CMP2 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	CMP1IF	CMP1 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	CMP0IF	CMP0 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	LVWIF	VCC LVW Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending

		Write: 0: This bit is cleared to “0” 1: No effect Note: This bit is not set to “1” by hardware when LVD interrupt is disabled.

7.5.13 INT_SR4 (0xF6)

Bit	7	6	5	4	3	2	1	0
Name	OVPIF	OCPIF	RSV		DMA1IF	DMA0IF	IF0	ADCIF
Type	R/W0	R/W0	-	-	R/W0	R/W0	R/W0	R/W0
Reset	0	0	-	-	0	0	0	0

Bit	Name	Description
[7]	OVPIF	OVP Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	OCPIF	OCP Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[5:4]	RSV	Reserved
[3]	DMA1IF	DMA Channel 1 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	DMA0IF	DMA Channel 0 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	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
[0]	ADCIF	ADC Interrupt Flag This bit is set to “1” by hardware when ADC conversion is completed. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

7.5.14 INT_SR5 (0xF7)

Bit	7	6	5	4	3	2	1	0
Name	RSV			SPIIF	WCOL	MODF	RXOVRN	I2CIF
Type	-	-	-	R/W0	R/W0	R/W0	R/W0	R
Reset	-	-	-	0	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	SPIIF	SPI Interrupt Flag This bit is set to “1” by hardware each time after a data frame (8-bit) is transferred. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[3]	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. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	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. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	RXOVRN	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 automatically cleared to “0” by hardware, and can be cleared to “0” by software only. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	I2CIF	I^2C Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending When I2C_SR[STR] = 1, an interrupt is generated in both Master and Slave modes. When I2C_SR[I2CSTP] = 1, an interrupt is generated only in Slave mode.

7.5.15 INT_SR6 (0xFA)

Bit	7	6	5	4	3	2	1	0
Name	RSV			LINERR	LINWAKEUP	LINIDLE	LINDONE	LINREQ
Type	-	-	-	R/W0	R/W0	R/W0	R/W0	R/W0
Reset	-	-	-	0	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	LINERR	LIN Error Interrupt Flag Read: 0: No error occurs. 1: An error occurs. Write: 0: The error bit is cleared. 1: No effect
[3]	LINWAKEUP	LIN Bus Wakeup Interrupt Flag This bit is set to “1” when the bus line stays idle for more than 4s. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	LINIDLE	LIN Bus Idle Interrupt Flag This bit is set to “1” when the bus line stays idle for more than 4s. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	LINDONE	Transmission Completion Interrupt Flag This bit is set to “1” after the slave receives or sends the data, and cleared to “0” when a new frame arrives or software writes “0” to LIN_SR[LINDONE]. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	LINREQ	Header Reception Interrupt Flag This bit is set to “1” after a frame header is received and its ID is correct. This bit is cleared to “0” when a new frame arrives, or software writes “1” to LIN_CSR[LINACK] or “0” to LIN_SR[LINREQ]. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

8 Clock Gating

Timer2, Timer3, Timer4, I²C, LIN, SPI and UART modules are designed with a separate clock gating circuit. The corresponding clock enable bit in CK_CR register controls clock gating feature of these modules.

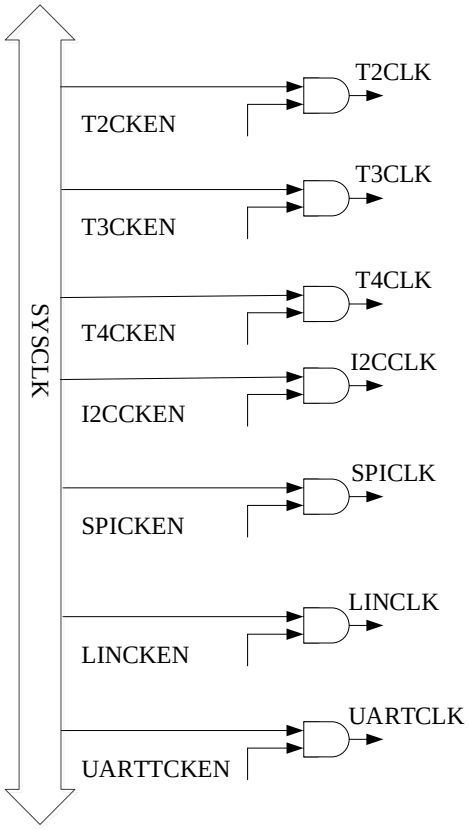


Figure 8-1 Clock Tree

8.1 Clock Gating Register

8.1.1 CK_CR (0x91)

Bit	7	6	5	4	3	2	1	0
Name	RSV	UARTCKEN	LINCKEN	SPICKEN	I2CCKEN	T4CKEN	T3CKEN	T2CKEN
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]	UARTCKEN	UART Clock Enable 0: Disable 1: Enable
[5]	LINCKEN	LIN Clock Enable 0: Disable 1: Enable
[4]	SPICKEN	SPI Clock Enable 0: Disable 1: Enable
[3]	I2CCKEN	I²C Clock Enable 0: Disable 1: Enable

[2]	T4CKEN	Timer4 Clock Enable 0: Disable 1: Enable
[1]	T3CKEN	Timer3 Clock Enable 0: Disable 1: Enable
[0]	T2CKEN	Timer2 Clock Enable 0: Disable 1: Enable

9 I²C

9.1 I²C Introduction

The I²C module provides an industry standard two-wire serial interface and is a simple bi-directional synchronous serial bus for communication between MCU and external I²C devices, as shown in Figure 9-1. The bus consists of two serial lines: SDA (serial data line) and SCL (serial clock line). P0.2 serves as SDA port and P0.3 as SCL port. After I²C is enabled, P0.2/P0.3 automatically shift into open-drain outputs.

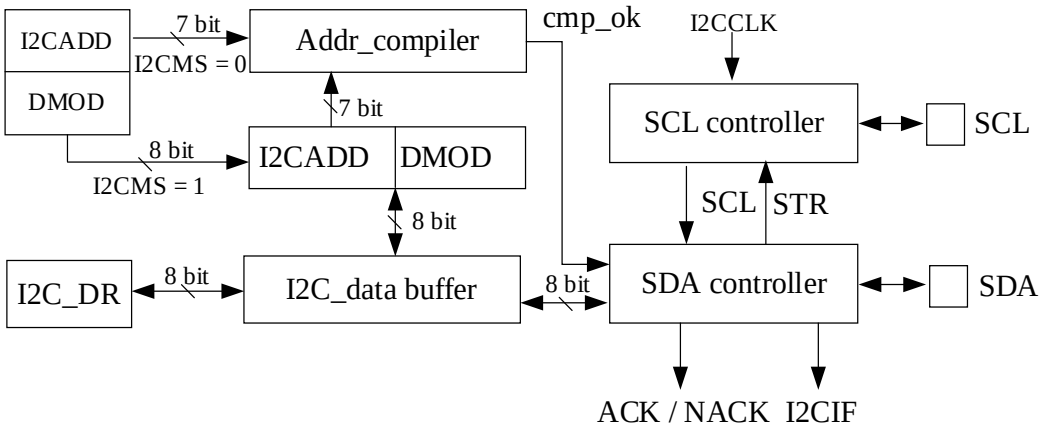


Figure 9-1 I²C Block Diagram

Main features:

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

Both SDA and SCL lines are high level when the bus is idle, which is the only basis for detecting whether the bus is idle or not. Only one master device and at least one slave device are active on the bus during the transmission. When the bus is occupied, other devices must wait for the bus idle to start an I²C communication. The master starts the bus to transfer data. Clock signal is sent to all devices via SCL and the slave address and read/write mode are sent via SDA. When a device on the bus matches the address, it acts as a slave. The relationships between masters and slaves or data transfer direction on the bus are not constant. The process for the master to send data to the slave is shown in Figure 9-2. The master first addresses the slave and waits for its response. And then, it sends data to the slave. Finally, the master terminates the data transmission. The process for the master to receive data from the slave is shown in Figure 9-3. The master first addresses the slave and waits for its response. And then, it receives the data from the slave. Finally, the master terminates the data transmission. In this case, the master generates the timing clock and stops the data transmission.

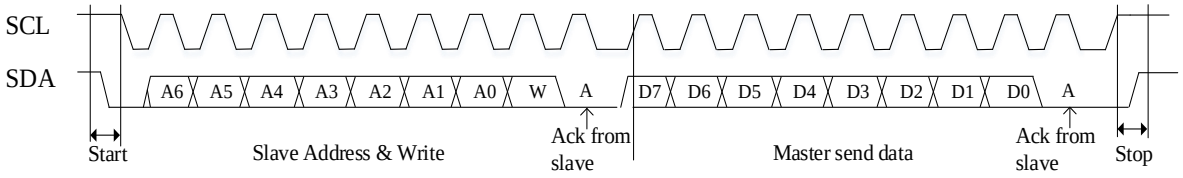


Figure 9-2 Master Transmits Data to Slave

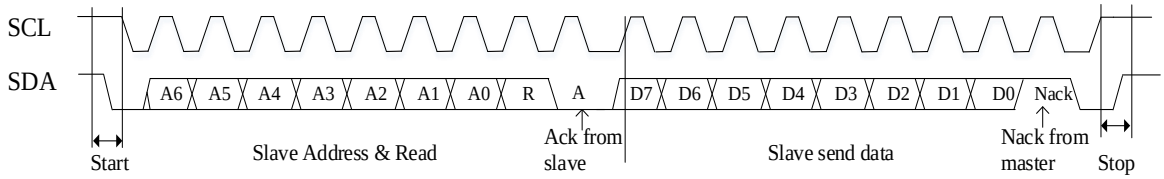


Figure 9-3 Master Receives Data from Slave

9.2 I²C Operations

9.2.1 Master Mode

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

9.2.2 Slave Mode

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

call mode;

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

9.2.3 I²C Interrupt Sources

The interrupt sources of I²C include:

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

9.3 I²C Registers

9.3.1 I2C_CR (0x4028)

Bit	7	6	5	4	3	2	1	0
Name	I2CEN	I2CMS	EROMVDD5PD	I2CDMANAKINT	I2CDMAAUTO	I2CSPD		I2CIE
Type	R/W	R/W	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]	I2CEN	I²C Enable Enable the associated GPIO and switch to I²C mode, serving as open-drain output. The pull-up setting decides whether to pull I²C HIGH. 0: Disable 1: Enable
[6]	I2CMS	Master/Slave Mode Selection 0: Slave 1: Master
[5]	EROMVDD5PD	EEPROM 5V Power Supply or Power Down 0: EEPROM 5V Power Supply 1: EEPROM Power Down
[4]	I2CDMANAKINT	An interrupt is generated when the device does not acknowledge (NAK) the received the data during DMA transmission. 0: Disable 1: Enable
[3]	I2CDMAAUTO	Automatically transfer the first byte of data during DMA transmission 0: Disable 1: Enable
[2:1]	I2CSPD	I²C transfer rate setting, valid only in Master Mode 00: 100kHz 01: 400kHz 10: 1MHz 11: Reserved
[0]	I2CIE	I²C Interrupt Enable 0: Disable 1: Enable Note: See INT_SR5 (0xF7) for I²C Interrupt Flag Bit.

9.3.2 I2C_ID (0x4029)

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

Bit	Name	Description
[7:1]	I2CADD	Slave address
[0]	GC	General call, valid only in Slave Mode 0: General call is disabled 1: General call is enabled, namely, i.e., the receiving device also responds at address 0x00.

9.3.3 I2C_DR (0x402A)

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

Bit	Name	Description
[7:0]	I2C_DR	I^2C Data Register Read: Data to be sent or received Write: Data to be sent

9.3.4 I2C_SR (0x402B)

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

Bit	Name	Description
[7]	I2CBSY	I^2C Busy Flag When I2C_CR[I2CEN] = 0, I2C_SR[I2CBSY] is cleared to "0" by hardware. **Master Mode:** After START is transmitted, I2C_SR[I2CBSY] is set to "1" by hardware; after STOP is transmitted, I2C_SR[I2CBSY] is cleared to "0" by hardware. **Slave Mode:** After START is received and address matches, I2C_SR[I2CBSY] is set to "1" by hardware; after STOP is received, I2C_SR[I2CBSY] is cleared to "0" by hardware.
[6]	DMOD	I^2C R/W Flag 0: WRITE (master transmits the data, and slave receives the data) 1: READ (master receives the data, and slave transmits the data) Note: Read only in Slave Mode
[5]	RSV	Reserved
[4]	I2CSTA	**Master Mode:** When this bit is set to "1" by software, START and address bytes are transmitted after both SCL and SDA are HIGH confirmed by the hardware. This bit is cleared to "0" by hardware automatically when the transmission is completed, and I2C_SR[I2CSTA] writing is forbidden during data transmission. After the data is sent or received, I2C_SR[I2CSTA] is set to "1" to transmit RESTART. 0: Not START and address bytes 1: Transmit START or RESTART and address bytes **Slave Mode:** This bit is set to "1" after hardware receives START and address matches, and cleared to "0" by software. Table 9-1 Mapping of I2C_SR[I2CSTA] and I2C_SR[I2CSTP] with I^2C Data Type in Slave Mode

I2CSTA	I2CSTP	I^2C Data Type
0	0	Data byte
0	1	STOP
1	0	START + address bytes
1	1	STOP received first, then START + address bytes

Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTA] is automatically cleared to "0".

[3]	I2CSTP	**Master Mode:** This bit cannot be written to "1" by software unless I2C_SR[I2CBSY] = 1; STOP is transmitted after I2C_SR[STR] is cleared to release SCL. After the transmission, this bit is cleared to "0" automatically by hardware. If I2C_SR[I2CSTA] and I2C_SR[I2CSTP] are written to "1" at the same time and I2C_SR[I2CBSY] is "1", I^2C first sends STOP, then START and address bytes. After START and address bytes are transmitted, I2C_SR[STR] is set to "1" by hardware. I2C_SR[I2CSTP] writing is forbidden during data transmission. 0: STOP is not transmitted. 1: STOP is transmitted. **Slave Mode:** This bit is set to "1" by hardware after STOP is received, and cleared to "0" by software. See Table 9-1 for status flags. Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTP] is automatically cleared to "0" by hardware.
[2]	STR	I^2C Bus Pending Flag **Master Mode:** After START and address or DATA byte are transmitted, I2C_SR[STR] are set to "1" by hardware and SCL is pulled LOW. SCL is released after I2C_SR[STR] is cleared by software. When I2C_SR[I2CSTA] and I2C_SR[I2CSTP] are both "1", I2C_SR[STR] is set to "1" only after hardware sends STOP and START & address bytes. **Slave Mode:** When DATA byte is received or START is received and address matches, I2C_SR[STR] is set to "1" by hardware and SCL is pulled LOW. SCL is released after I2C_SR[STR] is cleared by software. Note: This bit is set to "1" by hardware and cleared to "0" by software. When I2C_CR[I2CEN] = 0, I2C_SR[STR] is automatically cleared to "0".
[1]	NACK	This bit refers to the feedback from a receiver to a sender after a byte is transferred via I^2C. It is automatically cleared to "0" when I2C_SR[I2CEN] = 0. 0: ACK, indicating that the receiver can continue to receive data. 1: NACK, indicating that the receiver attempts to stop data transmission. When the device is in READ mode, I2C_SR[NACK] is configured to transmit ACK/NACK after the 8^{th} bit of data is received. 0: Bit9 transmits ACK 1: Bit9 transmits NACK When the device is in WRITE mode, I2C_SR[NACK] is read to receive ACK/NACK after the 8^{th} bit of data is transmitted. 0: Bit9 receives ACK 1: Bit9 receives NACK
[0]	RSV	Reserved

10 SPI

10.1 SPI Introduction

SPI provides access to a high-speed and full-duplex synchronous serial bus, with its block diagram shown in Figure 10-1. 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.

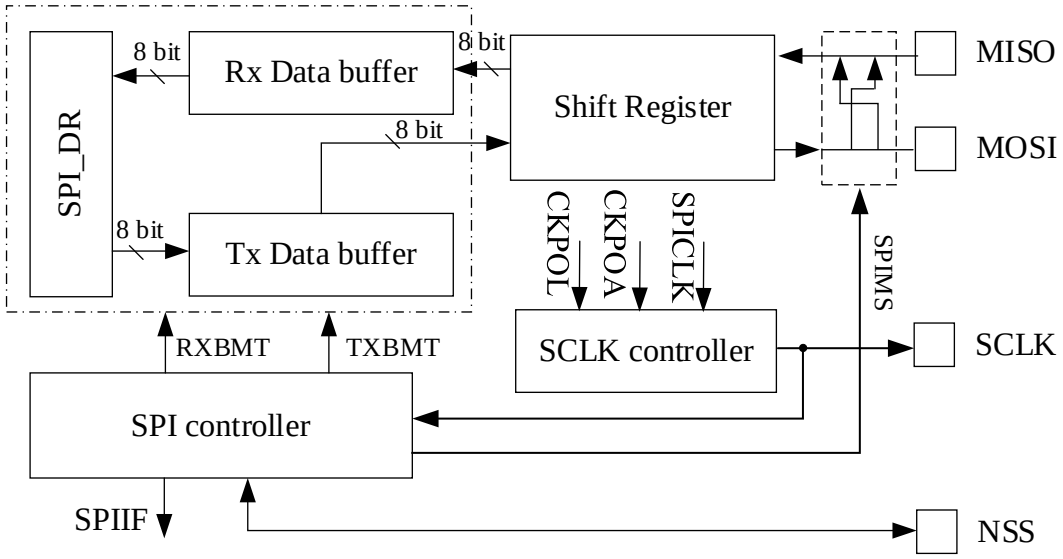


Figure 10-1 SPI Block Diagram

10.2 SPI Operations

10.2.1 Signal Descriptions

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

10.2.1.1 Master Out, Slave In (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.

10.2.1.2 Master In, Slave Out (MISO)

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

10.2.1.3 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. In 4-wire slave mode, SCLK is ignored when the slave device is not selected (NSS=1).

10.2.1.4 Slave Select (NSS)

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

When SPI_CR1[NSSMOD] = 00, SPI operates in 3-Wire Mode. NSS port is not necessary in this mode and there is only one master and one slave on the SPI bus. The connection diagram is shown in Figure 10-2.

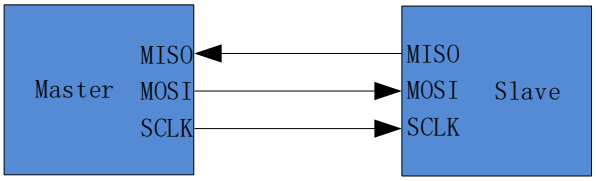


Figure 10-2 Connection Diagram of 3-Wire SPI Mode

When SPI_CR1[NSSMOD] = 01, SPI operates in 4-Wire Slave/Multi-Master Mode. In this mode, NSS pins on the SPI bus are configured as inputs, waiting to be addressed by the master. When SPI_CR0[SPIMS] = 0, SPI operates in 4-Wire Slave Mode. If NSS is set to “0”, the slave is selected; while NSS is set to “1”, the slave is not selected. When SPI_CR0[SPIMS] = 1, SPI operates in Master Mode and defaults to Multi-Master Mode. When SPI operates in Multi-Master Mode, NSS is configured as an input to disable the master SPI. When NSS pin of a master on the SPI bus is pulled low, SPI_CR0[SPIMS] and SPI_CR1[SPIEN] are cleared to “0” by hardware to disable the SPI, and the Mode Fault Flag INT_SR5[MODF] is set to “1”. In this case, SPI communication remains halted before the SPI is re-enabled by software. In this mode, multiple masters are allowed for communication on the SPI bus. The connection diagram is shown in Figure 10-3.

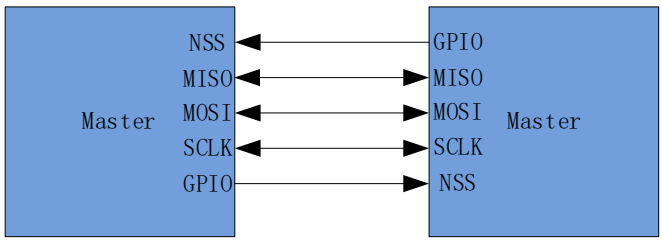


Figure 10-3 Connection Diagram of 4-Wire Multi-Master Mode

When SPI_CR1[NSSMOD] = 1X, SPI operates in 4-Wire Single Master Mode. In this mode, NSS pin of the master on the bus is configured as an output, and NSS pin of the slave devices are configured as inputs. SPI_CR1[NSSMOD0] setting decides the output level of NSS pin serving as signal to select a slave. Other slaves can be selected using GPIO pins. The connection diagram is shown in Figure 10-4.

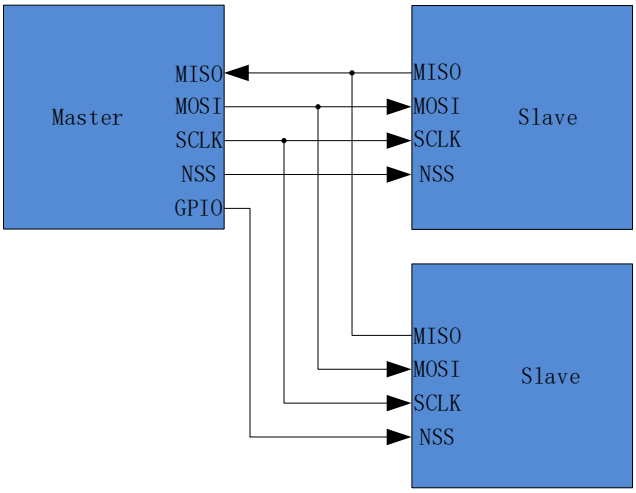


Figure 10-4 Connection Diagram of 4-Wire Single Master Mode

10.2.2 SPI Master Mode

When SPI_CR0[SPIMS] = 1, SPI operates in master mode, which provides SCLK signal for the bus. When the data is written to SPI_DR, it is firstly written to the transmit buffer and SPI_CR1[TXBMT] is cleared to “0”. If the shift register is empty, the data in the transmit buffer is 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, INT_SR5 [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, INT_SR5 [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 INT_SR5[WCOL] is set to “1” and the data in the transmit buffer keeps unchanged.

10.2.2.1 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 SCLK 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 INT_SR5 [SPIIF] is set to "1", SPI_DR is read to receive the data.

10.2.3 SPI Slave Mode

When SPI_CR0[SPIMS] = 0, SPI operates in slave mode. In this mode, SCLK signal is sent by the master SPI. The data is shifted in from MOSI pin and shifted out from MISO pin. If no SCLK signal is input, shift register of the slave 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, INT_SR5[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, INT_SR5[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 INT_SR5[WCOL] is set to "1" and the data in the transmit buffer keeps unchanged.

10.2.3.1 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.

10.2.4 SPI Interrupt Sources

The interrupt sources of SPI include:

- SPI interrupt flag INT_SR5[SPIIF] is set to "1" each time after the byte is transferred.
- If SPI_DR is written when the data in transmit buffer has not been transferred to the shift register, the

write conflict flag INT_SR5[WCOL] is set to "1" and the write operation will not be implemented.

- When SPI works as a master in a multi-master system and NSS pin is pulled LOW, the mode error flag INT_SR5[MODF]is set to "1". When a mode error occurs, SPI_CR0[SPIMS] and SPI_CR1[SPIEN] are cleared. SPI is forbidden to allow another master to control the bus.
- The receive overflow flag INT_SR5[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.

10.2.5 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). The timing relationships of SCL and SDA in clock phase and polarity combinations are shown in Figure 10-5 and Figure 10-6.

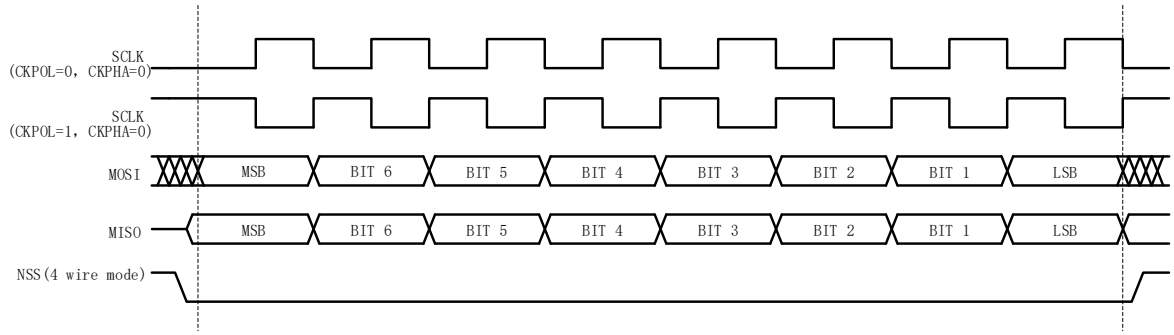


Figure 10-5 SDA/SCL Line Timing Diagram (SPI_CR0[CPHA] = 0)

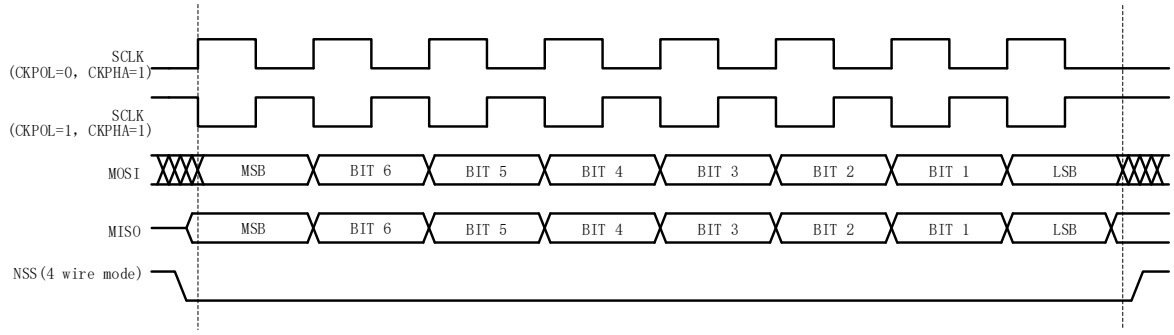


Figure 10-6 SDA/SCL Line Timing Diagram (SPI_CR0[CPHA] = 1)

10.3 SPI Registers

10.3.1 SPI_CR0 (0x4030)

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

Bit	Name	Description
[7]	SPIBSY	SPI Busy Flag 0: No data is transferring via SPI transfer. 1: Data is transferring via SPI.
[6]	SPIMS	Master/Slave Mode Selection 0: Slave 1: Master
[5]	CPHA	Clock Phase 0: Data is received on leading edge and transmitted on trailing edge of active SCLK. 1: Data is transmitted on leading edge and received on trailing edge of active SCLK.
[4]	CPOL	Clock Idle Polarity 0: Low level in idle state. 1: High level in idle state.
[3]	SLVSEL	NSS Select Flag This bit is set to “1” when the filtered signal of NSS is low, indicating that the device is selected as slave. When NSS is high, this bit is cleared to “0”, indicating that the device is not selected as slave. 0: The device is not selected as slave 1: The device is selected as slave
[2]	NSSIN	NSS real-time signal, unfiltered
[1]	SRMT	Shift Register Empty Flag (valid only in Slave Mode) 0: Data has been shifted out of the Transit Buffer into the shift register or SCLK changes. 1: There is no data in the shift register or transmit and receive buffer. Note: SPI_CR0[SRMT] = 1 in Master Mode.
[0]	RXBMT	Receive Buffer Empty Flag (valid only in Slave Mode) 0: New data in the receive buffer has not been read. 1: Data has been read and there is no new data in the receive buffer. Note: SPI_CR0[RXBMT] = 1 in Master Mode.

Notes: Clock phase and idle polarity modes SPI_CR0[CPHA:CPOL]:

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

10.3.2 SPI_CR1 (0x4031)

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

Bit	Name	Description
[7:4]	RSV	Reserved
[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 or Multi-Master Mode (Default). 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 byte is written to the transmit buffer. 1: Data in the transmit buffer has been transferred to the shift register.
[0]	SPIEN	SPI Enable 0: Disable 1: Enable

10.3.3 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 = SPICLK/2/(SPI_CLK + 1) Example: If baud rate = 2000kHz, then SPI_CLK = (20M/2/2000k) – 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 2MHz to prevent erroneous data transmitted from the slave SPI.

10.3.4 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. Read: Receive the data from Receive Buffer Write: Write the data into Transit Buffer and initiate a transfer

11 UART

11.1 UART Introduction

UART is a full-duplex or half-duplex serial data exchange interface as shown in Figure 11-1. The baud rate is configurable and supports DMA transmission. Figure 11-2 depicts the UART timing.

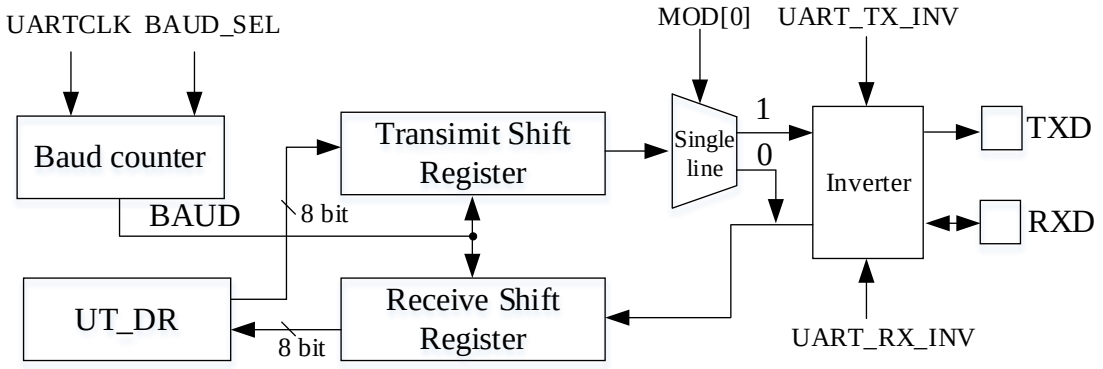


Figure 11-1 UART Block Diagram

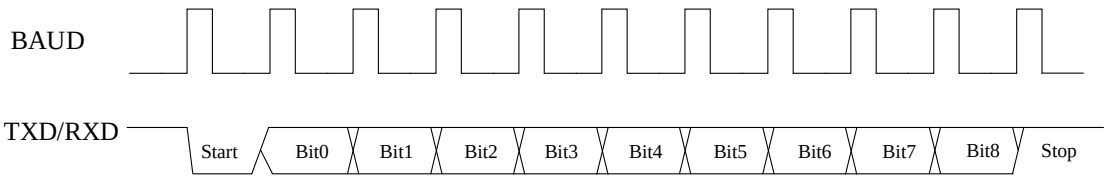


Figure 11-2 UART Timing Diagram

11.2 UART Operating Instructions

The corresponding registers shall be enabled before using UART feature. See 23.3.7 PH_SEL (0x404C) (bit [6]) ~ 23.3.8 PH_SEL1 (0x404D) (bit [7:6]) for details.

11.2.1 UART Mode0

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

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

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

11.2.2 UART Mode1

UART 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 and 1-bit stop) to

receive or transmit data. The baud rate is configured by UT_BAUD[BAUD].

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

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

11.2.3 UART Mode2

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

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

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

11.2.4 UART Mode3

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

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

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

11.2.5 UART Interrupt Sources

UART interrupt sources include:

- After the data is transmitted via UART, UT_CR[TI] is set to "1" by hardware.
- After the data and STOP are received via UART, UT_CR[RI] is set to "1" by hardware.

11.3 UART Registers

11.3.1 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: Mode0 01: Mode1 10: Mode2 11: Mode3
[5]	SM2	Communication Mode Selection 0: Single-device Communication 1: Multi-device Communication
[4]	REN	Serial Input Enable 0: Disable 1: Enable
[3]	TB8	Bit9 of the Transmitted Data in Mode2 and Mode3
[2]	RB8	Bit9 of the Transmitted Data in Mode2 and Mode3
[1]	TI	Data Transmitting Completed Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: The interrupt event is generated.
[0]	RI	Data Receiving Completed Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: The interrupt event is generated.

11.3.2 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	Transmit/ Receive Data Read: Data received Write: Data to be transmitted Note: UART data buffer consists of two independent buffers, i.e., a receive buffer and a transmit buffer, which can send and receive data at the same time. The transmit buffer can be written only but not read, while the receive buffer can be read only but not written. Both buffers share a same address.

11.3.3 UT_BAUD (0x9A, 0x9B)

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

Bit	**Name**	**Description**
[15]	BAUD_SEL	Frequency Multiplier Enable 0: Disable 1: Enable
[14]	UART_RX_INV	Receive Inverting Enable 0: Disable 1: Enable
[13]	UART_TX_INV	Transmit Inverting Enable 0: Disable 1: Enable
[12]	RSV	Reserved
[11:0]	BAUD	Baud Rate Setting Baud rate = UARTCLK/(16/(1 + UT_BAUD[BAUD_SEL]))/(UT_BAUD[BAUD] + 1) Example: If baud rate = 9600 and UT_BAUD[BAUD_SEL] = 0, then UT_BAUD[BAUD] = (20M/16/9600/(1 + 0)) – 1 = 129, i.e., 0x81

12 LIN

12.1 LIN Introduction

LIN is an asynchronous and serial communication interface mainly used in automotive network. It complies with relevant test specifications of ISO 17987, where the built-in LIN transceiver and LIN data link layer are implemented in accordance with ISO 17987 standard and qualified for the related tests. As shown in Figure 12-1, the controller owns a complete LIN hardware interface, which works in slave mode and supports baud rate adaption. P0.0/P0.1 and P1.0/P1.1 (function switching) are translated into LIN pins, among which P1.1 is configured as RXD/TXD pin. Data transmission between LIN and DMA is possible by setting DMAx_CR0[DMACFG] = 11X, and LIN_CR[LINRW] bit determines the R/W direction.

For more information and specifications regarding LIN protocol, see LIN Consortium (http://www.lin-subbus.org).

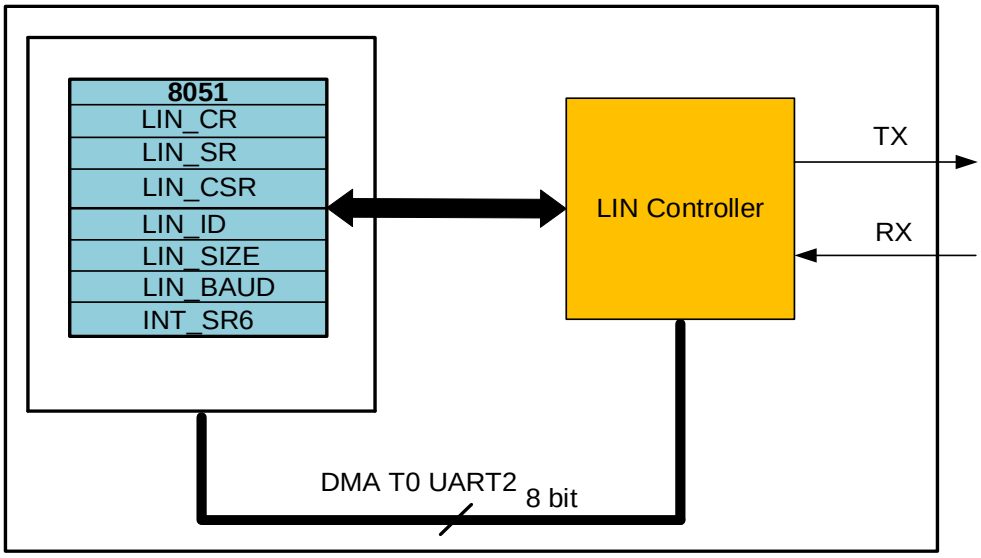


Figure 12-1 LIN Block Diagram

The LIN controller has three main components, as shown in Figure 12-1.

- LIN access register: Provide the interface between the CPU and the LIN controller through XSFR addressing of 8051 core.
- LIN data buffer: Transmit and receive the data by configuring DMA0/1.
- LIN control register: Process data transmission and control states of the LIN bus.

12.2 LIN Slave Mode Operations

When LIN is configured to operate in slave mode, it must wait for a command from a master node.

When LIN Interrupt is enabled, an interrupt is generated in any of the following five cases as shown in Table 12-1.

Table 12-1 LIN Interrupt Sources and Descriptions

Interrupt Source	Description	Interrupt Flag	Clear Flag
Bus Idle	Bus remains idle for 4s or more.	INT_SR6[LINIDLE]	Write “0” to INT_SR6[LINIDLE]
External Wakeup	Wake signal is received.	INT_SR6[LINWAKEUP]	Write “0” to INT_SR6[LINWAKEUP]
Reception of Frame Header	Frame header is received and ID check is correct.	INT_SR6[LINREQ]	Write “1” to LIN_CSR[LINACK]/ Write “0” to INT_SR6[LINREQ]/ A new frame header is detected
Data Transmission or Reception Completed	Data is received or sent by the salve.	INT_SR6[LINDONE]	Write “0” to LIN_SR[LINDONE]/ A new frame header is detected
Error Occurs	An error interrupt request is received: Bit error/Format error Sync error/ ID check error/ Data check error	LIN_SR[ERRBIT] LIN_SR[ERRSYNC] LIN_SR[ERRPRTY] LIN_SR[ERRCHK]	Write “0” to INT_SR6[LINERR]/ A new frame header is detected

LIN slave transmits and receives the data as follows:

1. LIN controller detects the header (Synch Break Field and Synch Field signals) of a message frame from the master on LIN bus. The baud rate of the data is automatically identified by the synchronization signal. INT_SR6[LINREQ] is set to “1” when the slave recognizes the ID and the ID checksum is correct. Otherwise, LIN_SR[ERRPRTY] is set to “1” and ID checksum error occurs;
2. Data Transmission: Set LIN_CR[LINRW] to “1” to load the data length into LIN_SIZE and data bytes into DMA buffer. Set LIN_CSR[LINACK] to “1” and frame header transfers data to the master;
3. Data Reception: Clear LIN_CR[LINRW] to “0” and set LIN_CSR[LINACK] to “1”, and frame header receives the data sent by the master;
4. INT_SR6[LINDONE] is set to “1” after the slave receives or transmits the data.

12.3 Sleep and Wakeup

To reduce the system’s power consumption, LIN Protocol Specification defines a Sleep Mode.

After the slave receives and correctly decodes a Sleep Mode request from the master, the software puts the device into the Sleep Mode by setting LIN_CSR[LINSLP] to “1”.

INT_SR6[LINIDLE] is set to “1” when the bus stays idle for more than 4s and the LIN slave is not in the sleep mode. In this case, the software may assume that the LIN bus is in Sleep Mode and put the device into the Sleep Mode by setting LIN_CSR[LINSLP] to “1”.

Sending a wake-up signal from the master or any slave node (setting INT_SR6[LINWAKEUP] to “1”) terminates the Sleep Mode of the LIN bus. The LIN slave can also send a wake-up signal (setting LIN_CSR[TXWAKEUP] to “1”) to wake up the master or other slaves.

12.4 Error Detection and Handling

When LIN slave detects an error, INT_SR6[CLRERR] is set to "1", and LIN generates an error interrupt request and stops the processing of current frame. The type of error, i.e., bit error/format error, sync error, data check error or ID check error, is determined via LIN_SR[ERRBIT], LIN_SR[ERRSYNC], LIN_SR[ERRCHK] and LIN_SR[ERRPRTY]. INT_SR6[CLRERR] is cleared to "0" after the error is processed.

12.5 LIN Auto-addressing

When LIN slave operates in single-wire mode, the auto-addressing feature is implemented by LININ and LINOUT pins, as shown in Figure 12-2.

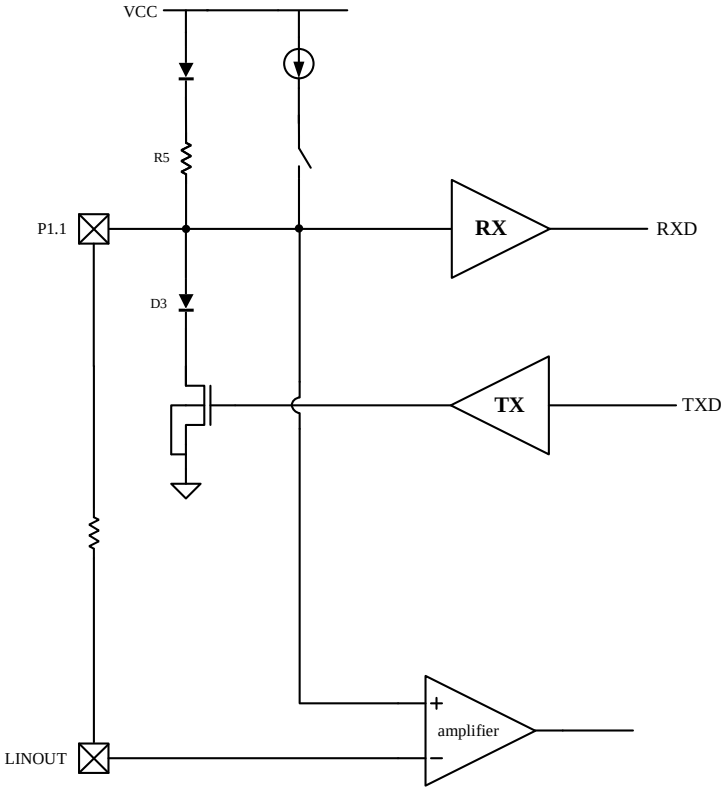


Figure 12-2 Schematic Diagram LIN Auto-addressing

The on-off time for current sources in LIN auto-addressing mode is related to baud rate. The baud rate of LIN module shall be configured properly and match with that of the master device, ensuring on-off actions of the current sources are synchronized to avoid timing problems during sampling.

All nodes must stop the motor and all LIN services before LIN auto-addressing.

Operation procedures of LIN auto-addressing are as follows:

1. Configure LIN_CSR[LINADADR] and LIN_CSR[LINADADR_CUR] to "1" to implement initialization, so that pull-up resistors are switched off and current sources are switched on when the slave node is addressed.
2. When the master node sends LIN message, the slave node is addressed during synch break field. If the

addressing is successful, LIN_SR[LINADADR_ISME] is set to “1”.

3. After the addressing, LIN_CSR[LINADADR_CUR] in the slave device is set to “0” to keep its current sources switched off.
4. After all slave nodes are addressed, LIN_CSR[LINADADR] in all slave nodes is set to “0”.

12.6 Baud Rate Modes

LIN module supports automatic baud rate mode and fixed baud rate mode to fulfill LIN auto-addressing feature and other features.

Automatic Baud Rate Mode: LIN module detects LIN bus at a frequency of 20k and enters data receiving mode if a synch break field greater than 11-bit is detected. When the byte "0x55" is identified in the synch field, LIN module calculates baud rate of LIN message based on the field length, and sends or receives the next LIN message based on this baud rate.

Fixed Baud Rate Mode: LIN module operates in fixed baud rate mode when LIN_CR[AUTOBAUD] is set to “1”. The baud rate (1000-20000) is configured by LIN_BAUD register, and the received baud rate is generally ±10% of the configured value. LIN module receives LIN message according to the configured baud rate. If LIN message is carried with unmatched baud rate, sync error occurs and LIN module refuses receiving the LIN message.

12.7 Other Matters

When LIN slave mode is enabled and the device is not in the Sleep Mode, the slave may detect a new frame header (including Synch Break Field, Synch Field and PID).

Configuring LIN_CSR[LINSTOP] to “1” aborts the processing of the current frame during data reception or transmission at slave mode. LIN_SR[ABORT] is set to “1”.

12.8 LIN Registers

12.8.1 LIN_CR (0xB9)

Bit	7	6	5	4	3	2	1	0
Name	MBAUD	DMASEL	RSV	CHSEL	LINIE	CHKMOD	LINRW	AUTOSIZE
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]	MBAUD	Fixed Baud Rate Mode Selection 0: Automatic Baud Rate Mode 1: Fixed Baud Rate Mode. The received baud rate is generally ±10% of the configured value.
[6]	DMASEL	LIN DMA Channel Selection 0: DMA0 1: DMA1
[5]	RSV	Reserved
[4]	CHSEL	LIN Mode and Port Pin Configuration 0: Two-wire mode, P0.0 configured as LTXD pin and P0.1 as LRXD pin 1: Single-wire mode, P1.1 configured as LTXD/LRXD pin
[3]	LINIE	LIN Interrupt Enable 0: Disable 1: Enable Note: See INT_SR6 (0xFA) for LIN Interrupt Flag Bit.
[2]	CHKMOD	Checksum Selection 0: Enhanced checksum 1: Classic checksum
[1]	LINRW	Transmit/Receive Selection 0: Current frame is a receive operation 1: Current frame is a transmit operation
[0]	AUTOSIZE	Data Length Enable Dependent on ID Bit (LIN_ID[5:4]) 0: Disable 1: Enable Mapping between LIN_ID[5:4] and Data Length: 0X: 2 bytes 10: 4 bytes 11: 8 bytes

12.8.2 LIN_SR (0xBA)

Bit	7	6	5	4	3	2	1	0
Name	RSV	LINADADR_ISME	ERRBIT	ERRSYNC	ERRCHK	ERRPRTY	ABORT	LINACT
Type	-	R	R	R	R	R	R	R
Reset	-	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	LINADADR_ISME	LIN Addressing Success Flag Bit. In LIN auto-addressing mode, it is used to determine whether the device is addressed successfully. 0: The addressing fails. 1: The addressing succeeds.

[5]	ERRBIT	Bit Error/Format Error. After LIN read-back feature is enabled, this bit is set to “1” when the logic level of LIN TXD pin is different from LIN bus level, or a format error occurs in the received LIN bus message. 0: No bit error/format error occurs. 1: A bit error/format error occurs.
[4]	ERRSYNC	Sync Error (synchronization timeout or synchronization overhead). This bit is cleared to “0” by hardware when a new frame arrives or when LIN[CLRERR] is cleared to “0”. 0: No sync error occurs. 1: A sync error occurs.
[3]	ERRCHK	Data Check Error. This bit is cleared to “0” by hardware when a new frame arrives or when LIN_CSR[CLRERR] is cleared to “0”. 0: No data check error occurs. 1: A data check error occurs.
[2]	ERRPRTY	ID Check Error. This bit is cleared to “0” by hardware when a new frame arrives or when LIN_CSR[CLRERR] is cleared to “0”. 0: No ID check error occurs. 1: An ID check error occurs.
[1]	ABORT	Transmission Completion Interrupt Flag This bit is set to “1” after the slave receives or sends the data, and cleared to “0” when a new frame arrives or software writes “0” to LIN_SR[LINDONE]. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	LINACT	LIN Bus Active Flag 0: No data is transmitting on LIN bus 1: Data is transmitting on LIN bus

12.8.3 LIN_CSR (0xBB)

Bit	7	6	5	4	3	2	1	0
Name	LINADADR_CUR	RDBAKDIS	LINSLP	LINADADR	TXWAKEUP	LINACK	LINSTOP	LINEN
Type	R/W	R/W	R/W	R/W	W1	W1	W1	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	LINADADR_CUR	LIN Auto-addressing Current Source Enable. This bit is configured only after LIN_CSR[LINADADR] is enabled. 0: Disable 1: Enable
[6]	RDBAKDIS	LIN Read-Back Enable 0: Enable 1: Disable
[5]	LINSLP	LIN Sleep Mode Enable Read: 0: LIN is active 1: LIN is in sleep mode Write: 0: LIN exits sleep mode 1: LIN enters sleep mode
[4]	LINADADR	LIN Auto-addressing Enable Bit. When it is enabled, pull-up resistors are automatically switched off during synch break field. 0: Disable 1: Enable

[3]	TXWAKEUP	LIN Wake-up Write: 0: No effect 1: Send a wake-up signal
[2]	LINACK	This bit is used to send an ACK for the frame header. The checksum, R/W mode, data content and length must be configured before this bit is set to “1”. 0: No effect 1: Send an ACK for the frame header
[1]	LINSTOP	After this bit is set to “1”, LIN stops sending or receiving the data and waits for a new frame header. LIN_SR[ABORT] is set to “1” as well. 0: No effect 1: LIN stops processing the current frame and waits for a new frame header
[0]	LINEN	LIN Enable 0: Disable 1: Enable

12.8.4 LIN_ID (0xBC)

Bit	7	6	5	4	3	2	1	0
Name	RSV		LIN_ID					
Type	-	-	R	R	R	R	R	R
Reset	-	-	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	LIN_ID	ID received by LIN

12.8.5 LIN_SIZE (0xBD)

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

Bit	Name	Description
[7:4]	RSV	Reserved
[3:0]	LIN_SIZE	Frame length of data received/transmitted

12.8.6 LIN_BAUD (0xBF,0xBE)

LIN_BAUDH(0xBF)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	LIN_BAUD[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	1	1
LIN_BAUDL(0xBE)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LIN_BAUD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	0	1	1	1

Bit	Name	Description
[15:0]	LIN_BAUD	Baud Rate Baud rate = SYSCLK/(LIN_BAUD + 1)

13 MDU

13.1 MDU Introduction

MDU 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 PID operation. MDU module can be invoked in different interrupt services and master programs, and the results are independent from each other.

13.2 MDU Features

The MDU 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
  - Low-pass filter (LPF)
  - Coordinate transformation (SIN/COS)
  - Arctangent (ATAN)
  - PI/PID

13.3 MDU Instructions

13.3.1 MDU Operations

MDU is operated as follows.

1. Configure MDU_CR[MDUMOD] register to select computing mode of the MDU module;
2. Write the data to the associated computing units, and configure MDU_CR[MDUSTA] to select computing unit of the MDU module, and start MDU computing;
3. Wait for MDU_CR[MDUBUSY] to be cleared to “0” by hardware.

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

13.3.2 16-bit Signed Multiplication with the Result Shifted Left by One-bit

When MDU_CR[MDUMOD] = 000, MDU 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 Result Shifted Left by One-bit

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

13.3.3 16-bit Signed Multiplication

When MDU_CR[MDUMOD] = 001, MDU module works in the 16-bit signed multiplication mode. As shown in Table 13-2, 31-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

13.3.4 16-bit Unsigned Multiplication

When MDU_CR[MDUMOD] = 010, MDU 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

13.3.5 32-bit/16-bit Unsigned Division

When MDU_CR[MDUMOD] = 011, MDU 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 are obtained after 32-bit dividend and a 16-bit 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	-
DIVx_DQ	-	Quotient
DIVx_DR	-	Remainder

13.3.6 LPF

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

The calculation formula of LPF is:

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

Wherein,

Y_k: Filtered data

Y_{k-1}: Previous filtered output

K: Filter coefficient

X_k: Value to be filtered

As shown in Table 13-5, Y_k and Y_{k-1} are 32-bit signed data, X_k and K are 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

13.3.7 Coordinate Transformation

When MDU_CR[MDUMOD] = 100, MDU 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:

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

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

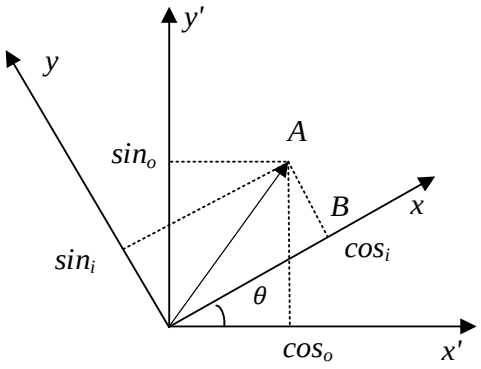


Figure 13-1 Coordinate Transformation

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

13.3.8 Arctangent

When MDU_CR[MDUMOD] = 101, MDU 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:

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

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

Wherein,

$Usin\theta$: 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

13.3.9 PI/PID

13.3.9.1 PI/PID Introduction

PI/PID 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:

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

PID algorithm:

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

Wherein,

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, K_i and K_d: Proportional (P), integral (I) and differential (D) coefficients of regulator

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

13.3.9.2 PI/PID 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”.

13.3.9.3 PI/PID Operations

1. Initialize MDU before the operations, and configure Kp, Ki, Kd and the maximum and minimum values of U_k;
2. Set MDU_CR[MDUMOD] to 111, and then select Comp_Unit0 and Comp_Unit1 as the PI Mode, and Comp_Unit2 and Comp_Unit3 as the PID Mode. Later, configure MDU_CR[MDUSTA] bit to select the desired computing unit and start PI/PID 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_KD = KD;                    //Initialize Kd
PIx_UKMAX = UKMAX;              //Initialize maximum output
PIx_UKMIN = UKMIN;              //Initialize minimum output
PIx_EK1 = X;                    //Initialize Ek-1
PIx_UKH = Y1;                   //Initialize 16 high-order bits of Uk-1
PIx_UKL = Y2;                   //Initialize 16 low-order bits of Uk-1
```

13.4 MDU Registers

13.4.1 MDU_CR (0xC1)

Bit	7	6	5	4	3	2	1	0
Name	MDUBUSY	MDUSTA				MDUMOD		
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]	MDUBUSY	MDU Busy Flag A write of MDU_CR[6:3] starts MDU module. This bit is set to “1” after MDU completes operations.
[6:3]	MDUSTA	This bit is used to configure computing unit of the MDU. Four options are available. MDU module starts operation after initiating the computing. 0001: Comp_Unit0 is activated 0010: Comp_Unit1 is activated 0100: Comp_Unit2 is activated 1000: Comp_Unit3 is activated
[2:0]	MDUMOD	MDU Mode Selection 000: 16-Bit Signed Multiplication (the result shifted left by one-bit) 001: 16-Bit Signed Multiplication 010: 16-Bit Unsigned Multiplication 011: 32-Bit/16-Bit Unsigned Division 100: Coordinate Transformation (SIN/COS) 101: ATAN 110: LPF 111: PI/PID, which decides the computing unit. Computing unit 0 and 1 are selected in PI mode, and computing unit 2 and 3 in PID mode.

13.4.2 MUL0_MA (0x0FA0, 0x0FA1)

MUL0_MAH (0x0FA0)								
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 (0x0FA1)								
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

13.4.3 MUL0_MB (0x0FA2, 0x0FA3)

MUL0_MBH (0x0FA2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	MUL0_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
MUL0_MBL (0x0FA3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MUL0_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]	MUL0_MB	Data register B of MUL0; Multiplicand of the multiplication

13.4.4 MUL0_MC (0x0FA4, 0x0FA5, 0x0FA6, 0x0FA7)

MUL0_MCHH (0x0FA4)								
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 (0x0FA5)								
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 (0x0FA6)								
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 (0x0FA7)								
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 is held by MUL0_MCH and the 16 low-order bits by MUL0_MCL.

13.4.5 MUL1_MA (0x0F98, 0x0F99)

MUL1_MAH (0x0F98)								
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 (0x0F99)								
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

13.4.6 MUL1_MB (0x0F9A, 0x0F9B)

MUL1_MBH (0x0F9A)								
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 (0x0F9B)								
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

13.4.7 MUL1_MC (0x0F9C, 0x0F9D, 0x0F9E, 0x0F9F)

MUL1_MCHH (0x0F9C)								
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 (0x0F9D)								
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 (0x0F9E)								
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 (0x0F9F)								
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 is held by MUL1_MCH and the 16 low-order bits by MUL1_MCL.

13.4.8 MUL2_MA (0x0F40, 0x0F41)

MUL2_MAH (0x0F40)								
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 (0x0F41)								
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

13.4.9 MUL2_MB (0x0F42, 0x0F43)

MUL2_MBH (0x0F42)								
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 (0x0F43)								
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

13.4.10 MUL2_MC (0x0F44, 0x0F45, 0x0F46, 0x0F47)

MUL2_MCHH (0x0F44)								
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 (0x0F45)								
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 (0x0F46)								
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 (0x0F47)								
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 MUL2. The 16 high-order bits of the data is held by MUL2_MCH and the 16 low-order bits by MUL2_MCL.

13.4.11 MUL3_MA (0x0F38, 0x0F39)

MUL3_MAH (0x0F38)								
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 (0x0F39)								
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

13.4.12 MUL3_MB (0x0F3A, 0x0F3B)

MUL3_MBH (0x0F3A)								
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 (0x0F3B)								
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

13.4.13 MUL3_MC (0x0F3C, 0x0F9D, 0x0F3E, 0x0f3F)

MUL3_MCHH(0x0F3C)								
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(0x0F3D)								
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(0x0F3E)								
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(0x0F3F)								
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 is held by MUL3_MCH and the 16 low-order bits by MUL3_MCL.

13.4.14 DIV0_DA (0x0F8C, 0x0F8D, 0x0F8E, 0x0F8F)

DIV0_DAHH (0x0F8C)								
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 (0x0F8D)								
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 (0x0F8E)								
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 (0x0F8F)								
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 is held by DIV0_DAH and the 16 low-order bits by DIV0_DAL.

13.4.15 DIV0_DB (0x0F90, 0x0F91)

DIV0_DBH (0x0F90)								
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 (0x0F91)								
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

13.4.16 DIV0_DQ (0x0F92, 0x0F93, 0x0F94, 0x0F95)

DIV0_DQHH (0x0F92)								
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 (0x0F93)								
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 (0x0F94)								
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 (0x0F95)								
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 is held by DIV0_DQH and the 16 low-order bits by DIV0_DQL.

13.4.17 DIV0_DR (0x0F96, 0x0F97)

DIV0_DRH (0x0F96)								
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 (0x0F97)								
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

13.4.18 DIV1_DA (0x0F80, 0x0F81, 0x0F82, 0x0F83)

DIV1_DAHH (0x0F80)								
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 (0x0F81)								
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 (0x0F82)								
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 (0x0F83)								
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 is held by DIV1_DA and the 16 low-order bits by DIV1_DA.

13.4.19 DIV1_DB (0x0F84, 0x0F85)

DIV1_DBH (0x0F84)								
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 (0x0F85)								
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

13.4.20 DIV1_DQ (0x0F86, 0x0F87, 0x0F88, 0x0F89)

DIV1_DQHH (0x0F86)								
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 (0x0F87)								
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 (0x0F88)								
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 (0x0F89)								
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 is held by DIV1_DQH and the 16 low-order bits by DIV1_DQL.

13.4.21 DIV1_DR (0x0F8A, 0x0F8B)

DIV1_DRH (0x0F8A)								
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 (0x0F8B)								
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

13.4.22 DIV2_DA (0x0F2C, 0x0F2D, 0x0F2E, 0x0F2F)

DIV2_DAHH (0x0F2C)								
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 (0x0F2D)								
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 (0x0F2E)								
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 (0x0F2F)								
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 is held by DIV2_DA and the 16 low-order bits by DIV2_DA.

13.4.23 DIV2_DB (0x0F30, 0x0F31)

DIV2_DBH (0x0F30)								
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 (0x0F31)								
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

13.4.24 DIV2_DQ (0x0F32, 0x0F33, 0x0F34, 0x0F35)

DIV2_DQHH (0x0F32)								
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 (0x0F33)								
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 (0x0F34)								
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 (0x0F35)								
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 is held by DIV2_DQH and the 16 low-order bits by DIV2_DQL.

13.4.25 DIV2_DR (0x0F36, 0x0F37)

DIV2_DRH (0x0F36)								
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 (0x0F37)								
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

13.4.26 DIV3_DA (0x0F20, 0x0F21, 0x0F22, 0x0F23)

DIV3_DAHH (0x0F20)								
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 (0x0F21)								
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 (0x0F22)								
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 (0x0F23)								
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 is held by DIV3_DAH and the 16 low-order bits by DIV3_DAL.

13.4.27 DIV3_DB (0x0F24, 0x0F25)

DIV3_DBH (0x0F24)								
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 (0x0F25)								
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

13.4.28 DIV3_DQ (0x0F26, 0x0F27, 0x0F28, 0x0F29)

DIV3_DQHH (0x0F26)								
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 (0x0F27)								
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 (0x0F28)								
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 (0x0F29)								
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 is held by DIV3_DQH and the 16 low-order bits by DIV3_DQL.

13.4.29 DIV3_DR (0x0F2A, 0x0F2B)

DIV3_DRH (0x0F2A)								
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 (0x0F2B)								
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

13.4.30 SCAT0_COS (0x0F16, 0x0F17)

SCAT0_COSH (0x0F16)								
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 (0x0F17)								
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	Input COS of SIN/COS and ATAN modes of Computing Unit SCAT0

13.4.31 SCAT0_SIN (0x0F18, 0x0F19)

SCAT0_SINH (0x0F18)								
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 (0x0F19)								
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	Input SIN of SIN/COS and ATAN modes of Computing Unit SCAT0

13.4.32 SCAT0_THE (0x0F1A, 0x0F1B)

SCAT0_THEH 0x0F1A								
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 (0x0F1B)								
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	Input THE of SIN/COS mode of Computing Unit SCAT0

13.4.33 SCAT0_RES1 (0x0F1C, 0x0F1D)

SCAT0_RES1H (0x0F1C)								
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 (0x0F1D)								
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	Output COS of SIN/COS mode and output U of ATAN mode of Computing Unit SCAT0

13.4.34 SCAT0_RES2 (0x0F1E, 0x0F1F)

SCAT0_RES2H (0x0F1E)								
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 (0x0F1F)								
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	Output SIN of SIN/COS mode and output θ of ATAN mode of Computing Unit SCAT0

13.4.35 SCAT1_COS (0x0F0C, 0x0F0D)

SCAT1_COSH (0x0F0C)								
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 (0x0F0D)								
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	Input COS of SIN/COS and ATAN modes of Computing Unit SCAT1

13.4.36 SCAT1_SIN (0x0F0E, 0x0F0F)

SCAT1_SINH (0x0F0E)								
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 (0x0F0F)								
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	Input SIN of SIN/COS and ATAN modes of Computing Unit SCAT1

13.4.37 SCAT1_THE (0x0F10, 0x0F11)

SCAT1_THEH (0x0F10)								
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 (0x0F11)								
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	Input THE of SIN/COS mode of Computing Unit SCAT1

13.4.38 SCAT1_RES1 (0x0F12, 0x0F13)

SCAT1_RES1H (0x0F12)								
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 (0x0F13)								
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	Output COS of SIN/COS mode and output U of ATAN mode of Computing Unit SCAT1

13.4.39 SCAT1_RES2 (0x0F14, 0x0F15)

SCAT1_RES2H (0x0F14)								
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 (0x0F15)								
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	Output SIN of SIN/COS mode and output θ of ATAN mode of Computing Unit SCAT1

13.4.40 SCAT2_COS (0x0F02, 0x0F03)

SCAT2_COSH (0x0F02)								
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 (0x0F03)								
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	Input COS of SIN/COS and ATAN modes of Computing Unit SCAT2

13.4.41 SCAT2_SIN (0x0F04, 0x0F05)

SCAT2_SINH (0x0F04)								
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 (0x0F05)								
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 the SIN/COS and ATAN modes of Computing Unit SCAT2

13.4.42 SCAT2_THE (0x0F06, 0x0F07)

SCAT2_THEH (0x0F06)								
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 (0x0F07)								
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	Input THE of SIN/COS mode of Computing Unit SCAT2

13.4.43 SCAT2_RES1 (0x0F08, 0x0F09)

SCAT2_RES1H (0x0F08)								
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 (0x0F09)								
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	Output COS of SIN/COS mode and output U of ATAN mode of Computing Unit SCAT2

13.4.44 SCAT2_RES2 (0x0F0A, 0x0F0B)

SCAT2_RES2H (0x0F0A)								
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 (0x0F0B)								
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	Output SIN of SIN/COS mode and output θ of ATAN mode of Computing Unit SCAT2

13.4.45 SCAT3_COS (0x0EF8, 0x0EF9)

SCAT3_COSH (0x0EF8)								
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 (0x0EF9)								
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	Input COS of SIN/COS and ATAN modes of Computing Unit SCAT3

13.4.46 SCAT3_SIN (0x0EFA, 0x0EFB)

SCAT3_SINH (0x0EFA)								
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 (0x0EFB)								
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	Input SIN of SIN/COS and ATAN modes of Computing Unit SCAT3

13.4.47 SCAT3_THE (0x0EFC, 0x0EFD)

SCAT3_THEH (0x0EFC)								
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 (0x0EFD)								
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	Input THE of SIN/COS mode of Computing Unit SCAT3

13.4.48 SCAT3_RES1 (0x0EFE, 0x0EFF)

SCAT3_RES1H (0x0EFE)								
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 (0x0EFF)								
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	Output COS of SIN/COS mode and output U of ATAN mode of Computing Unit SCAT3

13.4.49 SCAT3_RES2 (0x0F00, 0x0F01)

SCAT3_RES2H (0x0F00)								
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 (0x0F01)								
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	Output SIN of SIN/COS mode and output θ of ATAN mode of Computing Unit SCAT3

13.4.50 LPF0_K (0x0FD0, 0x0FD1)

LPF0_KH (0x0FD0)								
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 (0x0FD1)								
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]	Input K of LPF0

13.4.51 LPF0_X (0x0FD2, 0x0FD3)

LPF0_XH (0x0FD2)								
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 (0x0FD3)								
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]	Input X of LPF0						

13.4.52 LPF0_Y (0x0FD4, 0x0FD5, 0x0FD6, 0x0FD7)

LPF0_YHH (0x0FD4)								
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 (0x0FD5)								
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 (0x0FD6)								
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 (0x0FD7)								
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]	LPF0 unit input and output Input: LPF0_Y_{k-1} Output: LPF0_Y_k						

13.4.53 LPF1_K (0x0FC8, 0x0FC9)

LPF1_KH (0x0FC8)								
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 (0x0FC9)								
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]	Input K of LPF1						

13.4.54 LPF1_X (0x0FCA, 0x0FCB)

LPF1_XH (0x0FCA)								
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 (0x0FCB)								
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]	Input X of LPF1

13.4.55 LPF1_Y (0x0FCC, 0x0FCD, 0x0FCE, 0x0FCF)

LPF1_YHH (0x0FCC)								
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 (0x0FCD)								
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 (0x0FCE)								
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 (0x0FCF)								
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]	LPF1 unit input and output Input: LPF1_Y_{k-1} Output: LPF1_Y_k

13.4.56 LPF2_K (0x0F78, 0x0F79)

LPF2_KH (0x0F78)								
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 (0x0F79)								
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]	Input K of LPF2

13.4.57 LPF2_X (0x0F7A, 0x0F7B)

LPF2_XH (0x0F7A)								
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 (0x0F7B)								
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]	Input X of LPF2

13.4.58 LPF2_Y (0x0F7C, 0x0F7D, 0x0F7E, 0x0F7F)

LPF2_YHH (0x0F7C)								
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 (0x0F7D)								
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 (0x0F7E)								
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 (0x0F7F)								
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]	LPF2 unit input and output Input: LPF2_Y_{k-1} Output: LPF2_Y_k

13.4.59 LPF3_K (0x0F70, 0x0F71)

LPF3_KH (0x0F70)								
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 (0x0F71)								
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]	Input K of LPF3

13.4.60 LPF3_X (0x0F72, 0x0F73)

LPF3_XH (0x0F72)								
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 (0x0F73)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LPF3_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]	LPF3_K[15:0]	Input X of LPF3

13.4.61 LPF3_Y (0x0F74, 0x0F75, 0x0F76, 0x0F77)

LPF3_YHH (0x0F74)								
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 (0x0F75)								
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 (0x0F76)								
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 (0x0F77)								
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]	LPF3 unit input and output this Input: LPF3_Y_{k-1} Output: LPF3_Y_k

13.4.62 PI0_KP (0x0FB8, 0x0FB9)

PI0_KPH (0x0FB8)								
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 (0x0FB9)								
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 factor of PI0

13.4.63 PI0_EK1 (0x0FBA, 0x0FBB)

PI0_EK1H (0x0FBA)								
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 (0x0FBB)								
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

13.4.64 PI0_EK (0x0FBC, 0x0FBD)

PI0_EKH (0x0FBC)								
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 (0x0FBD)								
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

13.4.65 PI0_KI (0x0FBE, 0x0FBF)

PI0_KIH (0x0FBE)								
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 (0x0FBF)								
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 factor of PI0

13.4.66 PI0_UKH (0x0FC0, 0x0FC1)

PI0_UKHH (0x0FC0)								
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 (0x0FC1)								
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

13.4.67 PI0_UKL (0x0FC2, 0x0FC3)

PI0_UKLH (0x0FC2)								
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 (0x0FC3)								
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

13.4.68 PI0_UKMAX (0x0FC4, 0x0FC5)

PI0_UKMAXH (0x0FC4)								
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 (0x0FC5)								
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 allowed output of PI0

13.4.69 PI0_UKMIN (0x0FC6, 0x0FC7)

PI0_UKMINH (0x0FC6)								
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 (0x0FC7)								
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 allowed output value of PI0

13.4.70 PI1_KP (0x0FA8, 0x0FA9)

PI1_KPH (0x0FA8)								
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 (0x0FA9)								
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 factor of PI1

13.4.71 PI1_EK1 (0x0FAA, 0x0FAB)

PI1_EK1H (0x0FAA)								
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 (0x0FAB)								
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

13.4.72 PI1_EK (0x0FAC, 0x0FAD)

PI1_EKH (0x0FAC)								
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 (0x0FAD)								
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

13.4.73 PI1_KI (0x0FAE, 0x0FAF)

PI1_KIH (0x0FAE)								
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 (0x0FAF)								
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 factor of PI1

13.4.74 PI1_UKH (0x0FB0, 0x0FB1)

PI1_UKHH (0x0FB0)								
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 (0x0FB1)								
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

13.4.75 PI1_UKL (0x0FB2, 0x0FB3)

PI1_UKLH (0x0FB2)								
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 (0x0FB3)								
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

13.4.76 PI1_UKMAX (0x0FB4, 0x0FB5)

PI1_UKMAXH (0x0FB4)								
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 (0x0FB5)								
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 allowed output of PI1

13.4.77 PI1_UKMIN (0x0FB6, 0x0FB7)

PI1_UKMINH (0x0FB6)								
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 (0x0FB7)								
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 allowed output of PI1

13.4.78 PI2_KP (0x0F5C, 0x0F5D)

PI2_KPH (0x0F5C)								
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 (0x0F5D)								
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 factor of PI2

13.4.79 PI2_EK1 (0x0F5E, 0x0F5F)

PI2_EK1H (0x0F5E)								
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 (0x0F5F)								
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

13.4.80 PI2_EK (0x0F60, 0x0F61)

PI2_EKH (0x0F60)								
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 (0x0F61)								
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

13.4.81 PI2_KI (0x0F62, 0x0F63)

PI2_KIH (0x0F62)								
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 (0x0F63)								
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 factor of PI2

13.4.82 PI2_UKH (0x0F64, 0x0F65)

PI2_UKHH (0x0F64)								
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 (0x0F65)								
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

13.4.83 PI2_UKL (0x0F66, 0x0F67)

PI2_UKLH (0x0F66)								
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 (0x0F67)								
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

13.4.84 PI2_UKMAX (0x0F68, 0x0F69)

PI2_UKMAXH (0x0F68)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKMAXL (0x0F69)								
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_UKMAX	Maximum allowed output of PI2

13.4.85 PI2_UKMIN (0x0F6A, 0x0F6B)

PI2_UKMINH (0x0F6A)								
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_UKMINL (0x0F6B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.86 PI2_KD (0x0F6C, 0x0F6D)

PI2_KDH (0x0F6C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_KD[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KDL (0x0F6D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI2_KD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.87 PI2_EK2 (0x0F6E, 0x0F6F)

PI2_EK2H (0x0F6E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI2_EK2[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EK2L (0x0F6F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Name	PI2_EK2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	PI2_EK2	Deviation before previous input of PI2

13.4.88 PI3_KP (0x0F48, 0x0F49)

PI3_KPH (0x0F48)								
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 (0x0F49)								
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

13.4.89 PI3_EK1 (0x0F4A, 0x0F4B)

PI3_EK1H (0x0F4A)								
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 (0x0F4B)								
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 coefficient deviation of PI3

13.4.90 PI3_EK (0x0F4C, 0x0F4D)

PI3_EKH (0x0F4C)								
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 (0x0F4D)								
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

13.4.91 PI3_KI (0x0F4E, 0x0F4F)

PI3_KIH (0x0F4E)								
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 (0x0F4F)								
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

13.4.92 PI3_UKH (0x0F50, 0x0F51)

PI3_UKHH (0x0F50)								
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 (0x0F51)								
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

13.4.93 PI3_UKL (0x0F52, 0x0F53)

PI3_UKLH (0x0F52)								
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 (0x0F53)								
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

13.4.94 PI3_UKMAX (0x0F54, 0x0F55)

PI3_UKMAXH (0x0F54)								
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 (0x0F55)								
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

13.4.95 PI3_UKMIN (0x0F56, 0x0F57)

PI3_UKMINH (0x0F56)								
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 (0x0F57)								
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

13.4.96 PI3_KD (0x0F58, 0x0F59)

PI3_KDH (0x0F58)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_KD[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KDL (0x0F59)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_KD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.97 PI3_EK2 (0x0F5A, 0x0F5B)

PI3_EK2H (0x0F5A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	PI3_EK2[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EK2L (0x0F5B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PI3_EK2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	PI3_EK2	Deviation before previous input of PI3

14 FOC

14.1 FOC Overview

14.1.1 FOC Introduction

The FOC module is used in sensorless and sensored FOC motor drive applications and SVPWM-based motor control applications. When DRV_CR0[FOCEN] = 0, FOC module is inactivated, and the FOC clock stops. The relevant FOC registers are forced into the reset state and cannot be written.

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

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

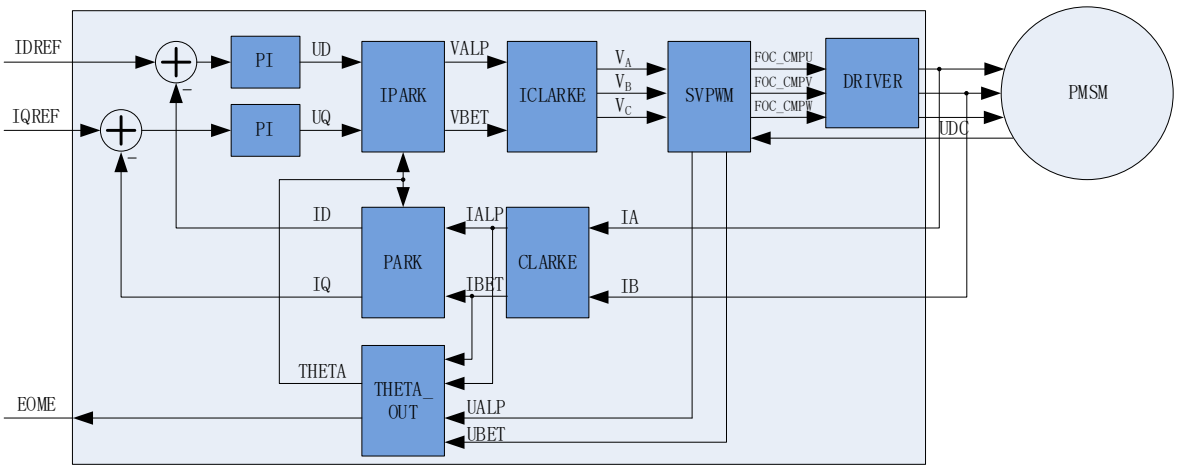


Figure 14-1 FOC Block Diagram

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

14.1.3 PI Controller

FOC module integrates two PI controllers:

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

14.1.4 Coordinate Transformation

14.1.4.1 Inverse Park Transformation

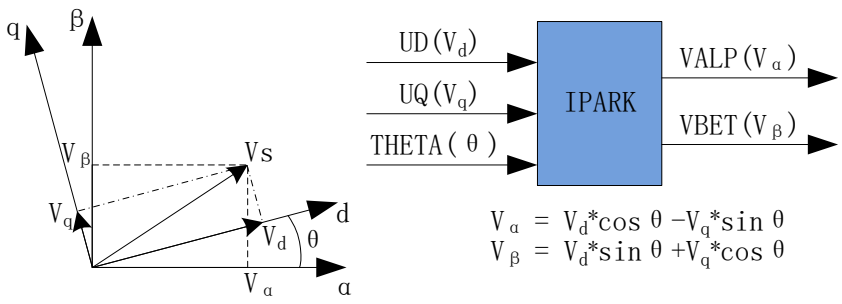


Figure 14-2 Inverse Park Transformation

Inverse Park transformation is used to transform two voltage vectors obtained by PI controller, FOC__UD and FOC__UQ, from d/q-axis coordinate to α/β-axis coordinate.

14.1.4.2 Inverse Clarke Transformation

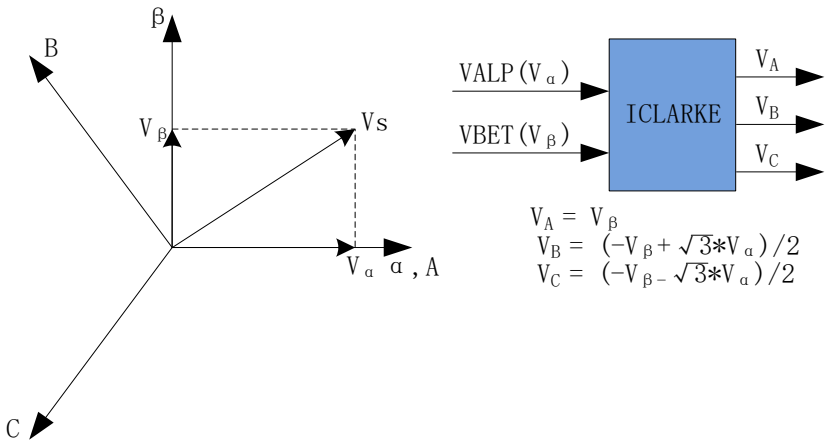


Figure 14-3 Inverse Clarke Transformation

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

14.1.4.3 Clarke Transformation

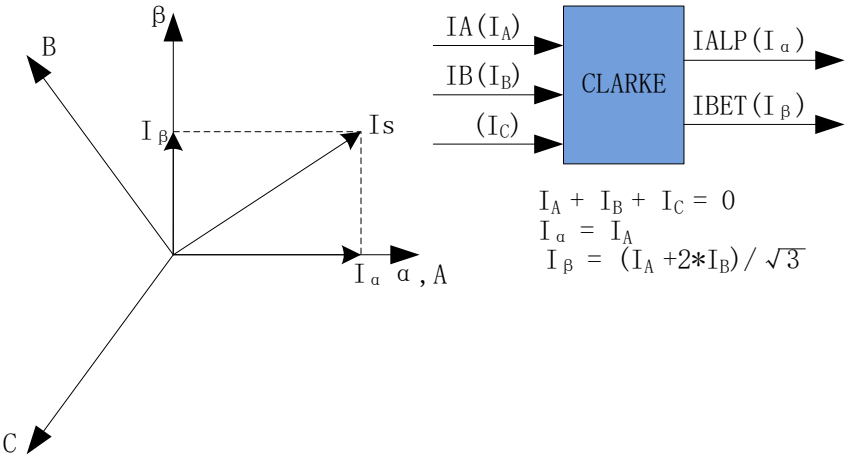


Figure 14-4 Clarke Transformation

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

14.1.4.4 Park Transformation

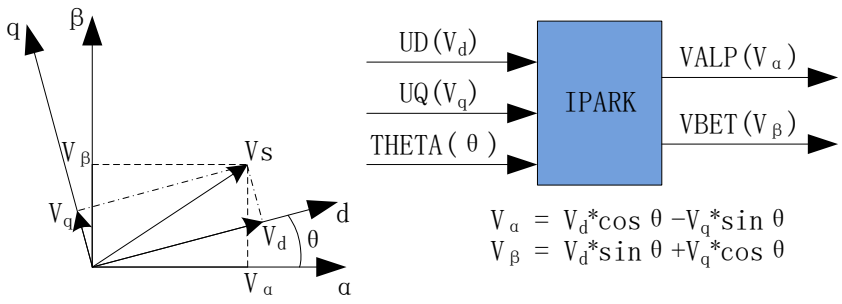


Figure 14-5 Park Transformation

Park transformation is used to transform the current vectors, obtained after Clarke transformation, from α/β-axis coordinate to d/q-axis coordinate to get the sampled d/q-axis current FOC__ID and FOC__IQ.

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

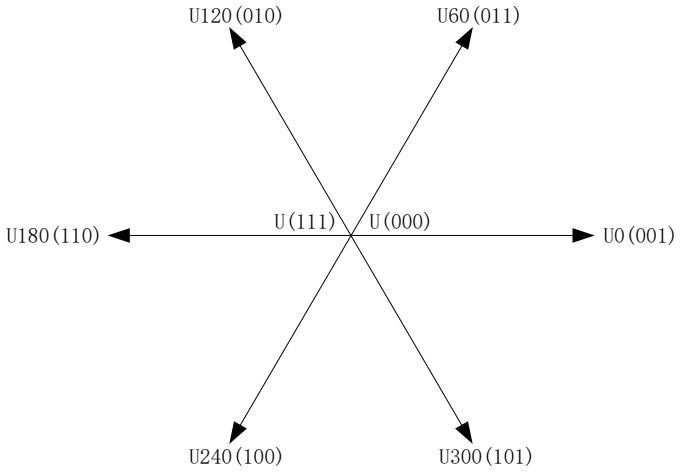


Figure 14-6 SVPWM Voltage Vector

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. Based on the principle of equal pulse, the effect, U0 applied 2*T1 time and U60 applied 2*T2 time, is equivalent to the U_{OUT}. The rest of time (T0) is applied by zero voltage vector.

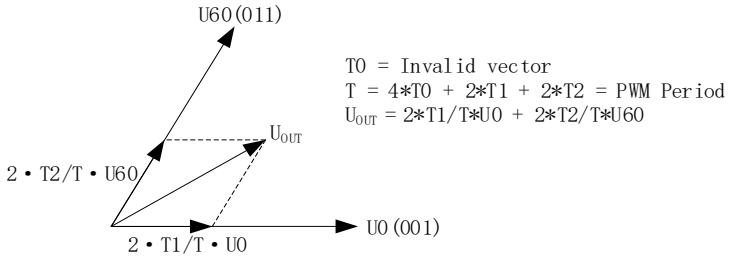


Figure 14-7 SVPWM Voltage Vector Synthesis

Table 14-1 States of SVPWM Inverter

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

14.1.5.1 Continuous SVPWM

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

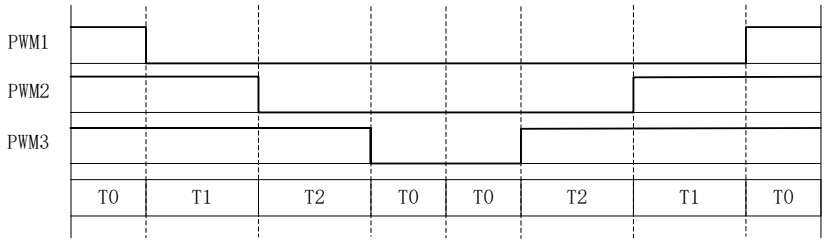


Figure 14-8 Output Level of Continuous SVPWM

14.1.5.2 Discontinuous SVPWM

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

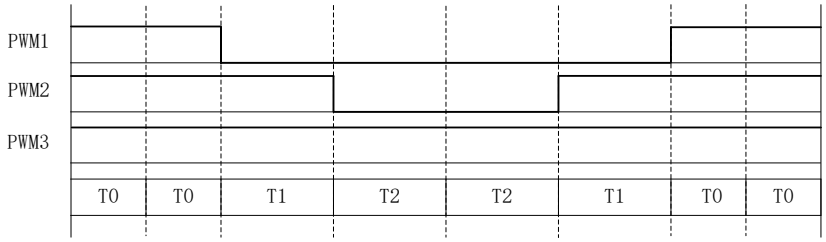


Figure 14-9 Output Level of Discontinuous SVPWM

14.1.6 Deadtime Compensation

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

14.1.7 Current and Voltage Sampling

In FOC mode, bus voltage and phase current are sampled by hardware automatically. The voltage division ratio of the bus voltage is 1/10, and the current sampling resistance is 0.1Ω. Before the FOC module operates, ADC and operational amplifier shall be enabled and the corresponding control registers be configured. No configuration is required for ADC channel and mode. Channel 2 can be selected for bus voltage sampling. 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 phase-A current (ia) and phase-B current (ib) respectively. In triple-shunt current sampling mode, ADC channel 0, channel 1 and channel 3 are the default sampling channels of ia, ib and phase-C current (ic) respectively.

14.1.7.1 Single-shunt Current Sampling Mode

FOC_CR1[CSM] is set to "00" to select the single-shunt current sampling mode. In this mode, FOC

module samples itrip (channel 4) twice during the DRV counter counting-up operation, and samples bus voltage during the DRV counter 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 moves the current sampling time, and this register shall be configured reasonably to ensure sampling is completed within T1' and T2'. For example, if FOC_TRGDLY = 5, the sampling time is delayed by 5*T = 208ns; and if FOC_TRGDLY = 0xFB(–5), the sampling time is advanced by 5*T = 208ns.

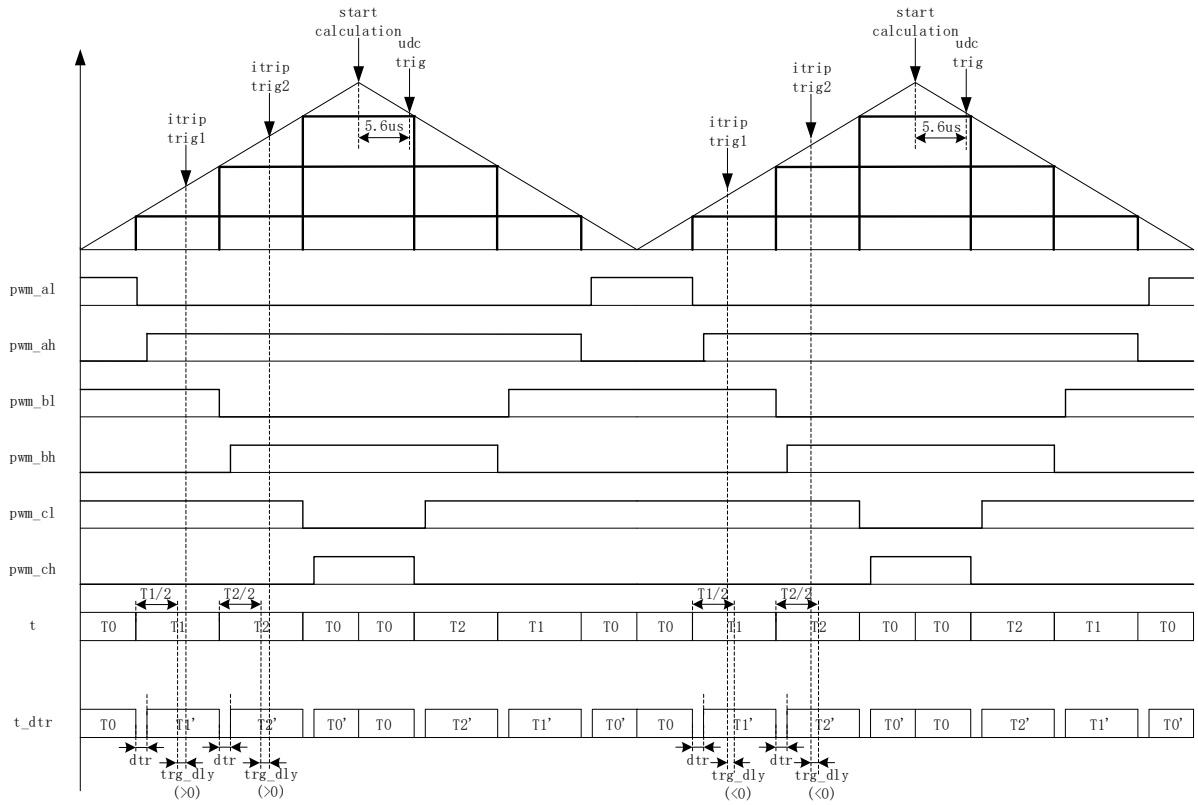


Figure 14-10 Single-shunt Current Sampling Timing

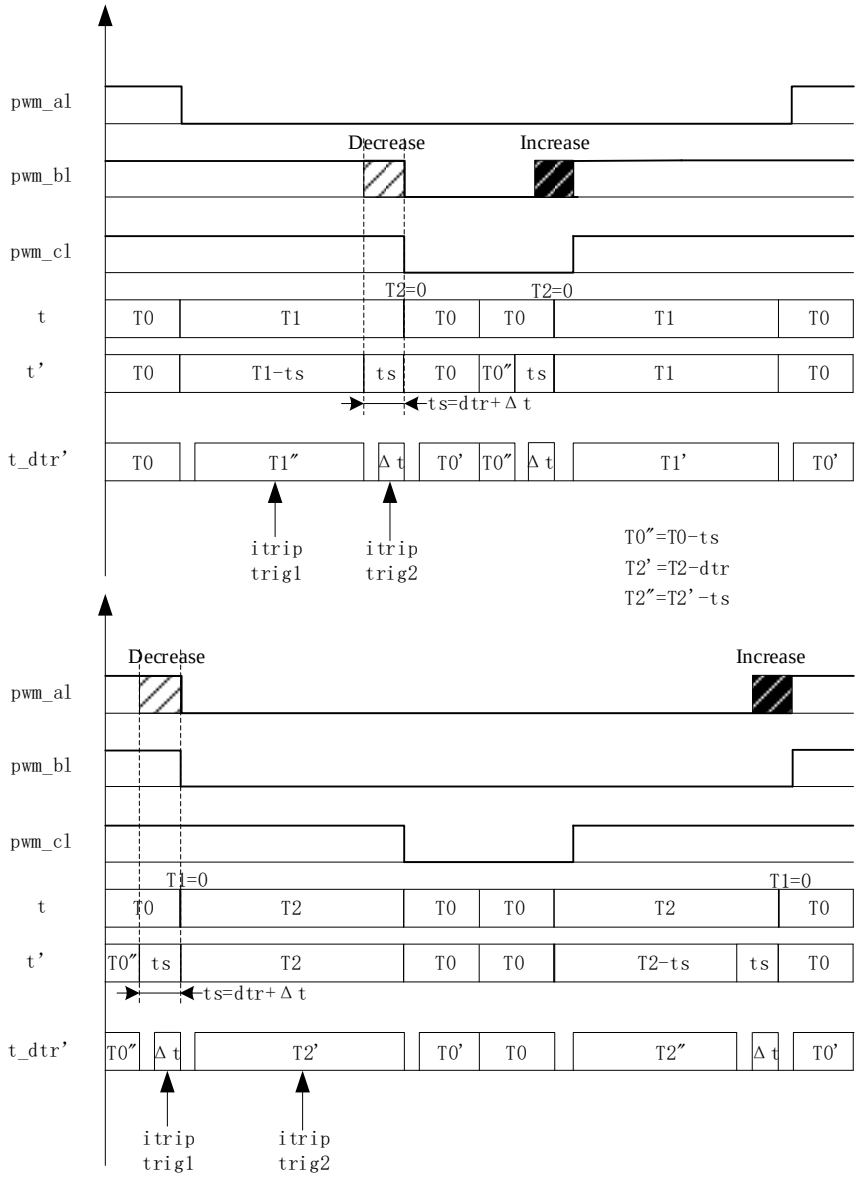


Figure 14-11 Single-shunt Current Sampling Time Compensation

The 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.

14.1.7.2 Dual/Triple-shunt Current Sampling Mode

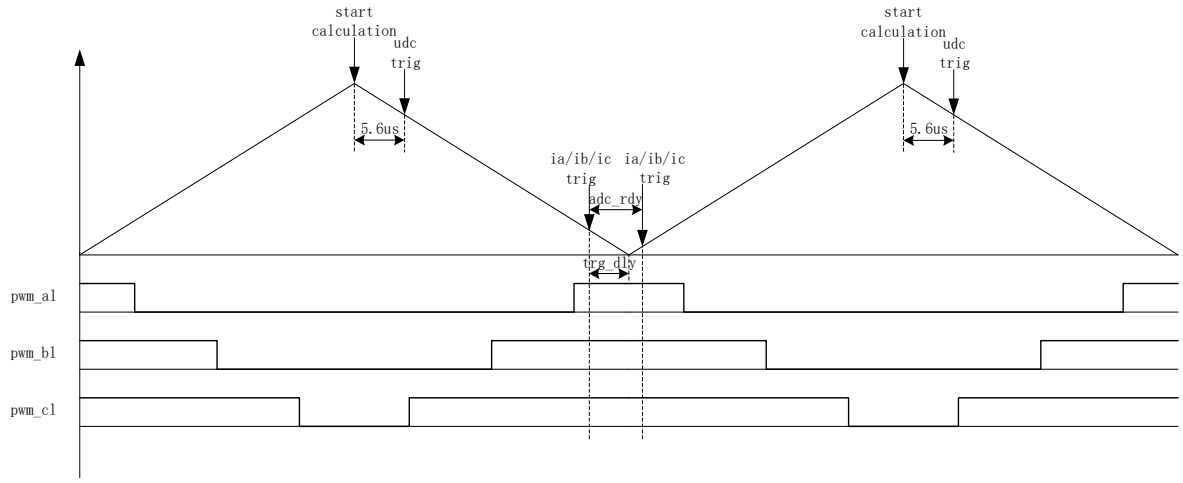


Figure 14-12 Dual/Triple-shunt Sequential Current Sampling Mode

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

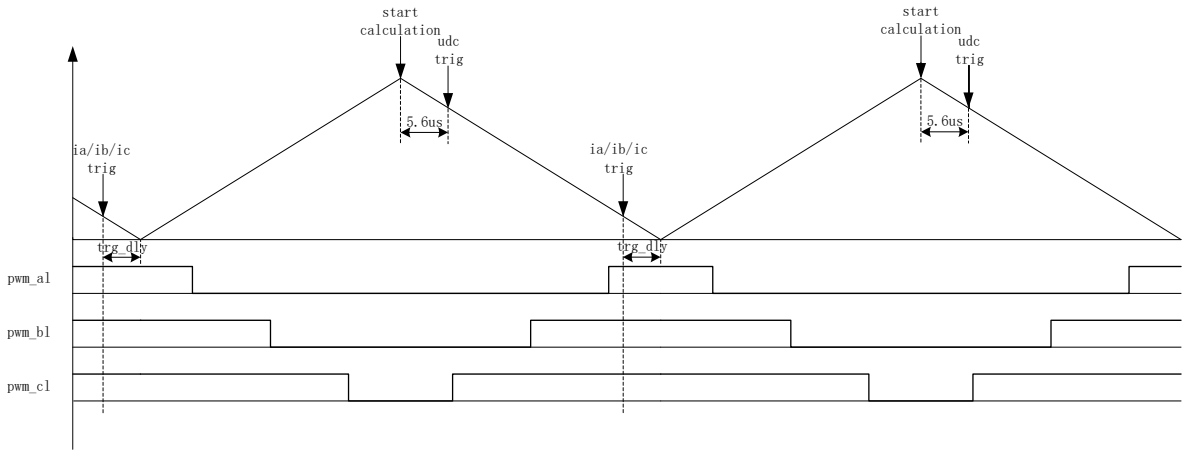


Figure 14-13 Dual/Triple-shunt Alternate Current Sampling Mode

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). TRG_DLY shall be configured reasonably to ensure sampling time for the ia/ib/ic current is within zero voltage vector (000). For example, when FOC_TRGDLY = 0xB2 and FOC counter counts down, phase current is sampled at 50*T = 2.08μs before an underflow event.

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

14.1.7.3 Current Sampling Offset

The current sampling offset voltage shall be added to sample full range of current due to the existence of the positive and negative phase current. When phase current is 0, ADC result is the offset value. ADC result minus this value, 0x4000 default, is the sampling current. Since ADC reference voltage and hardware are nonideal, there is a deviation between the default value and the real value. Therefore, it is necessary to calibrate the offset. The calibration procedure is as follows. When there is no current in three phases, MCU starts to sample the corresponding channel several times, averages all the sampled value, and write 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_CHC[CSOC] = 00/11, FOC_CSO is written to calibrate the offset of itrip and ic
- When FOC_CHC[CSOC] = 01, FOC_CSO is written to calibrate the offset of ia
- When FOC_CHC[CSOC] = 10, FOC_CSO is written to calibrate the offset of ib

14.1.8 Angle Mode

The sources of angle are as follows:

- Estimated angle of estimator
- Forced angle of estimator

14.1.8.1 Estimator Output Angle

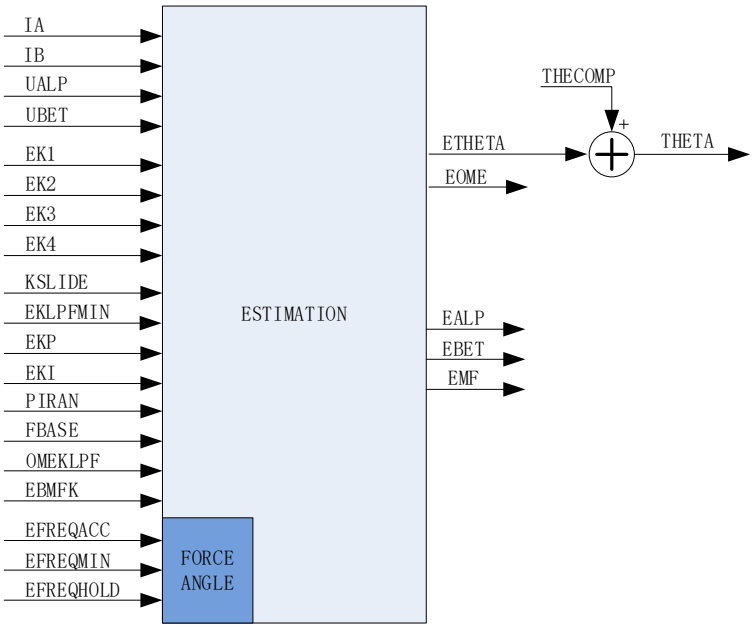


Figure 14-14 Schematic Block Diagram of Estimator

14.1.8.1.1 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 adaptive mode or SMO mode by configuring the FOC_CR3[MFP_EN] bit.

14.1.8.1.2 Forced Angle of Estimator

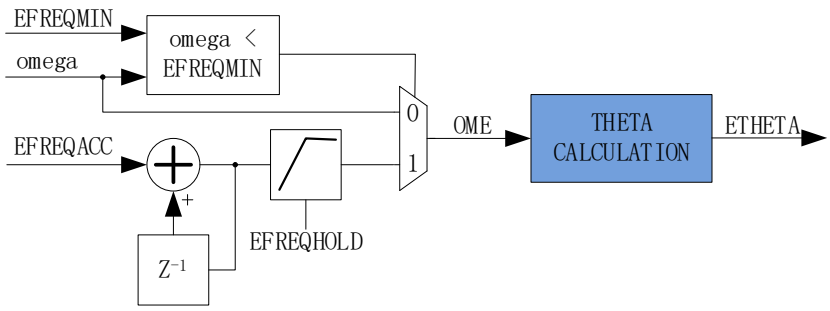


Figure 14-15 Schematic Diagram of Estimator Forced Angle

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

The forced angle feature of the estimator is enabled when FOC_CR1[RFAE] is set to "0" and FOC_CR1[EFAE] to "1". As shown in Figure 14-15, the estimator compares the value of real-time estimated speed (omega) and FOC_EFREQMIN to determine omega or forced speed (FOC__ETHETA) as the used speed (OME). When omega < FOC_EFREQMIN, the forced speed is selected as OME. The forced speed starts with 0 and increases by FOC_EFREQACC in each PWM cycle, with the maximum value FOC_EFREQHOLD. When omega ≥ FOC_EFREQMIN, omega is selected as OME.

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

14.1.8.1.3 Angle Compensation

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

14.1.9 Motor Real-time Parameters

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

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

14.1.9.1 Tailwind/Headwind Detection

FOC module provides tailwind/headwind detection feature. FOC module starts to operate when FOC_CR0[ESCMS] is set to "1", FOC_IDREF to "0" and FOC_IQREF to "0". Motor's rotor state is detected by FOC__ETHETA and FOC__EOME. If FOC__ETHETA decreases or FOC__EOME is a negative value, the motor rotates 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.

14.1.9.2 BEMF Detection

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

14.1.9.3 Motor Power

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

14.2 FOC Registers

14.2.1 FOC_CR0 (0x409F)

Bit	7	6	5	4	3	2	1	0
Name	OMIF	RSV	MERRS		RSV		ESCMS	EDIS
Type	R	-	R/W	R/W	-	-	R/W	R/W
Reset	0	-	0	0	-	-	0	0

Bit	Name	Description
[7]	OMIF	omega < FOC_ EFREQMIN Flag. This bit is valid even if FOC_ CR1 [EFAE] is 0. 0: omega ≥ FOC_EFREQMIN 1: omega < FOC_EFREQMIN
[6]	RSV	Reserved
[5:4]	MERRS	Maximum error of SMO Algorithm Selection 00: 0.5 01: 0.25 10: 0.125 11: 1.0
[3:2]	RSV	Reserved
[1]	ESCMS	Angle Mode Selection 0: Internal Test Mode 1: Recommended Mode
[0]	EDIS	FOC__EALP/FOC__EBET Auto-computation Disabled 0:Not forbid 1: Forbid

14.2.2 FOC_CR1 (0x40A0)

Bit	7	6	5	4	3	2	1	0
Name	STEP_EN	EFAE	VAB_SEL	RSV	CSM		RSV	
Type	R/W	R/W	R/W	-	R/W	R/W	-	-
Reset	0	0	0	-	0	0	-	-

Bit	Name	Description
[7]	STEP_EN	See descriptions on FOC_CR1 (0x40A0) in section Step Controller.
[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]	VAB_SEL	See descriptions on FOC_CR1 (0x40A0) in section Step Controller.
[4]	RSV	Reserved
[3:2]	CSM	Current Sampling Mode 00: Single-shunt Current Sampling Mode

		01: Dual-shunt Current Sampling Mode 10: Advanced Single-shunt Current Sampling Mode 11: Triple-shunt Current Sampling Mode Note: The Advanced Single-shunt Current Sampling Mode works better in low duty cycle output scenario than high duty cycle output scenario. You can switch between Single-shunt Current Sampling Mode and Advanced Single-shunt Current Sampling Mode according to the duty cycle, and modify FOC_TSMIN as required.
[1:0]	RSV	Reserved

14.2.3 FOC_CR2 (0x40A1)

Bit	7	6	5	4	3	2	1	0
Name	RSV	ICLR	F5SEG	DSS	CSOC		UQD	UDD
Type	-	R/W1	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]	ICLR	Clear FOC__IAMAX/FOC__IBMAX/FOC__ICMAX to “0” 0: No effect 1: This bit is automatically set to “0” after FOC__IAMAX/FOC__IBMAX/ FOC__ICMAX are cleared to “0”.
[5]	F5SEG	SVPWM Mode Selection 0: Continuous SVPWM 1: Discontinuous SVPWM (cannot be selected in single-shunt current sampling mode)
[4]	DSS	Dual/Triple-shunt Current Sampling Mode 0: Sequential Sampling Mode, where two-phase currents are sampled in each carrier period. 1: Alternate Sampling Mode. FOC module completes the calculation in every PWM cycle. Single-phase current is sampled in each PWM cycle, and two-phase currents 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 current sampling mode, “00” or “11” is written to calibrate itrip offset. In dual-shunt current sampling mode, “01” is written to calibrate ia offset and “10” to calibrate ib offset. In triple-shunt current sampling mode, “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: Enable 1: Disable
[0]	UDD	d-axis PI controller disabled, where the FOC__UD value is no longer updated by the PI controller. 0: Enable 1: Disable

14.2.4 FOC_CR3 (0x409E)

Bit	7	6	5	4	3	2	1	0
Name	EFAM	TAMD	MFP_EN	FOC_THECOMP_DIS	FOCFEN	MFP_MD	TSMINH9	TSMINH8
Type	R/W	R/W	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]	EFAM	OMEGA Startup Force Enable When FOC_CR1[EFAE] is set to “0” and FOC_CR1[EFAM] to “1”, FOC_OMEGA register is forced to stay as FOC_EFREQHOLD. 0: Disable 1: Enable
[6]	TAMD	Angle Calculation Method The angle derived from atan (ealpha/ebeta) is used as FOC_THETA. 0: Disable 1: Enable
[5]	MFP_EN	Adaptive Observer Enable 0: Disable 1: Enable
[4]	FOC_THECOMP_DIS	Algorithm without Compensation Angle Enable Bit. With this feature enabled, angle compensation of 26.5° is not executed even if the SMO or AO algorithm is selected. 0: Disable 1: Enable
[3]	FOCFEN	FOC Force Enable Bit When DRV_CR0[MESEL] is set to “1”, FOC module performs calculation even if DRV_CR0[OCS] = 0. 0: Disable 1: Enable
[2]	MFP_MD	Adaptive Observer Calculation Anti-overflow Enable 0: Disable 1: Enable. It can be applied in case of high speed (only 12 carrier cycles in one electrical cycle) and high current, but calculation accuracy is reduced.
[1:0]	TSMINH	Scale up by two bits of FOC_TSMIN, forming 10-bit data with the 0x40a2 register

14.2.5 FOC_CR4 (0x409D)

Bit	7	6	5	4	3	2	1	0
Name	RSV	DETH_EN	RSV		KSLIDE_SEL		UDQ_CPS_SEL	BLO_MD
Type	-	R/W	-	-	R/W	R/W	R/W	R/W
Reset	-	0	-	-	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	DETH_EN	Angle Difference Manual Input Enable With it enabled, angle difference cannot be automatically obtained by the estimator, and you have to enter the value manually. This feature is applied to some special algorithms for future extensions. 0: Disable 1: Enable
[5:4]	RSV	Reserved
[3:2]	KSLIDE_SEL	KSLIDE Selection 00: 0.85 01: 0.25 10: 0.5 11: 0.75

[1]	UDQ_CPS_SEL	FOC_UDQCPS Compensation Selection 0: UD is compensated 1: UQ is compensated
[0]	BLO_MD	Mode selection when the sampling window is not enough in triple-shunt current sampling mode 0: Mode 1 1: Mode 2

14.2.6 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 Current Sampling Mode: minimum window for sampling Dual/triple-shunt Current 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 cycle = 62.5μs, then FOC_TSMIN = (1 + 1)/62.5*4096 = 131. Advanced Single-shunt Current Sampling Mode: minimum window for sampling TSMIN = sampling window T_{window} + deadtime T_{DT} Example: Assuming that T_{window} = 1μs, T_{DT} = 1μs and TSMIN = 2μs, then FOC_TSMIN = (1 + 1)*24 = 48.

14.2.7 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 the driver must be longer than a certain time. After this bit is configured, high side of the driver 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 cycle = 62.5μs, then FOC_TGLI = (1 + 1)/62.5*4096 =131.

14.2.8 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 is turned on for less than FOC_TBLO, the current of this phase is not sampled but 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/50ns = 20.

14.2.9 FOC_TRGDLY (0x40A5)

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

Bit	Name	Description
[7:0]	FOC_TRGDLY	Time Configuration for Current Sampling When FOC_TRGDLY is set to "0", FOC module samples the current as follows. Single-shunt Current Sampling Mode: Midpoint between deadtime and applied time of active voltage vector Dual/Triple-shunt or Advanced Single-shunt Current Sampling Mode: Midpoint of vector 000 (Driver count value = 0) Range [-128, 127] 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 or Advanced Single-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 FOC_TRGDLY = 5 and Driver timer counts up, it samples the current at 5*T = 208ns after an overflow event occurs.

14.2.10 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)

14.2.11 FOC_EOMELPF (0x40AA, 0x40AB)

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

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

14.2.12 FOC__EMF (0x40AE, 0x40AF)

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

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

14.2.13 FOC_THECOMP (0x40AE, 0x40AF)

FOC_THECOMPH(0x40AE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_THECOMP[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_THECOMPL(0x40AF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_THECOMP[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_THECOMP	Angle Compensation Value The output angle FOC__THETA is derived from estimated angle of the estimator + compensation value; the format is same with that of FOC__THETA. Range [-32768, 32767]

14.2.14 FOC__OMEEST (0x40B0, 0x40B1)

FOC__OMEESTH(0x40B0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__OMEEST[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__OMEESTL(0x40B1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__OMEEST[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__OMEEST	FOC Calculated Speed of Estimator Range (0,32767)

14.2.15 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]

14.2.16 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]

14.2.17 FOC__ID (0x40BC, 0x40BD)

FOC__IDH(0x40BC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__ID[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

FOC__IDL(0x40BD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC__ID[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	FOC__ID	d-axis current from coordinate transformation Range [-32768, 32767]

14.2.18 FOC__IQ (0x40BE, 0x40BF)

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

Bit	**Name**	**Description**
[15:0]	FOC__IQ	q-axis current from coordinate transformation Range [-32768, 32767]

14.2.19 FOC__IBET (0x40C0, 0x40C1)

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

Bit	**Name**	**Description**
[15:0]	FOC__IBET	β-axis current from coordinate transformation. Range [-32768, 32767]

14.2.20 FOC__VBET (0x40C2, 0x40C3)

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

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

14.2.21 FOC_UDQCPS (0x40C2, 0x40C3)

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

Bit	Name	Description
[15:0]	FOC_UDQCPS	d-axis/q-axis Voltage Compensation Value The result of PI controller (FOC__UD/FOC__UQ) added to FOC_UDCPS is transferred to the next module, and then is compensated according to FOC_CR4[UDQ_CPS_SEL]. Rang e[-32768,32767]

14.2.22 FOC__VALP (0x40C4, 0x40C5)

FOC__VALPH(0x40C4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC__VALP[15:8]							
Type	W	W	W	W	W	W	W	W
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	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

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

14.2.23 FOC__IC (0x40C6, 0x40C7)

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

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

14.2.24 FOC__IB (0x40C8, 0x40C9)

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

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

14.2.25 FOC__IA (0x40CA, 0x40CB)

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

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

14.2.26 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 of 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°.

14.2.27 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. Write: Start Angle of Estimator Range [-32768, 32767]

14.2.28 FOC__EALP (0x40D0, 0x40D1)

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

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

14.2.29 FOC__EBET (0x40D2, 0x40D3)

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

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

14.2.30 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 of Estimator Range [-32768, 32767]

14.2.31 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]

14.2.32 FOC_EOMEKLPF (0x40D8)

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

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

14.2.33 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. A Phase Current Recorded maximum value of phase-A current; This value may be unreliable unless the motor rotates in a full electrical period. This maximum value will not be cleared to "0" automatically unless FOC_CR2[ICLR] is set to "1". Range [-32768, 32767]

14.2.34 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. B Phase Current Recorded maximum value of phase-B current. This value may be unreliable unless the motor rotates in a full electrical period. This value will not be cleared to "0" automatically unless FOC_CR2[ICLR] is set to "1". Range [-32768, 32767]

14.2.35 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. C Phase Current Recorded maximum value of phase-C current. This value may be unreliable unless the motor rotates in a full electrical period. This value will not be cleared to "0" automatically unless FOC_CR2[ICLR] is set to "1". Range [-32768, 32767]

14.2.36 FOC_DKP (0x4070, 0x4071)

FOC_DKPH(0x4070)								
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(0x4071)								
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]	FOC_DKP	KP of D-axis PI Controller Range (0,32767); MSB is always 0; Q12 format

14.2.37 FOC_EKP (0x4074, 0x4075)

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

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

14.2.38 FOC_EKI (0x4076, 0x4077)

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

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

14.2.39 FOC_DMAX (0x4078)

FOC_DMAX(0x4078)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DMAX[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_DMAX	Max. output of d-axis PI controller, corresponding to the high-order 8 bits of FOC__UD Range [-128,127]

14.2.40 FOC_DMIN (0x4079)

FOC_DMIN(0x4079)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_DMIN							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	FOC_DMIN	Min. output of d-axis PI controller, corresponding to the high-order 8 bits of FOC__UD Range [-256,255]

14.2.41 FOC_EKLPFMIN (0x407A, 0x407B)

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

Bit	Name	Description
[15:0]	FOC_EKLPFMIN/FOC_PLLKI	FOC_CR2[ESEL] = 0: The minimum value of BEMF low pass filter 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.

14.2.42 FOC_DKI (0x407C, 0x407D)

FOC_DKIH(0x407C)								
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(0x407D)								
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

14.2.43 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. MSB is always 0. Q15 format. Range [0, 32767]

14.2.44 FOC_FBASE (0x4080, 0x4081)

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

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

14.2.45 FOC_EFREQACC (0x4082, 0x4083)

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

Bit	Name	Description
[15:0]	FOC_EFREQACC	Speed Increment of the Forced Angle Mode. FOC_EFREQACC is an internal 24-bit variable and MSB is sign bit. Low-order 16 bits are written by software. Range [0, 65535] 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).

14.2.46 FOC_EFREQMIN (0x4084, 0x4085)

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

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

14.2.47 FOC_EFREQHOLD (0x4086, 0x4087)

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

Bit	Name	Description
[15:0]	FOC_EFREQHOLD	Maximum Value of Forced Speed of the Estimator FOC_EFREQHOLD is an internal 24-bit variable, and MSB is sign bit. High-order 16 bits are written by the software. Range [-32768, 32767] Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. If max. forced speed = 60rpm, then FOC_EFREQHOLD = 60rpm/speed_base*32768 = 655(0x028F).

14.2.48 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, and MSB is always 0. Q15 format. Range [0, 32767]

14.2.49 FOC_QMAX (0x408A)

FOC_QMAX(0x408A)								
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
[7:0]	FOC_QMAX	Max. output of q-axis PI controller, corresponding to the high-order 8 bits of FOC__UQ Range [-256,255]

14.2.50 FOC_QMIN (0x408B)

FOC_QMIN(0x408B)								
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
[7:0]	FOC_QMIN	Min. output of q-axis PI controller, corresponding to the high-order 8 bits of FOC__UQ Range [-32768, 32767]

14.2.51 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, and MSB is always 0. Q15 format. Range [0, 32767]

14.2.52 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, and MSB is always 0, Q15 format. Range [0, 32767]

14.2.53 FOC_IDREF (0x4090, 0x4091)

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

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

14.2.54 FOC_IQREF (0x4092, 0x4093)

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

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

14.2.55 FOC_QKP (0x4094, 0x4095)

FOC_QKPH(0x4094)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_QKP[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_QKPL(0x4095)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_QKP[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_QKP	KP coefficient of q-axis PI controller. MSB is always 0. Q12 format. Range [0,32767] corresponds to range of Q12 [0,8].

14.2.56 FOC_QKI (0x4096, 0x4097)

FOC_QKIH(0x4096)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	FOC_QKI[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_QKIL(0x4097)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_QKI[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_QKI	KI coefficient of q-axis PI controller. MSB is always 0. Q15 format. . The bit value range [0,32767] corresponds to the Q15 value range [0,1].

14.2.57 FOC__UDCFLT (0x4098, 0x4099)

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

Bit	**Name**	**Description**
[15:0]	FOC__UDCFLT	Filtered Bus Voltage FOC module samples the bus voltage and filters it to obtain FOC__UDCFLT. The voltage division ratio of the bus voltage is 1/10. ADC channel 2 can be selected. Range [0, 32767] Example: The bus voltage is scaled down by 1/10 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*10 = 30V.

15 Step Controller

15.1 Step Controller Introduction

15.1.1 Overview

The step controller supports SVPWM control for the two-phase step motor with three-phase voltage source inverter. When FOC_CR1[STEP_EN]=1, PWM output signals for a three-phase inverter are generated to drive the two-phase step motor. Topology structure of the step controller is shown in Figure 15-1.

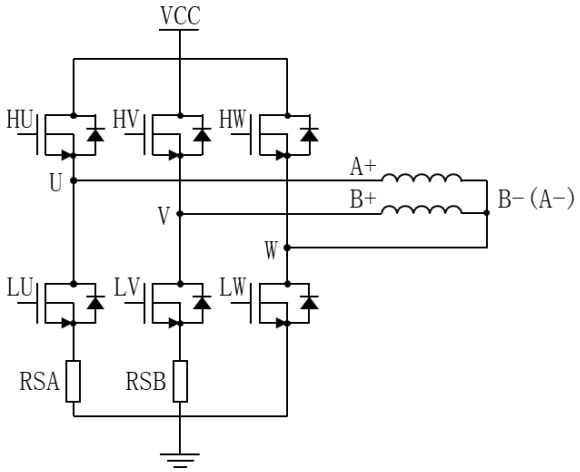


Figure 15-1 Two-phase Step Motor with Three-phase Voltage Source Inverter

Functional block diagram of the step controller is shown in Figure 15-2, including angle generator, PI controller, coordinate transform module, current sampling module and PWM output module, which achieves current closed-loop in hardware to improve efficiency.

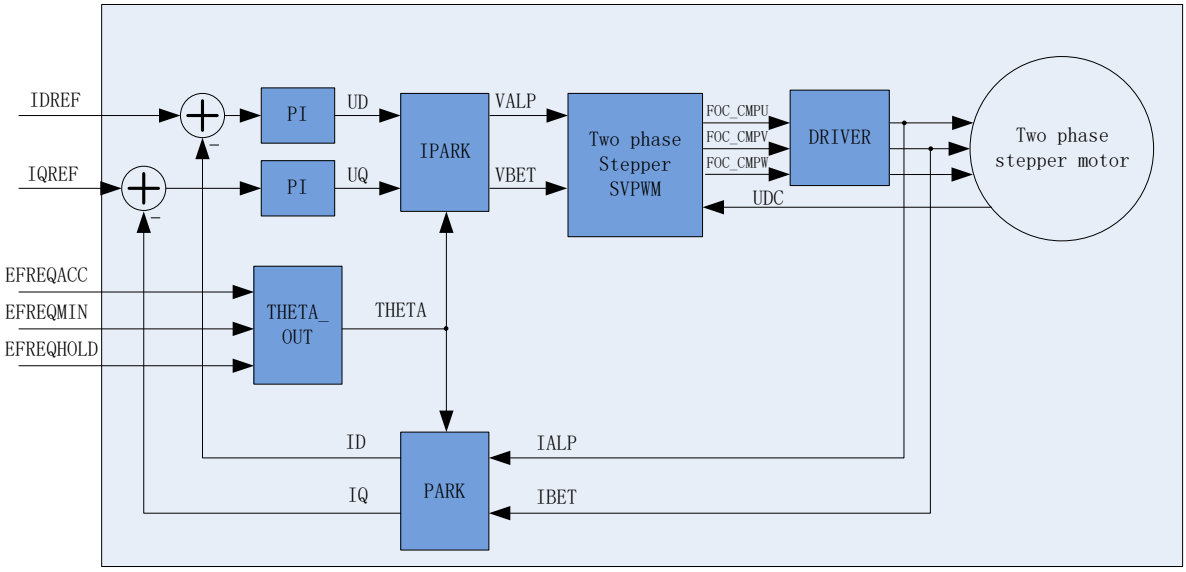


Figure 15-2 Functional Block Diagram of Step Controller

15.1.2 SVPWM for Two-phase Step Motor

The expected phase difference between the current in two windings of a two-phase step motor is 90°, so the input voltage phase of A and B phases shall be consistent with that of Alpha and Beta axis. The space voltage vectors of a two-phase step motor with three-phase voltage source inverter are distributed as shown in Figure 15-3. UALP (Alpha-axis voltage) and UBET (Beta-axis voltage) vary sinusoidally with the angle, and the action time of a basic space vector is calculated according to the sector where the vector exists. Thus, the output voltage vector UOUT is obtained. The direction of UOUT keeps consistent with the angle, and a quasi-circular rotating magnetic field is formed during the movement.

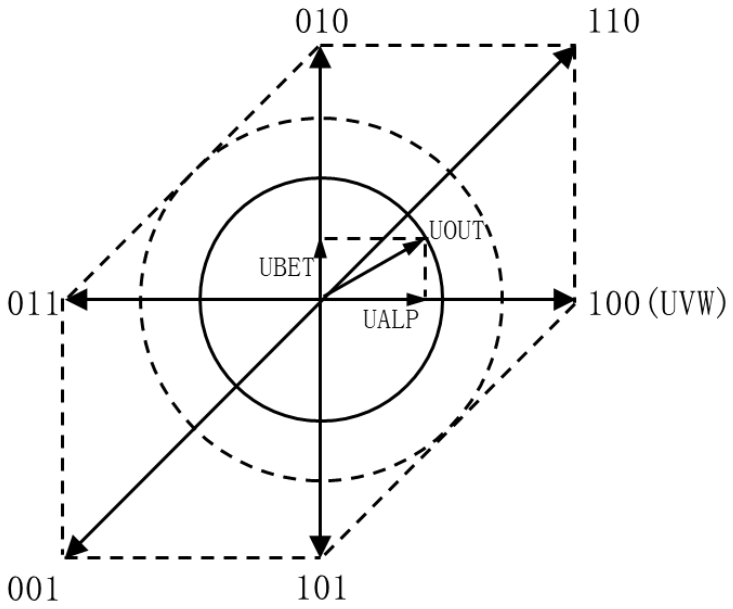


Figure 15-3 SVPWM Voltage Vectors

When |UOUT| > 0.707UDC, the output in sectors 3 and 6 reaches a point of saturation, and the action time of non-zero vectors t1 and t2 are greater than 1, that is, actual action time of the two vectors are greater than the carrier cycle. In this case, the action time has to be adjusted to ensure that SVPWM control is physically implemented in the system. The action time of non-zero vectors t1 and t2 are re-allocated as t1'=t1/(t1+t2) and t2'=t2/(t1+t2) respectively. Although it has been overmodulated, voltage amplitude in the sector has been saturated and cannot be increased, failing to form a quasi-circular rotating magnetic field.

15.1.3 Angle Mode

In forced pulling angle mode, FOC_CR3[EFAM] is set to "1" to write the target speed into FOC_EFREQHOLD, and the angle is added based on this speed during operation of the step motor.

Configuring FOC_CR1[EFAE] = 1 and then FOC_CR3[EFAM] = 1 enables the ramping feature during motor startup. If FOC_CR3[EFAM] is set to "1" first, the speed held by FOC_EFREQHOLD is directly implemented and the ramping feature is invalid. The speed increment in the ramping mode is written into

FOC_EFREQACC, and the target speed into FOC_EFREQMIN and FOC_EFREQHOLD. See chapter FOC for registers. After the ramping is completed, FOC_CR1[EFAE] is automatically set to “0”.

15.1.4 Deadtime Compensation

The deadtime compensation feature reduces voltage waveform distortion caused by deadtime in two-phase step mode. The step controller compensates the output comparison value according to current detection results, so that duty cycle of the phase voltage stay consistent with that of PWM wave.

15.1.5 Amplitude Compensation for Alpha-axis or Beta-axis Voltage

Alpha-axis or Beta-axis voltage can be compensated separately if current amplitudes of the two-phase step motor are inconsistent. FOC_CR1[VAB_SEL] determines the voltage to be compensated. If it is set to “0”, UALP (Alpha-axis voltage) is compensated; and if it is set to “1”, UBET (Beta-axis voltage) is compensated. ST_VABK decides the compensation ratio. If it is set to “0”, no compensation is performed. Otherwise, Ux' = Ux* ST_VABK / 256, where x refers to ALP or BET.

15.1.6 Phase Compensation for Beta-axis Voltage

This feature changes phase difference between Alpha-axis and Beta-axis voltages. It is enabled or disabled by configuring ST_BPCOMP. When ST_BPCOMP is set to “0”, no compensation is implemented and Alpha-axis voltage phase is 90 degrees ahead of Beta-axis voltage phase. Otherwise, Beta-axis voltage phase lags behind (90-ST_BPCOMP/32767*180) degrees.

15.1.7 Motor Real-time Parameters

The step controller monitors the state of motor using the following real-time variables provided by FOC module:

- Output angle FOC__THETA
- Estimated angle ST__ETHETA and estimated speed ST__OMEEST
- d-axis voltage FOC__UD and q-axis voltage FOC__UQ
- d-axis current FOC__ID and q-axis current FOC__IQ
- α-axis voltage FOC__VALP and β-axis voltage FOC__VBET
- Bus voltage FOC__UDCFLT
- Phase current FOC__IA and FOC__IB, and maximum phase current FOC__IAMAX and FOC__IBMAX
- α-axis BEMF FOC__EALP and β-axis BEMF FOC__EBET
- Magnitude of BEMF FOC__EMF
- Motor power FOC__POW

15.2 Step Controller Registers

15.2.1 FOC_CR1 (0x40A0)

Bit	7	6	5	4	3	2	1	0
Name	STEP_EN	EFAE	VAB_SEL	RSV	CSM		RSV	
Type	R/W	R/W	R/W	-	R/W	R/W	-	-
Reset	0	0	0	-	0	0	-	-

Bit	Name	Description
[7]	STEP_EN	Two-phase Step Controller Enable 0: Disable 1: Enable
[6]	EFAE	Forced Angle of Estimator Enable With it enabled, angle mode is determined by the estimator and automatically switches to estimated angle mode automatically. This bit is set to "1" when step controller is used to drive the motor. 0: Disable 1: Enable
[5]	VAB_SEL	Amplitude Compensation for Alpha-axis or Beta-axis Voltage 0: V_{alpha} is compensated. 1: V_{beta} is compensated.
[4]	RSV	Reserved
[3:2]	CSM	See descriptions on FOC_CR1 (0x40A0) in Chapter FOC.
[1:0]	RSV	Reserved

15.2.2 ST_VABK (0x40D8)

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

Bit	Name	Description
[7:0]	ST_VABK	This bit is available only when STEP_EN is set to "1" in step mode. In FOC control mode, this bit is FOC_EOMEKLPF. ST_VABK decides the compensation ratio for Alpha-axis or Beta-axis voltage. When it is set to "0", no compensation is performed. Otherwise, x-axis voltage input to SVPWM module Vx_d = Vx*VAB_K/256(x= Alpha, Beta).

15.2.3 ST_BPCOMP (0x40AE)

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

Bit	Name	Description
[7:0]	ST_BPCOMP	Compensated angle for Beta-axis voltage. When ST_BPCOMP is set to "0", no compensation is implemented and Alpha-axis voltage phase is 90 degrees ahead of that of Beta-axis voltage phase. When ST_BPCOMP!=0, Beta-axis voltage phase lags behind (90-ST_BPCOMP/32767*180) degrees.

15.2.4 ST_DZCOMP (0x40A2)

Bit	7	6	5	4	3	2	1	0
Name	ST_DZCOMP							
Type	R/W	R/W	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]	ST_DZCOMP	Compensation value for deadtime in step mode. (DRV_DTR + 1)/2 is recommended. If it is set to “0”, no compensation is provided.

15.2.5 ST__ETHETA(0x40AA, 0x40AB)

ST__ETHETAH(0x40AA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ST__ETHETA[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ST__ETHETAL(0x40AB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ST__ETHETA [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]	ST__ETHETA	The angle estimated by SMO in step mode. In forced pulling mode, FOC__THETA and FOC__ETHETA output forced pulling angle.

15.2.6 ST__OMEEST(0x40B0, 0x40B1)

ST__OMEESTH(0x40B0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ST__OMEEST[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ST__OMEESTL(0x40B1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ST__OMEEST[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]	ST__OMEESTH	The speed estimated by SMO in step mode. In forced pulling mode, FOC__EOME is generated by FOC_EFREQHOLD after low-pass filtering.

16 Timer1

16.1 Timer1 Operations

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

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

The internal structure of Timer1 is shown in Figure 16-1.

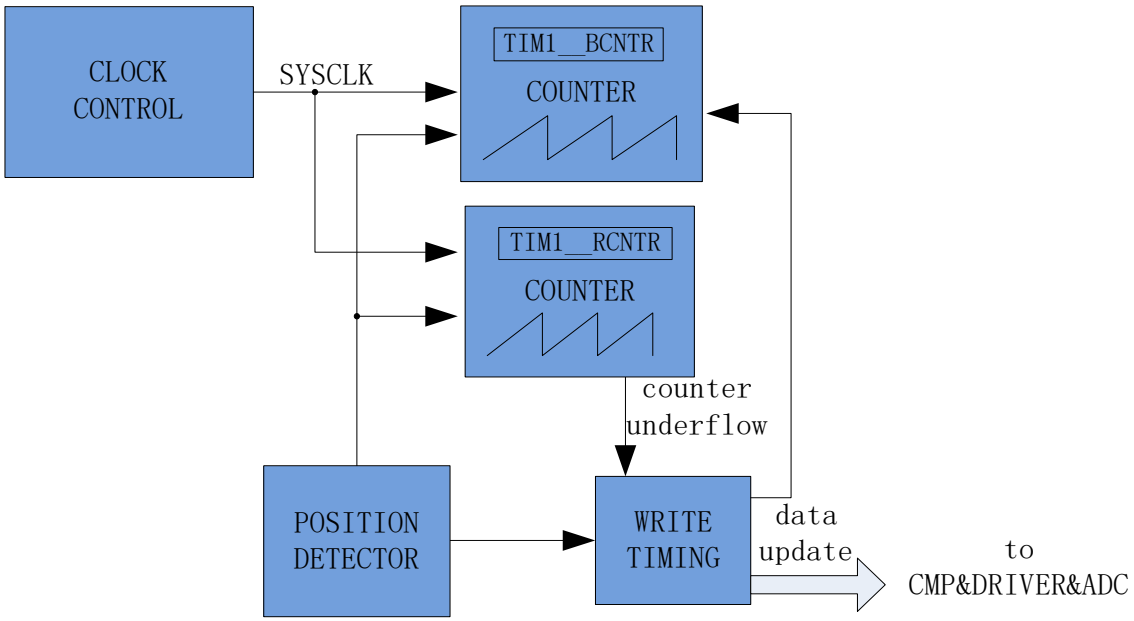


Figure 16-1 Timer1 Internal Structure

16.1.1 Timer1 Counter Module

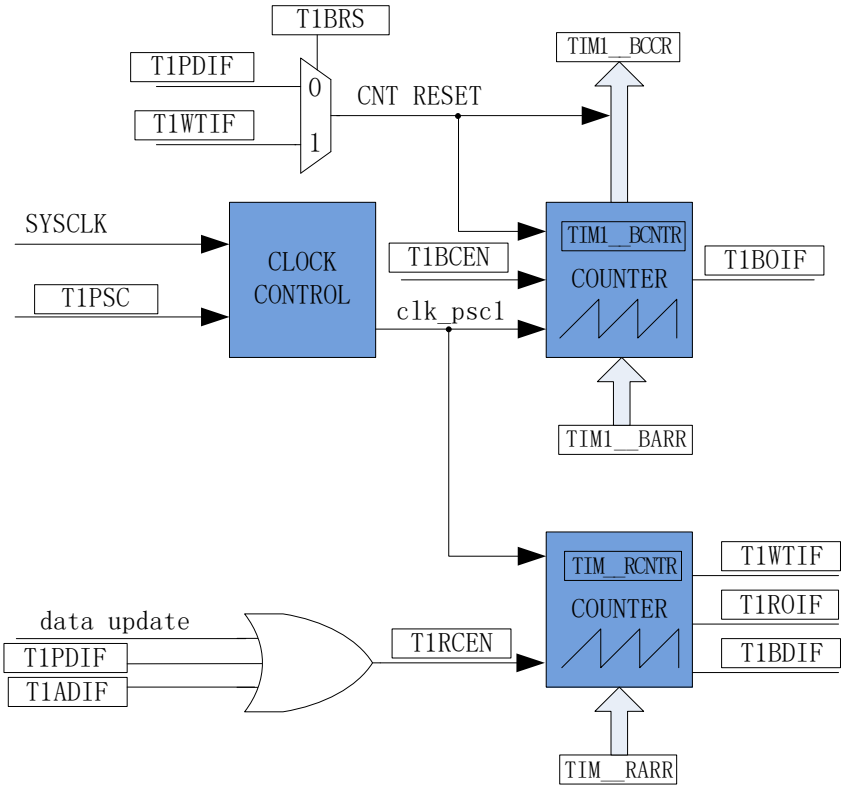


Figure 16-2 Timebase Unit

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

16.1.1.1 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 Base Timer and Reload Timer are not working. The clock rate clk_psc1 = SYSCLK/(2^TIM1_CR3[T1PSC]). The clock rate corresponding to TIM1_CR3[T1PSC] is shown in Table 16-1.

Table 16-1 Mapping between Clock Rate and TIM1_CR3[T1PSC] Bit

TIM1_CR3[T1PSC]	Division Factor	clk_psc1(Hz)	TIM1_CR3[T1PSC]	Division Factor	clk_psc1(Hz)
000	1	20M	100	16	1.25M
001	2	10M	101	32	625k
010	4	5M	110	64	312.5k
011	8	2.5M	111	128	156.25k

16.1.1.2 Base Timer

The Base Timer is a 16-bit up timer with its count value held in the TIM1__BCNTR register. TIM1__BCNTR value is loaded into Capture Register TIM1__BCCR upon a Position Detection Interrupt INT_SR0[T1PDIF] or a Write Timing Interrupt INT_SR0[T1WTIF] (selected by TIM1_CR2[T1BRS] bit). Meanwhile, TIM1__BCNTR bit is cleared to "0" and restarts the counter cycle. TIM1__BCCR captures the time between two Position Detection Interrupts or two Write Timing Interrupts (i.e. 60° commutation time). These time inputs are averaged multiple times (programmed by the TIM1_CR0[T1CFLT] bit) before loading the average as a 60° commutation base into the TIM1_BCOR register. When Auto-load Register TIM1__BARR is enabled (TIM1_CR1[BAPE] is set to "1"), TIM1__BARR loads the value of TIM1_BCOR by hardware. When count value of TIM1__BCNTR increases to TIM1__BARR, overflow interrupt flag INT_SR0[T1BOIF] of the Base Timer is set to "1". If forced commutation feature is enabled, phase commutation occurs and the Base Timer Register is cleared to "0". Otherwise, the Base Timer Register will not be cleared until it counts up to 0xFFFF and becomes overflowed.

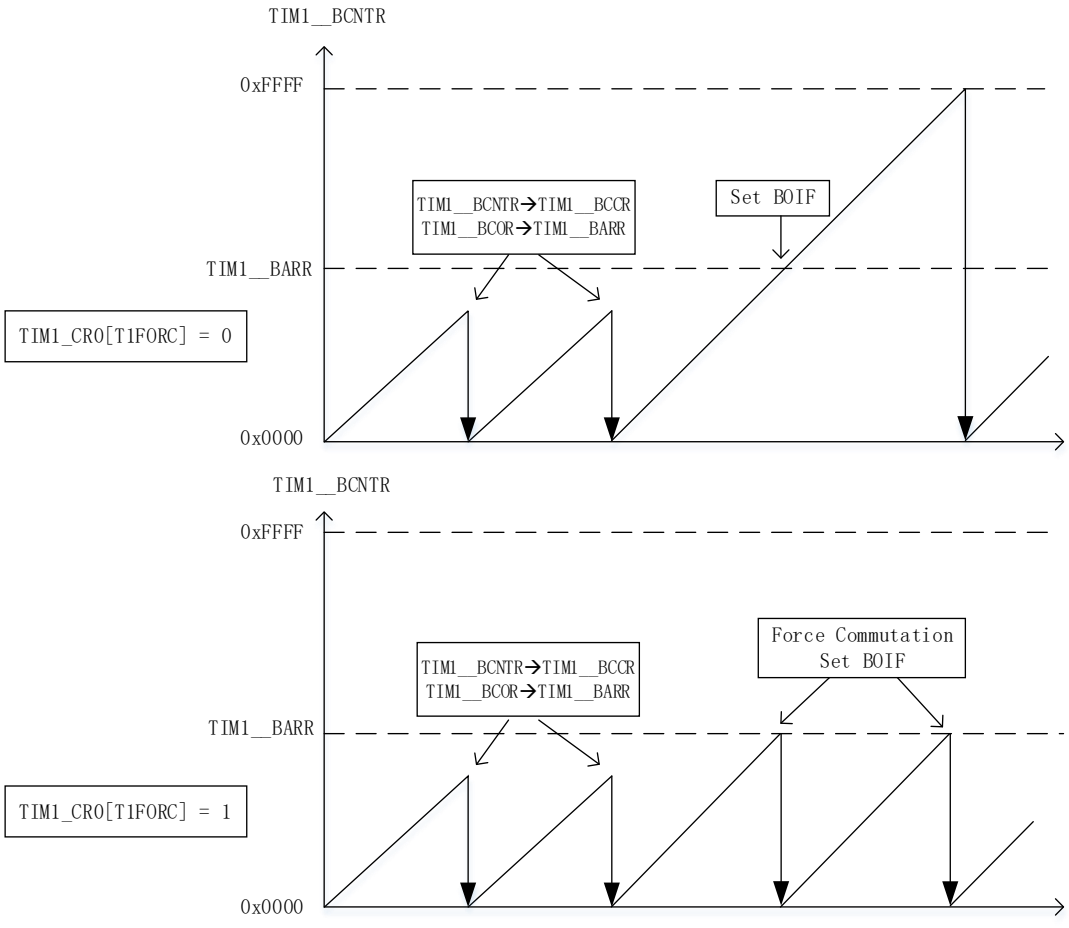


Figure 16-3 Waveform of Base Timer

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

16.1.1.3 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. It stops counting when INT_SR0[T1ROIF] (overflow interrupt flag of the reload counter) is set to "1", and TIM1__RCNTR and TIM1_CR0[T1RCEN] are cleared to "0". TIM1_CR0[T1RCEN] is set to "1" to restart Reload Timer when position detection interrupt or write timing interrupt is generated.

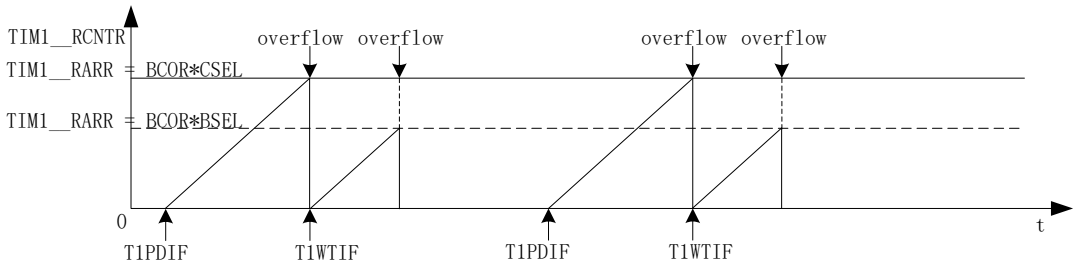


Figure 16-4 Waveform of Reload Timer

16.1.2 Position Detection

16.1.2.1 Position Detection Signal

The TIM1_CR3[T1TIS] bit selects the sources of Position Detection signal, including CMP0/1/2 (CMP Position Detection), GPIO (Hall Sensor Position Detection), ADC (ACD Position Detection) or CMP0/1/2 and ADC (CMP Position Detection and ACD Position Detection). HALL_CR[HALLSEL] bit is used to configure GPIO sourced by P1.4/P0.2/P0.3 (Hall signal input after function switching) or P1.4/P0.5/P0.6. TIM1_CR3[T1INM] bit decides whether CMP/GPIO signal is filtered. A Position Detection Interrupt is generated upon the completion of position detection. Position Detection Interrupts are divided into CMP/GPIO Position Detection Interrupt and ADC Position Detection Interrupt.

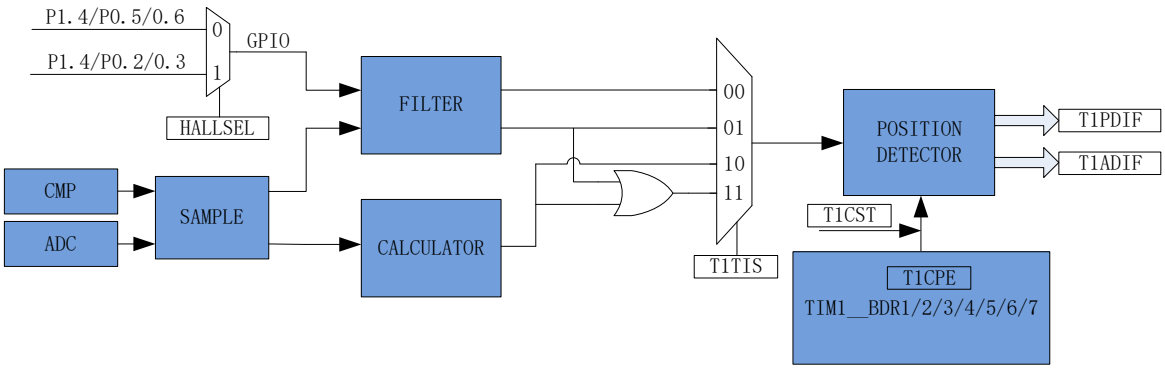


Figure 16-5 Functional Block Diagram of Position Detection

16.1.2.2 CMP/GPIO Position Detection Event

The register bank TIM1_DBR1/2/3/4/5/6/7[T1CPE] is configured to select the active edge of position detection signal. When an active edge of CMP/GPIO Position Detection signal is detected, it indicates the position detection is successfully done, allowing the CMP/GPIO Position Detection Interrupt Flag INT_SR0[T1PDIF] bit to become "1". TIM1_CR4[T1CST] bit selects TIM1_DBR1/2/3/4/5/6/7[T1CPE] timing.

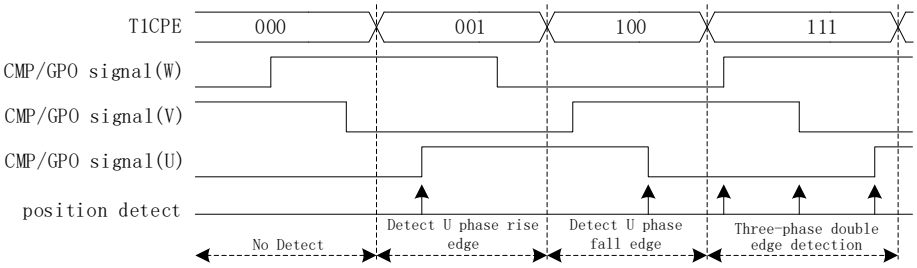


Figure 16-6 Timing Diagram of CMP/GPIO Position Detection

The relation between active edge and TIM1_DBR1/2/3/4/5/6/7[T1CPE] is shown in Table 16-2.

Table 16-2 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.

16.1.2.3 ADC Position Detection Event

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

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

Where,

K: ADC Position Detection Coefficient

TIM1__UCOP: ADC sampled value of active phase

TIM1__UFLP: ADC sampled value of floating phase

K, *TIM1__UCOP* and *TIM1__UFLP* definitions are determined by TIM1_DBR1/2/3/4/5/6/7[T1CPE] bit, as detailed in Table 16-3.

Table 16-3 Relation between TIM1_DBR1/2/3/4/5/6/7[T1CPE] and *K*, *TIM1__UCOP* and *TIM1__UFLP*

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

When TIM1__URES has a negative step or a positive step, an ADC Position Detection Interrupt is generated and INT_SR0[T1ADIF] (Position Detection Interrupt Flag) is set to "1". The position at which ADC Position Detection Interrupt is generated is controlled by setting the coefficient K. In this case, the phase commutation degree can be controlled flexibly.

16.1.2.4 Sampling

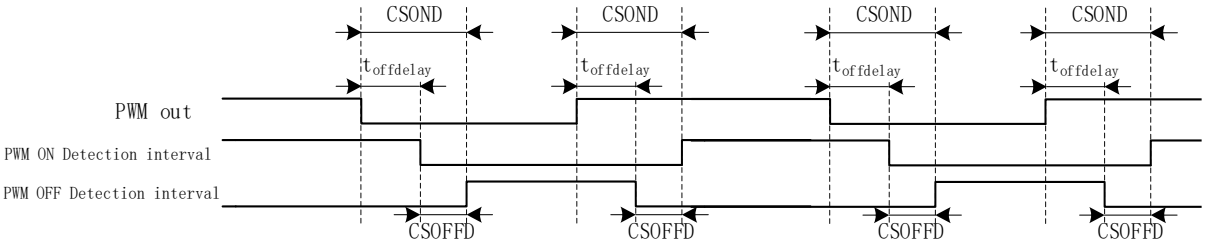


Figure 16-7 Timing Diagram of Sampling

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

See section 29.1.3 for details.

16.1.2.5 Filtering

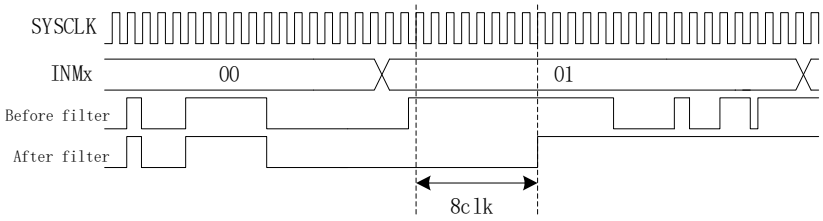


Figure 16-8 Timing Diagram of Filtering Module

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

16.1.3 Write Timing Interrupt

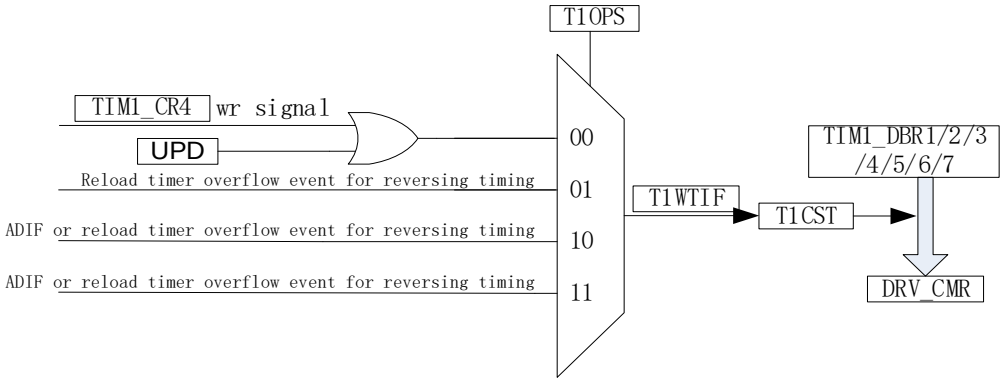


Figure 16-9 Write Timing Block Diagram

When the control logic, predefined in TIM1_DBR1/2/3/4/5/6/7, is sent to driver register DRV_CMR, a writing timing interrupt is generated. The triggered source is selected by the configuration of TIM1_CR0[T1OPS], and software, Reload Timer overflow event or position detection event can be selected. When a writing timing interrupt is generated, writing timing interrupt flag INT_SR0[T1WTIF] is set to “1”. If TIM1_CR4[T1CST] is in 001 ~ 110, TIM1_CR4[T1CST] adds 1 automatically.

16.1.4 Timer1 Interrupt

Timer1 supports 7 interrupt sources:

- Base Timer overflow interrupt
- Reload Timer overflow interrupt
- Writing timing interrupt

- Diode Freewheeling Masking End Interrupt
- CMP/GPIO Position Detection Interrupt
- ADC Position Detection Interrupt
- Adaptive Diode Freewheeling Masking Interrupt

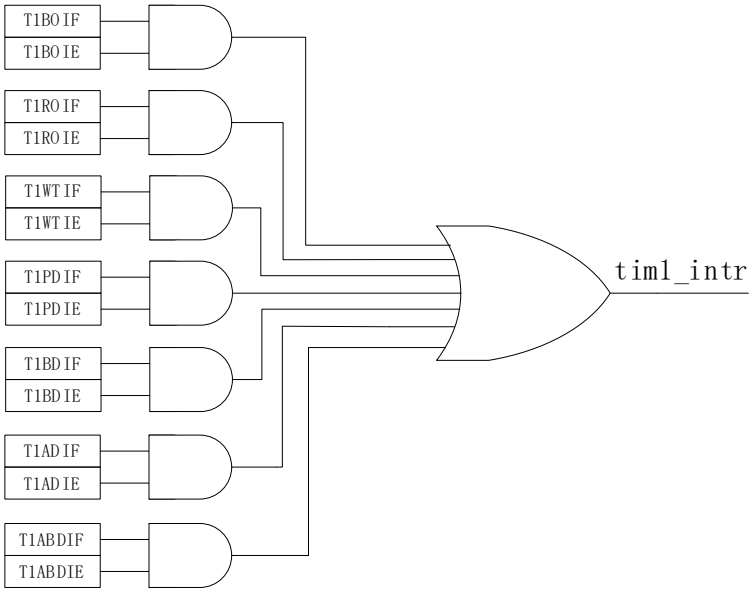


Figure 16-10 Timer1 Interrupt Sources

16.2 Square-wave Control Based BLDC Motor

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

- Automatic record of 60 degree time, filtered as 60 degree reference time
- Automatic forced phase commutation when position signal is not detected
- Automatic diode freewheeling masking, i.e., stopping position detection during diode freewheeling
- Automatic control of the time from position detection to phase commutation to achieve automatic commutation
- Take over CMP_CR2[CMP0SEL] to control CMP0/1/2 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 6 PWM outputs automatically

16.2.1 Six-step Phase Commutation of Square Wave Control

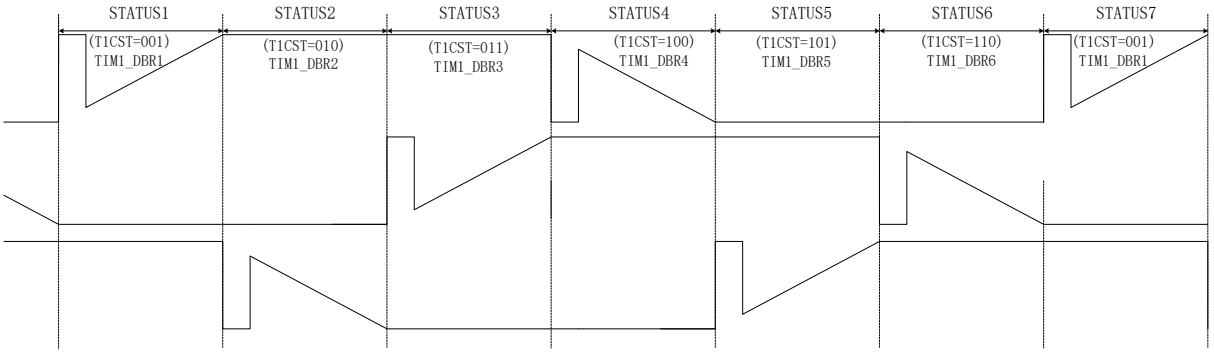


Figure 16-11 Diagram of Six-step Phase Commutation of Square Wave Control

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 timing 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.

16.2.2 Square Wave Control Working Principle

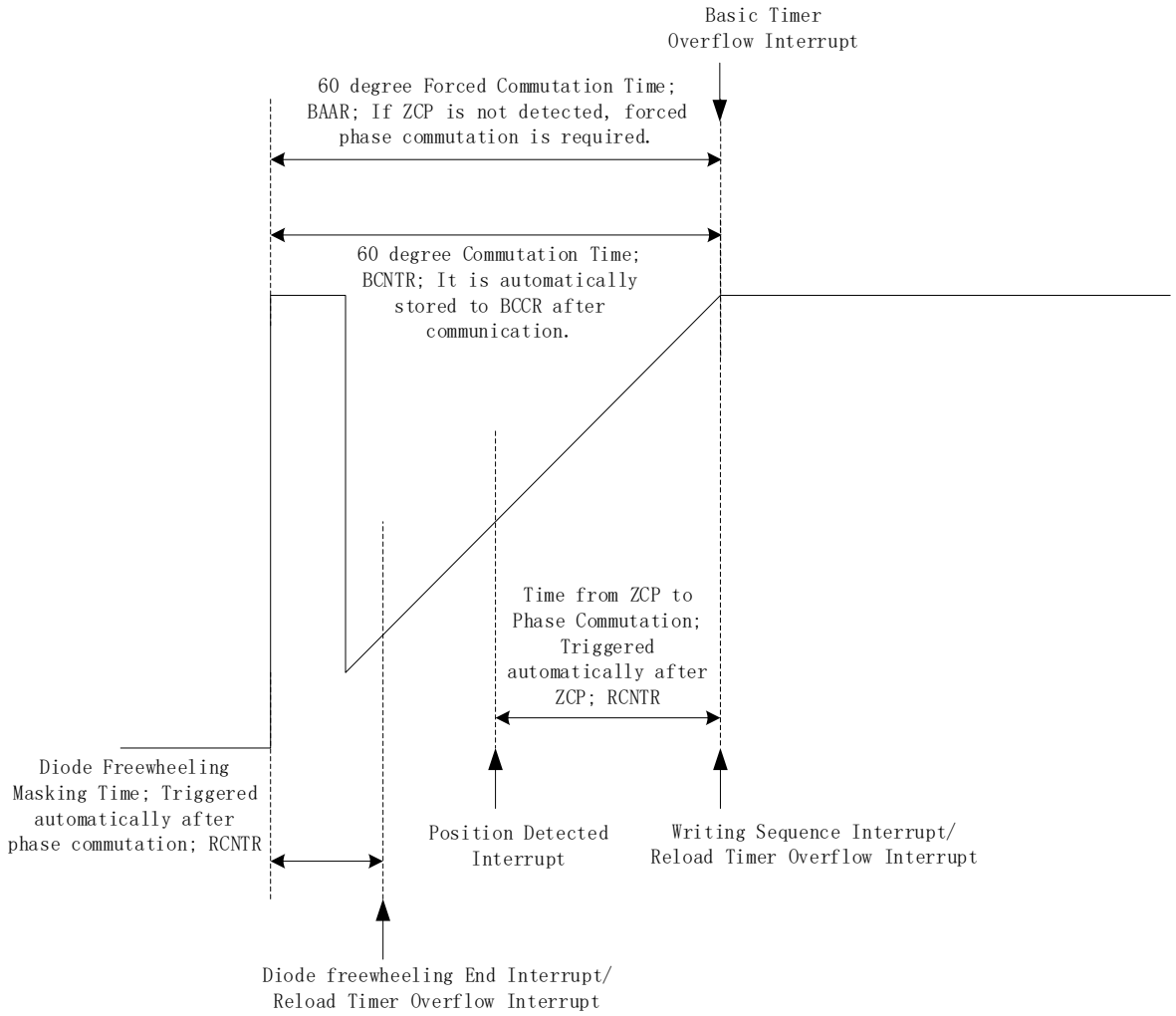


Figure 16-12 Square Wave Control Working Principle

16.2.2.1 60° Commutation Base Time

TIM1__BCCR captures the time of previous 60 degree. TIM1_CR2[T1BRS] is set to “0” to capture the time between two writing timing interrupts and TIM1_CR2[T1BRS] to “1” to capture the time between two position detection interrupts.

TIM1_BCOR is the filtered 60 degree time, i.e., 60 degree base time. TIM1_CR0[T1CFLT] can select the previous 1/2/4/8 TIM1__BCCR averaged to obtain TIM1_BCOR.

In square-wave control mode, the diode freewheeling masking time, the time from position detection to commutation, and the time to forced commutation are determined by the 60 degree base time TIM1_BCOR.

When Base Timer is auto-load enabled (TIM1_CR1[T1BAPE] = 1), and is reset due to a position detection interrupt or a write timing interrupt, TIM1_BCOR is transferred to TIM1__BARR to control the forced phase commutation.

16.2.2.2 Forced Commutation at 60°

When the motor rotates smoothly, ZCP is generally detected after 30 degrees of rotation after a phase commutation and a position detection interrupt is generated. If ZCP is not detected in 60 degree after the phase commutation, position detection fails and a forced phase commutation is required.

In this case, TIM1_CR0[T1FORC] is set to "1" to enable the forced commutation feature. During previous commutation, the counter TIM1__BCNTR is cleared to "0" by timing interrupt and restarts counting, while TIM1__BCCR captures the count value held in TIM1__BCNTR, which is filtered and stored in TIM1_BCOR as the 60 degree base time. When auto-load feature is enabled (TIM1_CR1[T1BAPE] = 1), the value held in TIM1_BCOR is loaded into TIM1__BARR after the Base Timer is cleared. If no ZCP is detected in 60 degree after commutation (TIM1__BCNTR matches TIM1__BARR), INT_SR0[T1BOIF] (overflow interrupt flag of the Base Timer) is set to "1" for forced phase commutation, and the counter TIM1__BCNTR is cleared to "0". But if an ZCP is detected within 60 degrees after phase commutation, even when TIM1__BCNTR > TIM1__BARR, the forced commutation will not be triggered and INT_SR0[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 INT_SR0[T1BOIF] is set to "1" and no forced phase commutation is automatically performed. Phase commutation can be performed manually by Base Timer overflow interrupt flag and the position detection interrupt flag.

16.2.2.3 Diode Freewheeling Masking by Timer

After the commutation, inductance energy of the phase is released to the power supply or ground through the diode since the original active phase becomes a floating phase. During diode freewheeling, the floating phase BEMF signal cannot be measured. By masking comparator signal or ADC sampling value during diode freewheeling, wrong commutation caused by wrong signal generated by the freewheeling is avoided. After freewheeling masking, the freewheeling masking end interrupt flag INT_SR0[T1BDIF] is generated.

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

16.2.2.4 Adaptive Diode Freewheeling Masking

During diode freewheeling, this feature ensures the freewheeling interval is larger than the actual one in condition that TIM1_CR1[BSEL] is not set to "0". Adaptive diode freewheeling masking is achieved by CMP or ADC. If it is implemented by CMP, TIM1_KRMAX is set to "0", TIM1_KFMIN ≠ 0xFF, TIM1_CR3[T1TIS] = 01/11 and related parameters to CMP0/1/2 are configured. If by ADC, TIM1_KRMAX ≠ 0, TIM1_CR3[T1TIS] =10/11 and related parameters to ADC are configured; and floating-phase voltage is controlled by TIM1_KRMAX and TIM1_KFMIN. Configuring TIM1_KRMAX=0 and TIM1_KFMIN=0xFF disables adaptive diode freewheeling masking feature. The interrupt flag bit INT_SR0[ABDIF] is set to "1" after adaptive diode

freewheeling masking is completed.

16.2.2.5 Angle of Position Detection to Phase Commutation

After phase commutation, a ZCP is detected (generating a position detection interrupt) and the hardware starts counting according to the software-set time between ZCP and the commutation. After the counting ends, the hardware automatically implements phase commutation and generates the writing timing interrupt flag INT_SR0[T1WTIF].

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

16.2.2.6 Cycle-by-cycle Current Limiting

See section 29.1.1.2.

16.3 Timer1 Registers

16.3.1 TIM1_CR0 (0x4068)

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

Bit	Name	Description
[7]	T1RWEN	Write to TIM1_CR0[T1RCEN] Enable 0: No effect 1: When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] and TIM1_CR0[T1RCEN] shall be configured simultaneously to enable or disable TIM1_CR0[T1RCEN]. A write of “0x81” to TIM1_CR0 enables TIM1_CR0[T1RCEN], and “0x80” to disables TIM1_CR0[T1RCEN].
[6:5]	T1CFLT	60 Degree Base Time Filtering Selection The average of previous x times (TIM1__BCCR)60 degree is used as the base time (TIM1__BCOR). 00: 1 time 60 degree 01: 2 times 60 degree 10: 4 times 60 degree 11: 8 times 60 degree
[4]	T1FORC	Forced Phase Commutation at 60° Enable 0: Disable 1: Enable Note: If a ZCP is detected, forced phase commutation is not implemented even if this bit is enabled.
[3:2]	T1OPS	Commutation Trigger Signal Select This bit selects trigger signal for TIM1_DBRx to transfer data to DRV_CMR. 00: The transfer is triggered upon a write of “1” to TIM1_IER[T1UPD] in software or on a write to TIM1_CR4[T1CST]. 01: The transfer is triggered upon a commutation counter overflow interrupt from reload timer 10: The transfer is triggered upon an ADC position detection event or a commutation counter overflow interrupt from reload timer. First come, first served. 11: The transfer is triggered upon an ADC position detection event before a CMP position detection event is generated, and the transfer is triggered upon a commutation counter overflow interrupt from reload timer after a CMP position detection event is generated. Note: When T1TIS=11, commutation counter of the reload timer is triggered by a CMP position detection event.
[1]	T1BCEN	Base Timer Enable 0: Disable 1: Enable
[0]	T1RCEN	Reload Timer Enable When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] andTIM1_CR0[T1RCEN] must be configured simultaneously to enable or disable TIM1_CR0[T1RCEN]. A write of “0x81” to TIM1_CR0 enables TIM1_CR0[T1RCEN] and “0x80” disables TIM1_CR0[T1RCEN]. TIM1_CR0[T1RCEN] is automatically enabled upon a Position Detection Interrupt and a Write Timing Interrupt. TIM1_CR0[T1RCEN] is cleared to “0” by hardware upon a Reload Timer Overflow Interrupt. TIM1_CR0[T1RCEN] cannot be automatically enabled or disabled by hardware in Manual mode. 0: Disable 1: Enable

16.3.2 TIM1_CR1 (0x4069)

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

Bit	Name	Description
[7]	T1BAPE	TIM1__BARR Register Auto-load Enable With this bit enabled, TIM1__BCOR is written to TIM1__BARR when Base Timer is reset due to a Position Detection Interrupt or a Write Timing Interrupt. It is used for forced phase commutation at 60°when no ZCP is detected. In manual mode, it has no effect on auto-load feature of TIM1__BARR Register. 0: Disable 1: Enable
[6:0]	BSEL	Diode Freewheeling Masking Angle Selection This bit is used to configure the angle 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.

16.3.3 TIM1_CR2 (0x406A)

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

Bit	Name	Description
[7]	T1BRS	Base Timer Reset Source 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 Detection Interrupt Reset
[6:0]	CSEL	Phase Commutation Angle Select After a position detection event, phase commutation is implemented after the degree configured by TIM1_CR2[CSEL]. Equation: Commutation angle = TIM1_CR2[CSEL]/128*60°

16.3.4 TIM1_CR3 (0x406B)

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

Bit	Name	Description
[7]	RSV	Reserved
[6:4]	T1PSC	Timer Clock Source Frequency Select 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: 20MHz 001: 10MHz 010: 5MHz 011: 2.5MHz 100: 1.25MHz 101: 625kHz 110: 312.5kHz 111: 156.25kHz

[3:2]	T1TIS	Position Detection Signal Select Flag 00: GPIO (select P1.4, P0.2 and P0.3 or P1.4, P0.5 and P0.6 according to HALL_CR[HALLSEL] bit) 01: Output signal of CMP0/1/2 10: Output signal of ADC 11: Output signals of CMP0/1/2 and ADC
[1:0]	T1INM	Filter Pulse Width for Position Detection Signal Select When pulse width of the input signal is less than the set value, it is filtered as noise. The filtering time changes according to CMP_CR4[FAEN]. When CMP_CR4[FAEN] = 0: 00: 4 system clock cycles 01: 8 system clock cycles 10: 16 system clock cycles 11: 24 system lock cycles When CMP_CR4[FAEN] = 1: 00: 32 system clock cycles 01: 64 system clock cycles 10: 96 system clock cycles 11: 128 system clock cycles

16.3.5 TIM1_CR4 (0x406C)

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

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

Table 16-4 Mapping between TIM1_CR4[T1CST] and TIM1_DBRx

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

16.3.6 TIM1_IER (0x406D)

Bit	7	6	5	4	3	2	1	0
Name	T1UPD	T1MAME	T1ADIE	T1BOIE	T1RUIE	T1WTIE	T1PDIE	T1BDIE
Type	W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	T1UPD	When TIM1_CR0[T1OPS] = 00, a write of "1" to this bit enables data transfer. This bit is cleared to "0" by hardware after "1" is written.
[6]	T1MAME	Manual Mode Enable With this bit enabled, Base Timer and Reload Timer acts as separate counters. Details: TIM1__BCNTR of the Base 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 can be configured 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 can be updated by software only. 0: Disable 1: Enable
[5]	T1ADIE	ADC Position Detection Interrupt Enable 0: Disable 1: Enable Note: See INT_SR0 (0xF2) for ADC Position Detection Interrupt Flag.
[4]	T1BOIE	Base Timer Overflow Interrupt Enable 0: Disable 1: Enable Note: See INT_SR0 (0xF2) for Base Timer Overflow Interrupt Flag.
[3]	T1ROIE	Reload Timer Overflow Interrupt Enable 0: Disable 1: Enable Note: See INT_SR0 (0xF2) for Reload Timer Overflow Interrupt Flag.
[2]	T1WTIE	Write Timing Interrupt Enable 0: Disable 1: Enable Note: See INT_SR0 (0xF2) for Writing Timing Interrupt Flag.
[1]	T1PDIE	CMP/GPIO Position Detection Interrupt Enable 0: Disable 1: Enable Note: See INT_SR0 (0xF2) for CMP/GPIO Position Detection Interrupt Flag.
[0]	T1BDIE	Diode Freewheeling Masking Interrupt Enable 0: Disable 1: Enable Note: See INT_SR0 (0xF2) for Diode Freewheeling Masking End Interrupt Flag.

16.3.7 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	This 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.

16.3.8 TIM1_CR5 (0x4072)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1POP	T1WTS	RSV		ITRIP_DIS	UCOP_DIS	T1AFL	
Type	R	R	-	-	R/W	R/W	R/W	R/W
Reset	0	0	-	-	0	0	0	0

Bit	**Name**	**Description**

[7]	T1ABDIE	Adaptive Diode Freewheeling Masking Interrupt Enable 0: Disable 1: Enable Note: See INT_SR5 (0xF7) for Adaptive Diode Freewheeling Masking Interrupt Flag.
[6]	T1WTS	Commutation enabled at PWM OFF to remove narrow pulses. PWM Synchronization Enable 0: Disable 1: Enable
[5:4]	RSV	Reserved
[3]	ITRIP_DIS	Bus Current Sampling Disable 0: Disable 1: Enable
[2]	UCOP_DIS	Active Phase Voltage Sampling Disable 0: Disable 1: Enable
[1:0]	T1AFL	ADC Sampled Voltage Calculation Filtering Counts 00: 1 01: 2 10: 4 11: 8

16.3.9 TIM1_DBR1 (0x4074, 0x4075)

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

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection 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 detection interrupt. See CMP/GPIO Position Detection Event and Table 16-2.
[11]	T1WHP	High-side Output Polarity of Phase W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of Phase W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of Phase V 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of Phase V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of Phase U 0: Active High 1: Active Low

[6]	T1ULP	Low-side Output Polarity of Phase U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of Phase W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of Phase V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of Phase V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of Phase U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of Phase U 0: Disable 1: Enable

Note: The high-side and low-side outputs of W, V and U-phases are complementary and dead time is automatically added (same for TIM1_DBR2~TIM1_DBR7) when TIM1_DBR1[T1WLE] and TIM1_DBR1[T1WHE], TIM1_DBR1[T1VLE] and TIM1_DBR1[T1VHE] or TIM1_DBR1[T1ULE] and TIM1_DBR1[T1UHE] are set to "1".

16.3.10 TIM1_DBR2 (0x4076, 0x4077)

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

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection 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 detection interrupt. See CMP/GPIO Position Detection Event and Table 16-2.
[11]	T1WHP	High-side Output Polarity of Phase W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of Phase W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of Phase V 0: Active High 1: Active Low

[8]	T1VLP	Low-side Output Polarity of Phase V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of Phase U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of Phase U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of Phase W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of Phase V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of Phase V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of Phase U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of Phase U 0: Disable 1: Enable

16.3.11 TIM1_DBR3 (0x4078, 0x4079)

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

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection 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 detection interrupt. See CMP/GPIO Position Detection Event and Table 16-2.
[11]	T1WHP	High-side Output Polarity of Phase W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of Phase W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of Phase V 0: Active High 1: Active Low

[8]	T1VLP	Low-side Output Polarity of Phase V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of Phase U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of Phase U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of Phase W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of Phase V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of Phase V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of Phase U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of Phase U 0: Disable 1: Enable

16.3.12 TIM1_DBR4 (0x407A, 0x407B)

TIM1_DBR4H(0x407A)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
TIM1_DBR4L(0x407B)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection 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 detection interrupt. See CMP/GPIO Position Detection Event and Table 16-2.
[11]	T1WHP	High-side Output Polarity of Phase W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of Phase W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of Phase V 0: Active High 1: Active Low

[8]	T1VLP	Low-side Output Polarity of Phase V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of Phase U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of Phase U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of Phase W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of Phase V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of Phase V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of Phase U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of Phase U 0: Disable 1: Enable

16.3.13 TIM1_DBR5 (0x407C, 0x407D)

TIM1_DBR5H(0x407C)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
TIM1_DBR5L(0x407D)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection 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 detection interrupt. See CMP/GPIO Position Detection Event and Table 16-2.
[11]	T1WHP	High-side Output Polarity of Phase W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of Phase W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of Phase V 0: Active High 1: Active Low

[8]	T1VLP	Low-side Output Polarity of Phase V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of Phase U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of Phase U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of Phase W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of Phase V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of Phase V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of Phase U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of Phase U 0: Disable 1: Enable

16.3.14 TIM1_DBR6 (0x407E, 0x407F)

TIM1_DBR6H(0x407E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
TIM1_DBR6L(0x407F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection 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 detection interrupt. See CMP/GPIO Position Detection Event and Table 16-2.
[11]	T1WHP	High-side Output Polarity of Phase W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of Phase W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of Phase V 0: Active High 1: Active Low

[8]	T1VLP	Low-side Output Polarity of Phase V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of Phase U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of Phase U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of Phase W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of Phase V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of Phase V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of Phase U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of Phase U 0: Disable 1: Enable

16.3.15 TIM1_DBR7 (0x4080, 0x4081)

TIM1_DBR7H(0x4080)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
TIM1_DBR7L(0x4081)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection 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 detection interrupt. See CMP/GPIO Position Detection Event and Table 16-2.
[11]	T1WHP	High-side Output Polarity of Phase W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of Phase W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of Phase V 0: Active High 1: Active Low

[8]	T1VLP	Low-side Output Polarity of Phase V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of Phase U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of Phase U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of Phase W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of Phase V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of Phase V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of Phase U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of Phase U 0: Disable 1: Enable

16.3.16 TIM1__BCNTR (0x4082, 0x4083)

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

Bit	Name	Description
[15:0]	TIM1__BCNTR	This bit holds count values of the Base Timer and is used for clocking commutation at 60°. Auto mode: TIM1__BCNTR register selects the reset source according to TIM1_CR2[T1BRS], and TIM1__BCNTR does not restart when TIM1__BCNTR overflow interrupt is generated. Manual mode: TIM1__BCNTR restarts when TIM1__BCNTR overflow interrupt is generated.

16.3.17 TIM1__BCCR (0x4084, 0x4085)

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

Name	TIM1__BCCR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1__BCCR	This bit is configured to capture count values held in Base Timer. Auto mode: When the Base Timer is reset on a Position Detection Interrupt or a Write Timing Interrupt, the count values before the reset are stored into TIM1__BCCR. Manual mode: When the Base Timer is reset on an Overflow Interrupt, the count values before the reset are stored into TIM1__BCCR.

16.3.18 TIM1__BARR (0x4086, 0x4087)

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

Bit	Name	Description
[15:0]	TIM1__BARR	Reload Value 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”.

16.3.19 TIM1__RARR (0x4088, 0x4089)

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

Bit	Name	Description
[15:0]	TIM1__RARR	Auto-reload Value in Reload Timer When count of the Reload Timer is equal to TIM1__RARR, an overflow interrupt is generated and the value of counter is cleared to “0”. Auto mode: The value of diode freewheeling masking angle held in TIM1_CR1[BSEL] is updated to TIM1__RARR when a Writing Timing Interrupt is generated. The value of commutation angle held in TIM1_CR2[CSEL] is updated to TIM1__RARR when a Position Detection Interrupt occurs. Manual mode: TIM1__RARR is written by software.

16.3.20 TIM1__RCNTR (0x408A, 0x408B)

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

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

16.3.21 TIM1__UCOP (0x408C, 0x408D)

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

Bit	Name	Description
[15:0]	TIM1__UCOP	ADC sampled value of active phase voltage (second-highest bit alignment)

16.3.22 TIM1__UFLP (0x408E, 0x408F)

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

Bit	Name	Description
[15:0]	TIM1__UFLP	ADC sampled value of floating phase voltage (second-highest bit alignment)

16.3.23 TIM1__URES (0x4090, 0x4091)

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

Bit	**Name**	**Description**
[15:0]	TIM1__URES	Result of ADC position detection formula; Q15 format

16.3.24 TIM1_KRMAX (0x4092)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1_KRMAX							
Type	R/W	R/W	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]	TIM1_KRMAX	Max. Coefficient of Raising Edge; Range (0, 255)

16.3.25 TIM1_KFMIN (0x4093)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1_KFMIN							
Type	R/W	R/W	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]	TIM1_KFMIN	Min. Coefficient of Falling Edge; Range (0, 255)

16.3.26 TIM1_KF (0x4094, 0x4095)

TIM1_KFH(0x4094)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1_KF[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_KFL(0x4095)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1_KF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIM1_KF	ADC position detection coefficient when floating phase voltage drops Range [0,32767]

16.3.27 TIM1_KR (0x4096, 0x4097)

TIM1_KRH(0x4096)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	TIM1_KR[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_KRL(0x4097)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM1_KR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	TIM1_KR	ADC position detection coefficient when floating phase voltage rises Range [0, 32767]

16.3.28 TIM1__ITRIP (0x4098, 0x4099)

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

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

17 Timer2

17.1 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 Count 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

17.1.1 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 Base Timer is not working. The clock source frequency formula is: clk_psc2 = T2CLK/(2^TIM2_CR0[T2PSC]). The clock rate corresponding to different TIM2_CR0[T2PSC] value as shown in Table 17-1.

Table 17-1 Mapping between Clock Rate and TIM2_CR0[T2PSC]

TIM2_CR0[T2PSC]	Division Factor	clk_psc2(Hz)	TIM2_CR0[T2PSC]	Division Factor	clk_psc2(Hz)
000	1	20M	100	16	1.25M
001	2	10M	101	32	625k
010	4	5M	110	64	312.5k
011	8	2.5M	111	128	156.25k

17.1.2 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, the 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.

17.1.3 Output Mode

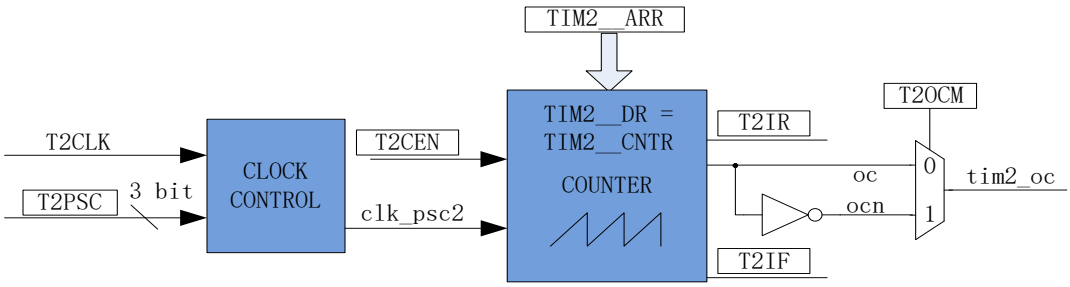


Figure 17-1 Output Mode Block Diagram

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

17.1.3.1 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 INT_SR2[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 and before the low byte is written.

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

17.1.3.2 High/Low Level Output

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

17.1.3.3 PWM Generation

In PWM generation mode, TIM2__ARR determines PWM cycle, TIM2__DR determines duty cycle, and duty cycle = TIM2__DR/TIM2__ARR*100%. If TIM2_CR0[T2OCM] = 0, the low level is output when TIM2__CNTR < TIM2__DR, and the high level is output when TIM2__CNTR ≥ TIM2__DR. If TIM2_CR0[T2OCM] = 1, the high level is output when TIM2__CNTR < TIM2__DR, and the low level is

output when TIM2__CNTR ≥ TIM2__DR. When TIM2__CNTR is increased to TIM2__ARR, the output signal is reversed.

17.1.3.4 Interrupts

- When TIM2__CNTR = TIM2__DR, a compare match event is generated and the interrupt flag bit INT_SR2[T2IR] is set to "1". The timer continues.
- When TIM2__CNTR = TIM2__ARR, an overflow event is generated, and the interrupt flag bit INT_SR2[T2IF] is set to "1". The timer is cleared to "0" and then restarts

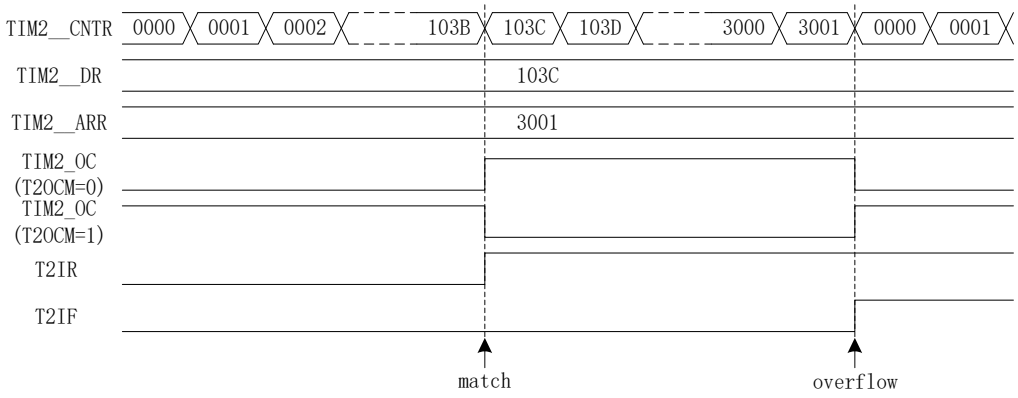


Figure 17-2 Output Mode Waveform

17.1.4 Input Signal Filtering and Edge Detection

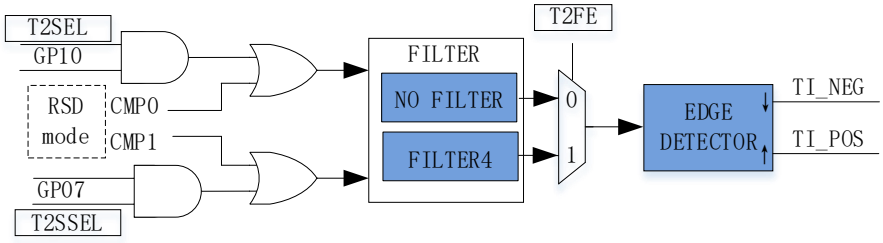


Figure 17-3 Block Diagram of Input Signal Filtering and Edge Detection

The input signal of Timer2 comes from P0.7 or P1.0, set by PH_SEL[T2SEL] and PH_SEL[T2SSEL] (see section 23.3.1). The filter of input signal is optional.

The filter circuit does not filter out or remove input noise less than 4 SYSCLK cycles. The filtering period is selected by setting TIM2_CR1[T2FE]. When TIM2_CR1[T2FE] is set to "0", filtering circuit is disabled; and when TIM2_CR1[T2FE] is set to "1", filtering circuit filters signals every 4 system cycles. The filtered signal is 4 clock cycles later than the signal before filtering. TIM2_CR0[T2CES] determines the active edge to count.

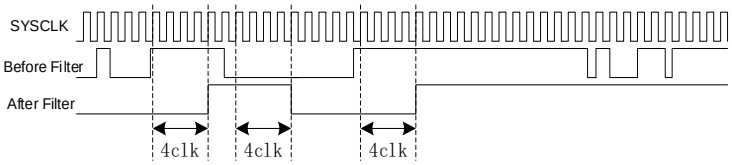


Figure 17-4 Timing Diagram of Filter Module

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

17.1.5 Input Capture Mode

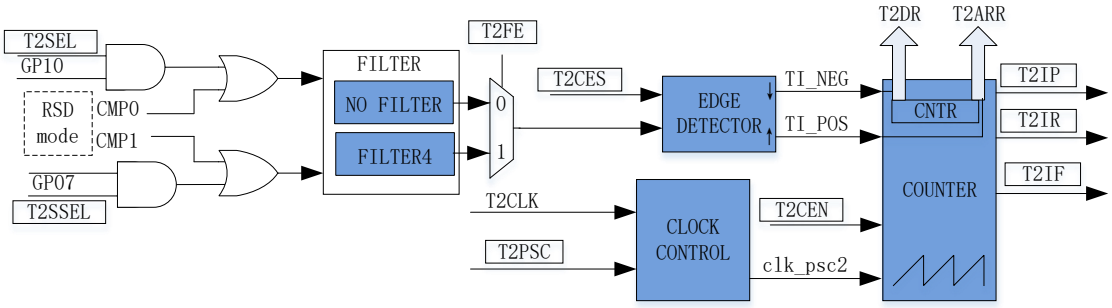


Figure 17-5 Schematic Diagram of Input Capture Mode

The input capture mode detects duty cycle and period of the PWM signal. When TIM_CR0[T2CES] = 0, the time between two adjacent rising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). When TIM_CR0[T2CES] = 1, the time between two adjacent falling edges forms one cycle, and the time from falling edge to rising edge forms the pulse width (LOW). When the predefined edge arrives, the count value TIM2__CNTR is stored in TIM2__DR and TIM2__ARR respectively to calculate the period and duty cycle of PWM waveform.

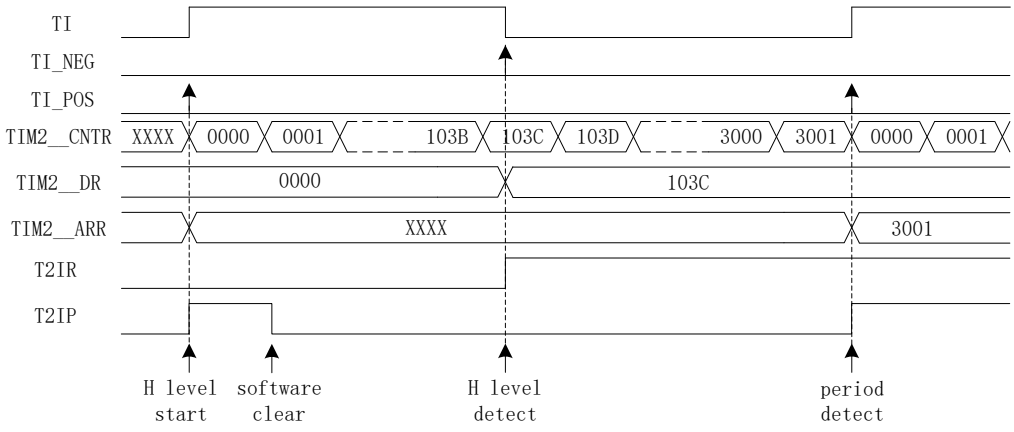


Figure 17-6 Timing Diagram of Input Capture Mode (TIM2_CR0[T2CES] = 0)

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, while the interrupt flag INT_SR2[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 INT_SR2[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 INT_SR2[T2IF] is set to “1”, and TIM2__CNTR is cleared to “0” and restarts. At this point, TIM2__ARR value is 0xFFFF, and the TIM2__DR value is determined by the input level and TIM2_CR0[T2OCM] XOR.

17.1.6 Input Counter Mode

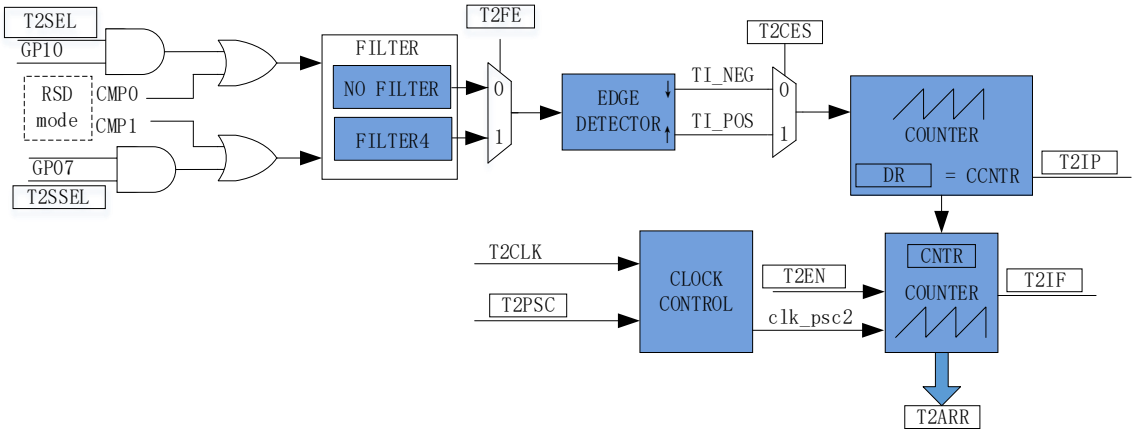


Figure 17-7 Schematic Diagram of Input Counter Mode

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 (INT_SR2[T2IP] = 1), overflow event (INT_SR2[T2IF] = 1) or special 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. When the number of input PWM counted by the special counter CCNTR reaches the set value (TIM2__DR), TIM2__CNTR of the Base Timer is stored in TIM2__ARR. 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.

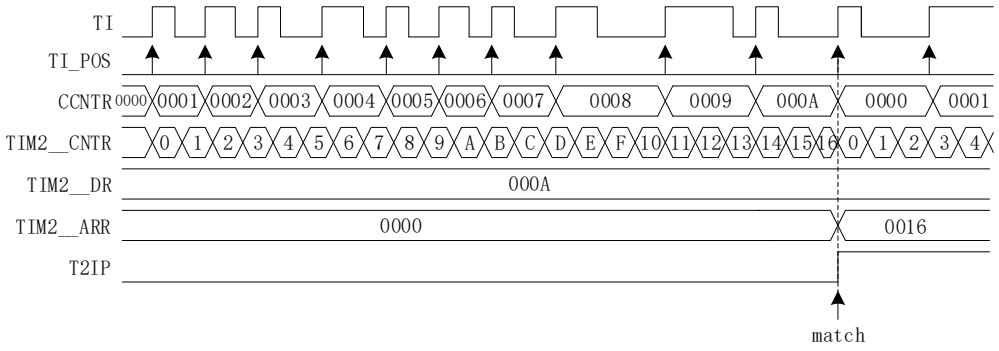


Figure 17-8 Timing Diagram of Input Count Mode

The Base 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 counter 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 occurs, and the interrupt flag INT_SR2[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.

17.1.7 QEP&RSD Mode

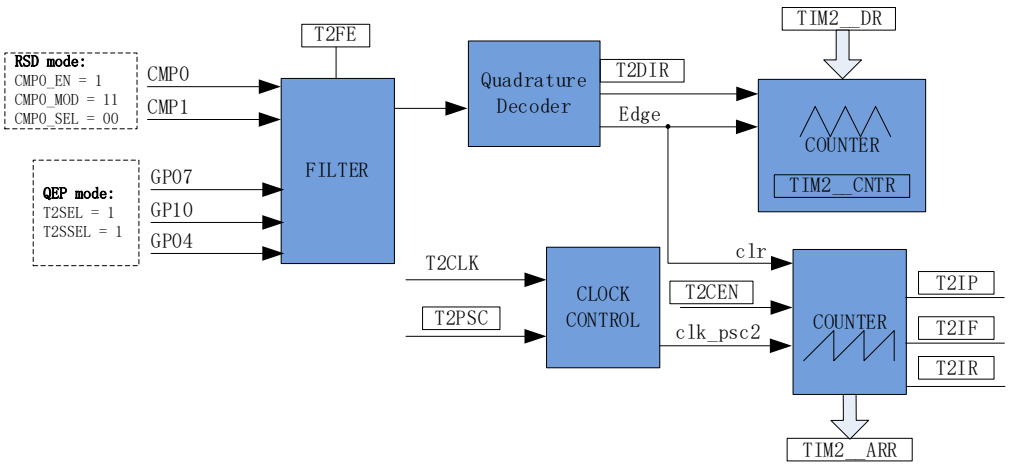


Figure 17-9 Schematic Diagram of QEP&RSD Mode

QEP & RSD mode obtains relative position, direction and speed of the motor by detecting orthogonal signals on two channels. P0. 7 and P1.0 (QEP mode) or CMP0, CMP1 (RSD mode) 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]).

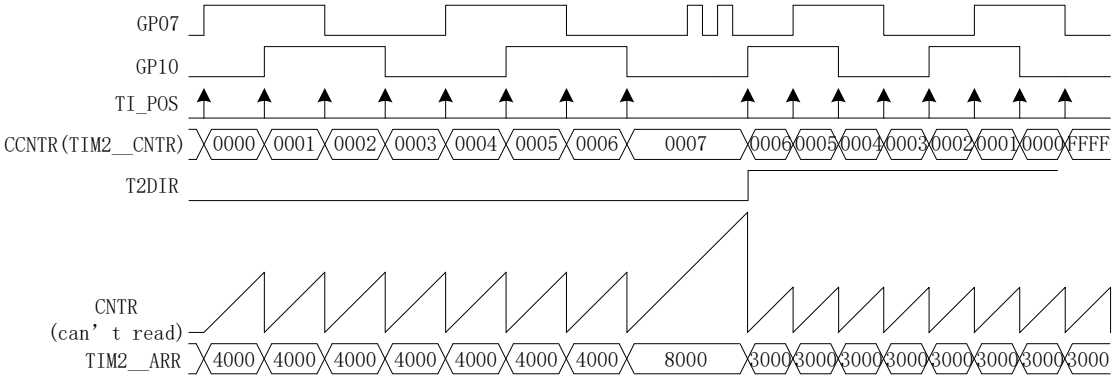


Figure 17-10 Timing Diagram of QEP&RSD Mode

The special timer is an up/down counter, and the signal source is the active edge from orthogonal decoding module. If TIM2_CR1[T2DIR] = 0, the direction is positive, and 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. In QEP Mode, after configuring the code value held in TIM2__DR, the count-up timer is cleared to "0" and restarts when it reaches TIM2__DR, and the count-down counter is reloaded with TIM2__DR when it decrements up to 0. The mechanical zero signal ("Z signal") of QEP encoder is input from P0.4 and generates the INT_SR2[T2IR] interrupt event flag.

The Base Timer is an up-counter used to record the time of two active counting edges. The clock source frequency can be divided. When the counting edge arrives, the value of Base Timer is stored in TIM2__ARR and cleared to "0", and INT_SR2[T2IP] interrupt flag bit is set to "1". When Base Timer counts to 0xFFFF, the count overflows and INT_SR2 [T2IF] interrupt flag is generated.

17.1.7.1 RSD Comparator Sampling

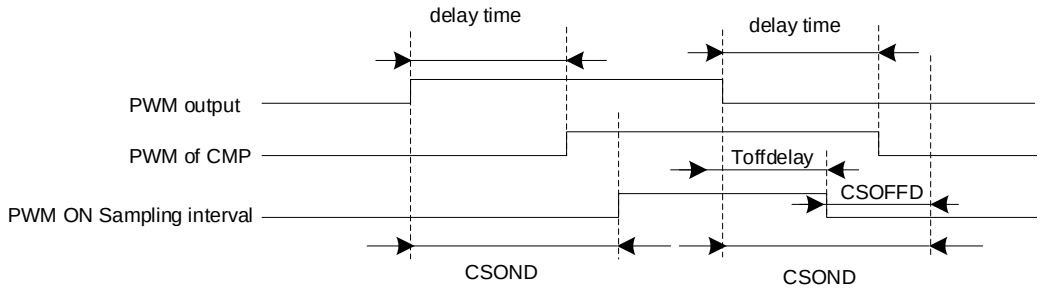


Figure 17-11 PWM ON Sampling Mode

The Start of Sampling ("SoS") time delay and End of Sampling ("EoS") time advance must be set in order to sample correct BEMF comparison signals in RSD Sampling mode.

See section 29.1.3 for details.

17.1.8 Step Mode

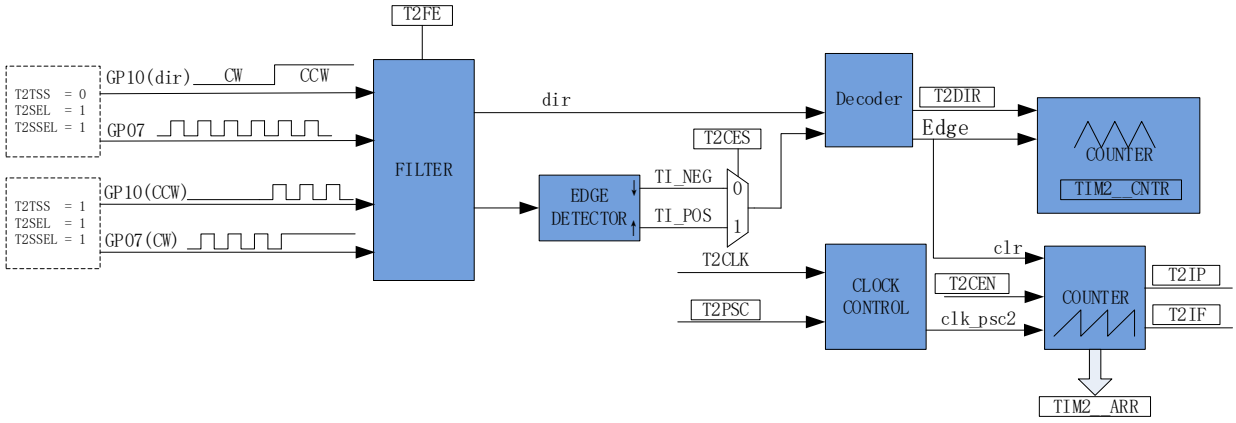


Figure 17-12 Step Mode Schematic Diagram

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].

Note: TIM2_CR1[T2DIR] will not change unless transition occurs at P1.0 and active edge is detected at P0.7. To generate an interrupt immediately after P1.0 changes, use INT1.

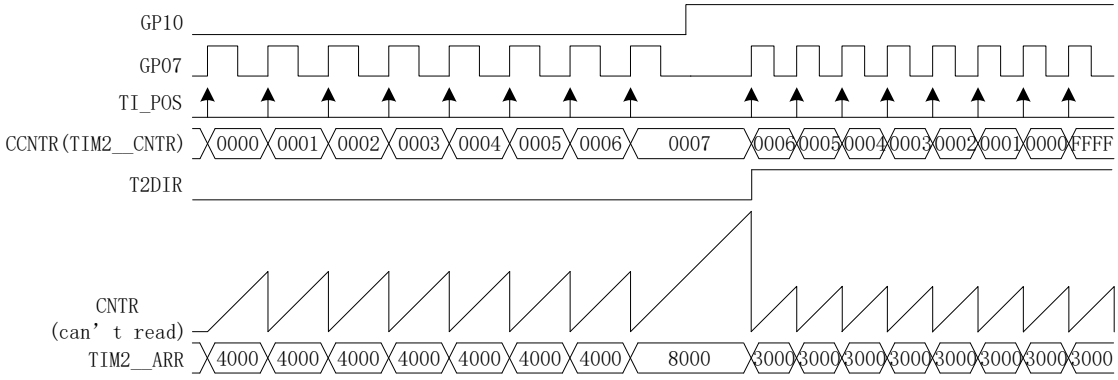


Figure 17-13 Timing Diagram of Step Mode

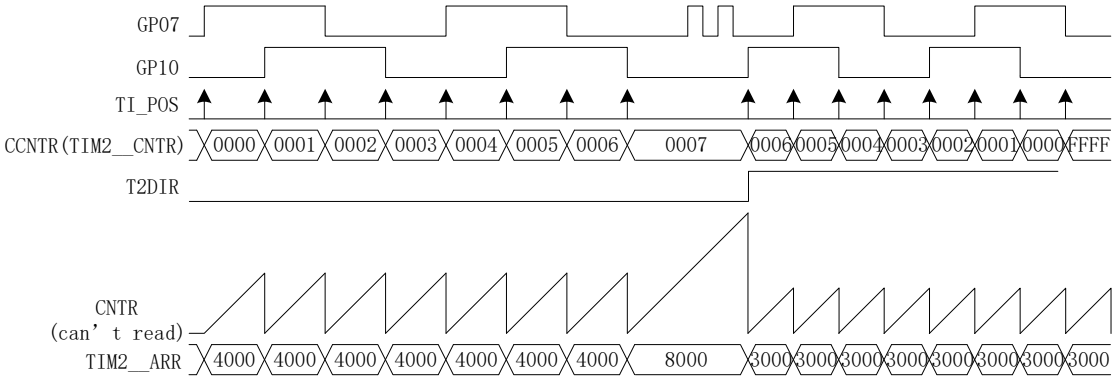


Figure 17-14 Timing Diagram of Positive + Negative Pulse Input State in Step Mode

(Raising Edge Selected as Active Edge)

The special timer is an up/down-counter, and the signal source is active edge of the encoding module. When P1.0 = 0, TIM2_CR1[T2DIR] = 0, the direction is forward. If active edge of P0.7 arrives, the special timer CCNTR increases by 1. When P1.0 = 1, TIM2_CR1[T2DIR] = 1 and the direction is reverse. If active edge of P0.7 arrives, CCNTR 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 Base Timer is stored in TIM2__ARR and then cleared to "0", and INT_SR2[T2IP] interrupt flag bit is set to "1". When Base Timer counts to 0xFFFF, the count overflows and INT_SR2[T2IF] interrupt flag is generated.

17.2 Timer2 Registers

17.2.1 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. The prescaled clock rates are configured as follows: 000: 20MHz 001: 10MHz 010: 5MHz 011: 2.5MHz 100: 1.25MHz 101: 625kHz 110: 312.5kHz 111: 156.25kHz
[4]	T2OCM	Output Mode: Output Mode Selection 0: Output “0” when TIM2__CNTR < TIM2__DR; output “1” when TIM2__CNTR ≥ TIM2__DR 1: Output “1” when TIM2__CNTR < TIM2__DR; output “0” when TIM2__CNTR ≥ TIM2__DR Input Count Mode: No effect. Input Capture Mode: TIM2__DR indicates the input level to be selected when timer TIM2__CNTR becomes overflowed. 0: TIM2__DR is reset to “0” by hardware for low level input upon an overflow interrupt and is set to “0xFFFF” for high level input upon an overflow interrupt. 1: TIM2__DR is reset to “0” by hardware for high level input upon an overflow interrupt and is set to “0xFFFF” for low level output upon an overflow interrupt. QEP&RSD Mode and Step Mode Selection 0: QEP&RSD Mode 1: Step Mode
[3]	T2IRE	Output Mode: Compare Match Interrupt Enable Input Capture Mode: Pulse Width Detection Interrupt Enable Input Count Mode: No effect QEP Mode: QEP Encoder Z Signal Interrupt Enable 0: Disable 1: Enable Note: See INT_SR2 (0xF4) for Timer2 IR Interrupt Flag.
[2]	T2CES	Output Mode: No effect 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). 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 Count Mode: Active Edge Selection 0: Falling Edge Count 1: Raising Edge Count QEP&RSD Mode: Enable Pulse Counter Cleared upon Z Signal Interrupt INT1 0: Disable 1: Enable Step Mode: Active Edge Selection 0: Falling Edge Count 1: Raising Edge Count
[1:0]	T2MOD	Mode Selection 00: Input Capture Mode 01: Output Mode 10: Input Count Mode 11: QEP&RSD Mode or Step Mode

17.2.2 TIM2_CR1 (0xA9)

Bit	7	6	5	4	3	2	1	0
Name	RSV		T2SS	T2IPE	T2IFE	T2FE	T2DIR	T2CEN
Type	-	-	R/W0	R/W	R/W	R/W	R	R/W
Reset	-	-	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	T2SS	Input Mode Selection of TIM2 Step Motor 0: Direction + Pulse Input Mode. Direction is input via P1.0, and pulse via P0.7. 1: Forward and Reverse Pulse Mode. Reverse pulse is input via P1.0, and forward pulse via P0.7.
[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 Note: See INT_SR2 (0xF4) for Timer2 IP Interrupt Flag.
[3]	T2IFE	Output Mode: Base Timer Overflow Interrupt Enable Input Capture Mode: Base Timer Overflow Interrupt Enable Input Count Mode: Base Timer Overflow Interrupt Enable QEP&RSD Mode and Step Mode: Base Timer Overflow Interrupt Enable 0: Disable 1: Enable Note: See INT_SR2 (0xF4) for Timer2 IF Interrupt Flag.
[2]	T2FE	Input Signal Filter Selection When TIM2_CR1[T2FE] = 1, input signals are filtered out as noise if the pulse width is less than 4 clock cycle. Assuming that the system clock runs at 24MHz (50ns) , then the pulse width for filtering is 200ns. 0: Not to filter signals 1: Signals filtered on every 4 clock cycles
[1]	T2DIR	QEP&RSD Mode: Indicator of Motor Rotation Direction Rotation direction of the motor is determined according to the phase relationship of the two input signals. Step Mode: Indicator of Motor Rotation Direction Rotation direction of the motor is determined according to the direction signal P1.0. 0: Forward 1: Backward
[0]	T2CEN	Base Timer Enable 0: Disable 1: Enable

17.2.3 TIM2__CNTR (0xAA, 0xAB)

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

Bit	Name	Description
[15:0]	TIM2__CNTR	Output Mode/Input Capture Mode/Input Count Mode: Count values held in the Base Timer QEP&RSD Mode and Step Mode: count values held in the special timer

17.2.4 TIM2__DR (0xAC, 0xAD)

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

Bit	Name	Description
[15:0]	TIM2__DR	Output Mode: Compare match value (written by software) Input Capture Mode: Count value of the detected input pulse width (written by hardware) Input Count Mode: PWM cycles to be counted (written by software) QEP Mode: Encoder value Step Mode: No effect

17.2.5 TIM2__ARR (0xAE, 0xAF)

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

Bit	Name	Description
[15:0]	TIM2__ARR	Output Mode: PWM cycle (written by software) Input Capture Mode: Count value held in Base Timer of a PWM cycle (written by hardware) Input Count 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)

18 Timer3/Timer4

18.1 Timer3/Timer4 Instructions

Timer3/Timer4 support output and input modes:

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

Timer3/Timer4 Features:

- 3-bit programmable prescaler divides system clock as the clock source for Base Timer (clock source of Timer3 can be doubled to 40MHz in input capture mode)
- 16-bit up-counting Base Timer; The output of the prescaler serves as the counting clock source
- Input signal filtering
- Input signal edge detection
- Output PWM signal, single compare output
- Timer4 FG generation
- Interrupt event

18.1.1 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 Base Timer is not working. The clock source frequency formula is: clk_pscx = TxCLK/(2^TxPSC). The clock rate corresponding to different TIMx_CR0[TxPSC] value as shown in Table 18-1.

Table 18-1 Mapping between Clock Rate and TIMx_CR0[TxPSC]

TIMx_CR0[TxPSC]	Division Factor	clk_pscx(Hz)	TIMx_CR0[TxPSC]	Division Factor	clk_pscx(Hz)
000	1	20M	100	16	1.25M
001	2	10M	101	32	625k
010	4	5M	110	64	312.5k
011	8	2.5M	111	128	156.25k

Note: In Input Capture Mode of Timer3, the clock rate is 40MHz when TIM3_CR0[T3PSC] = 111.

18.1.2 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.

18.1.3 Output Mode

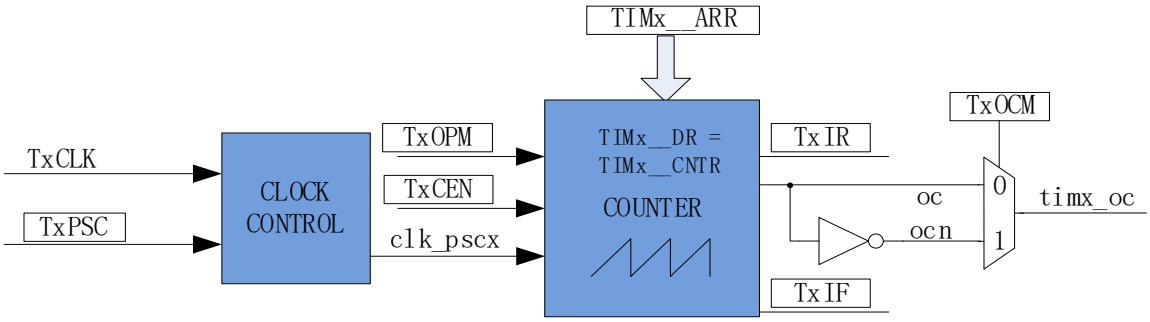


Figure 18-1 Output Mode Block Diagram

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

18.1.3.1 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.

18.1.3.2 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. When TIMx__CNTR > TIMx__ARR, the output signal is reversed.

18.1.3.3 Interrupt Event

- When TIMx__CNTR = TIMx__DR, a compare match interrupt is generated. The interrupt flag INT_SR1[TxIR] is set to “1”, and the timer continues.
- When TIMx__CNTR = TIMx__ARR, an overflow event is generated. The interrupt flag INT_SR1[TxIF] is set to “1”, and the counter 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.

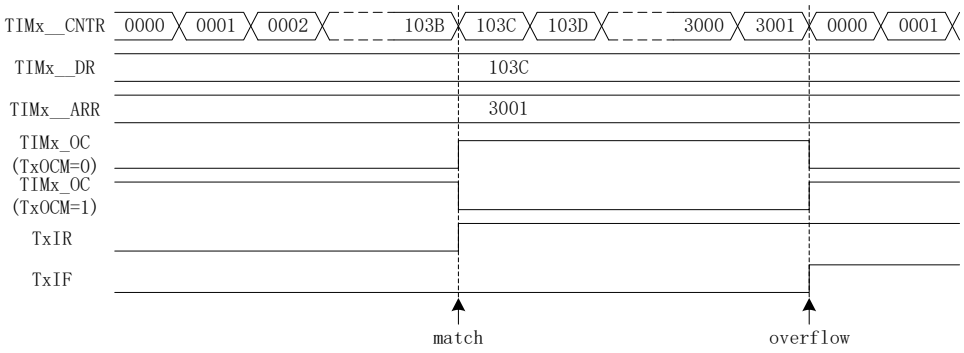


Figure 18-2 Output Waveform of Output Mode

18.1.4 Input Signal Filtering and Edge Detection

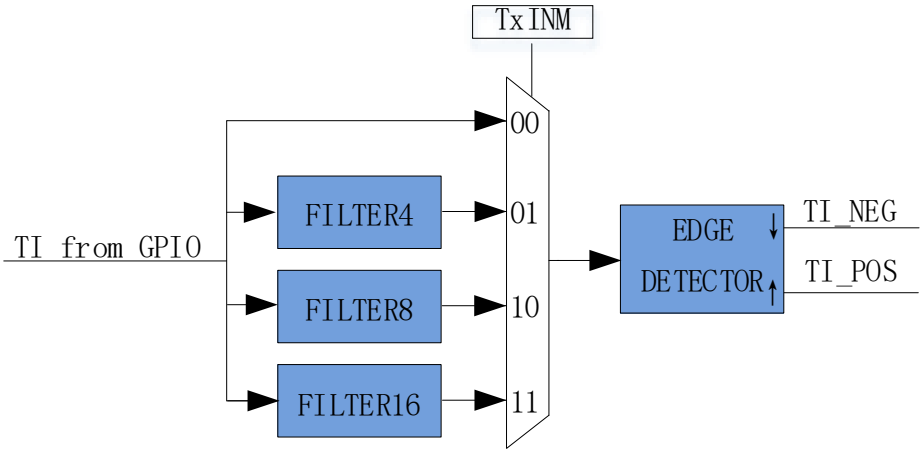


Figure 18-3 Block Diagram of Input Signal Filtering and Edge Detection

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

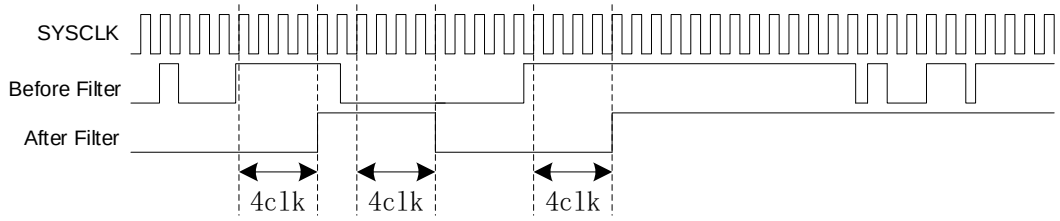


Figure 18-4 Timing Diagram of Filter Module

The edge detection module detects the filtered input signal from filtering module, and records the rising edge and falling edge for input capture mode.

18.1.5 Input Capture Mode

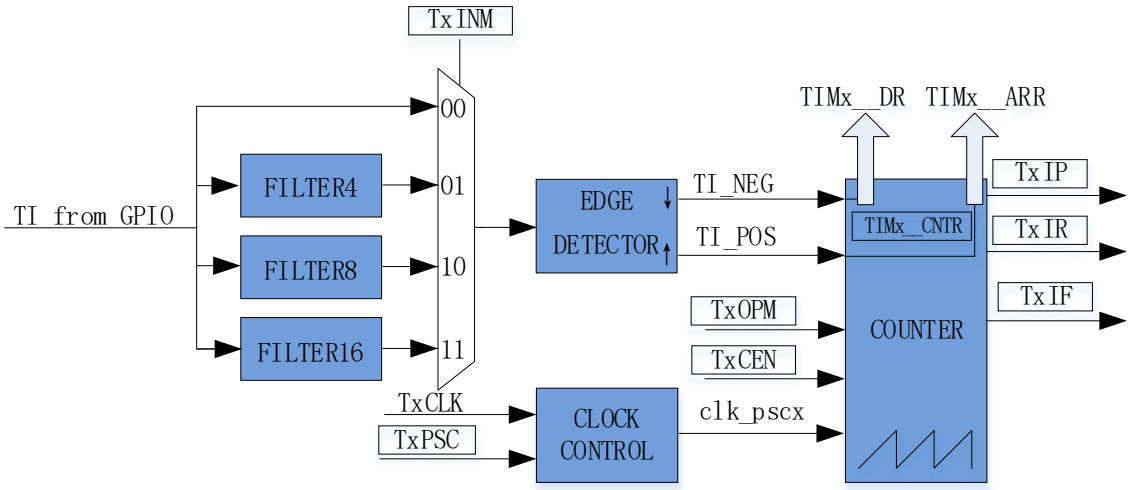


Figure 18-5 Schematic Diagram of Input Capture Mode

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.

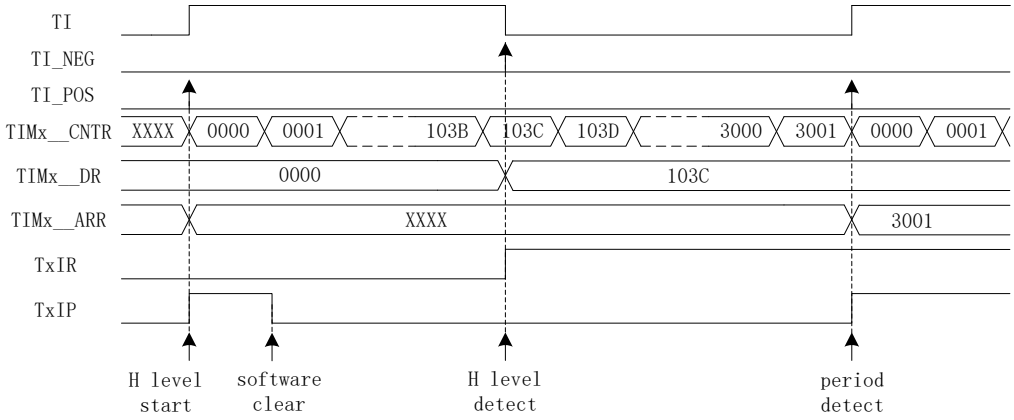


Figure 18-6 Timing Diagram of Input Capture Mode (TIMx_CR0[TxOCM] = 0)

For example, when TIMx_CR0[TxOCM]= 0, TIMx_CR1[TxEN] is set to “1” to enable the timer. The Base Timer is cleared to “0” and restarts when the first raising edge is detected. When the falling edge is detected, the value of TIMx__CNTR is stored into TIMx__DR. Meanwhile, the interrupt flag INT_SR1[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 INT_SR1[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. At this point, TIMx__ARR is 0xFFFF, and TIMx__DR is determined by the input level and TIMx_CR0[TxOCM] XOR.

18.1.6 Timer4 FG Generation Mode

FG signal is generated by FOC module and Timer4. FOC module calculates an FG result based on frequency base fbase. FGBASE is computed using the following algorithm: FGBASE=FBASE*32768*4/187500*X (X refers to frequency division and multiplication coefficient of FG signal, which can be a decimal;187500 is the pre-frequency-division of TIMER4). The software calculates the estimated speed OMEGA * FGBASE, and a low-order 24-bit result, or DELTA_THETA, is obtained after the 32-bit product shifting right by eight bits. Its high-order 16 bits are written to TIM4__ARR, and low-order 8 bits to TIM4__DR[15:8]. Configuring TIM4_CR0[T4FGM] = 1 enables Timer4 FG Generation mode. FG signals are output when the timer overflows.

18.2 Timer3/Timer4 Registers

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

Bit	7	6	5	4	3	2	1	0
Name	TxPSC			TxOCM	TxIRE	T4FGM	TxOPM	TxMOD
Type	R/W	R/W	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]	TxPSC	Base Timer Clock Prescaler Selection It is configured to divide the system clock frequency and generate the clock source for Base Timer. The prescaled clock rates are configured as follows: 000: 20MHz 001: 10MHz 010: 5MHz 011: 2.5MHz 100: 1.25MHz 101: 625kHz 110: 312.5kHz 111: 156.25kHz Note: In Input Capture Mode of Timer3, the clock rate is 40MHz when this bit is set to “111”.
[4]	TxOCM	Output Mode: Output Mode Selection 0: Output “0” when TIMx__CNTR < TIMx__DR; output “1” when TIMx__CNTR ≥ TIMx__DR 1: Output “1” when TIMx__CNTR < TIMx__DR; output “0” when TIMx__CNTR ≥ TIMx__DR Input Capture Mode: TIMx__DR indicates the input level to be selected when the active edge is detected or the timer becomes overflowed. 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).

		TIMx__DR indicates the input level to be selected when the timer becomes overflowed. 0: TIMx__DR is reset to “0” by hardware for low level input upon an overflow interrupt and is set to “0xFFFF” by hardware for high level input upon an overflow interrupt. 1: TIMx__DR is reset to “0” by hardware for high level input upon an overflow interrupt and is set to “0xFFFF” for low level input upon an overflow interrupt.
[3]	TxIRE	Output Mode: Compare Match Interrupt Enable Input Capture Mode: Pulse Width Detection Interrupt Enable 0: Disable 1: Enable Note: See INT_SR1 (0xF3) for Timerx IR Interrupt Flag.
[2]	T4FGM	Timer4 FG Mode Enable 0: Disable 1: Enable
[1]	TxOPM	Single Mode Base Timer stops in any of the following events: Output Mode: Base Timer overflow event Input Capture Mode: PWM Cycle Detection or Base Timer overflow event 0: Base Timer does not stop 1: Base Timer stops (TIMx_CR1[TxEN] is reset to “0”)
[0]	TxMOD	Working Mode Selection 0: Input Capture Mode 1: Output Mode

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

Bit	7	6	5	4	3	2	1	0
Name	RSV			TxIPE	TxIFE	TxINM		TxEN
Type	-	-	-	R/W	R/W	R/W	R/W	R/W
Reset	-	-	-	0	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	TxIPE	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Enable 0: Disable 1: Enable Note: See INT_SR1 (0xF3) for Timerx IP Interrupt Flag Bit.
[3]	TxIFE	Output Mode: Base Timer Overflow Interrupt Input Capture Mode: Base Timer Overflow Interrupt Enable Timer4 FG Generation Mode: FG Overflow Interrupt Enable 0: Disable 1: Enable Note: See INT_SR1 (0xF3) for Timerx IF Interrupt Flag Bit.
[2:1]	TxINM	Input Signal Filtering pulse width Selection Input signals are filtered as noise if pulse width is less than the defined value. 00: Not to filter signals 01: Filtered on every 4 SYSCLK cycles 10: Filtered on every 8 SYSCLK cycles 11: Filtered on every 16 SYSCLK cycles
[0]	TxEN	Base Timer Enable 0: Disable 1: Enable

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

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

Bit	Name	Description
[15:0]	TIMx__CNTR	Count values held in Base Timer

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

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

Bit	Name	Description
[15:0]	TIMx__DR	Output Mode: Compare match values (written by software) Timer4 FG Generation Mode: Low-order 8 bits of DELTA_THETA are written to the high-order 8 bits of this bit. Input Capture Mode: Count value of the detected input pulse width (written by hardware)

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

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

Bit	Name	Description
[15:0]	TIMx__ARR	Output Mode: Reload value (written by hardware) Timer4 FG Generation Mode: High-order 16 bits of DELTA_THETA Input Capture Mode: Count value of a detected PWM cycle (written by hardware)

19 Systick

19.1 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 SYST_CR[1:0] is not set to “0”, and the interrupt entry is accessed by 10.

19.2 Systick Registers

19.2.1 SYST_CR (0x4064)

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

Bit	Name	Description
[7:2]	RSV	Reserved
[1:0]	SYST_SEL	Systick Settings 00: Disable 01: 0.25ms 10: 0.5ms 11: 1ms Note: Systick Interrupt is automatically enabled when SYST_CR[SYST_SEL] is not set to “0”. See INT_SR2 (0xF4) for Systick Interrupt Flag.

20 Driver

20.1 Driver Instructions

20.1.1 FU6881Q1 Driver Introduction

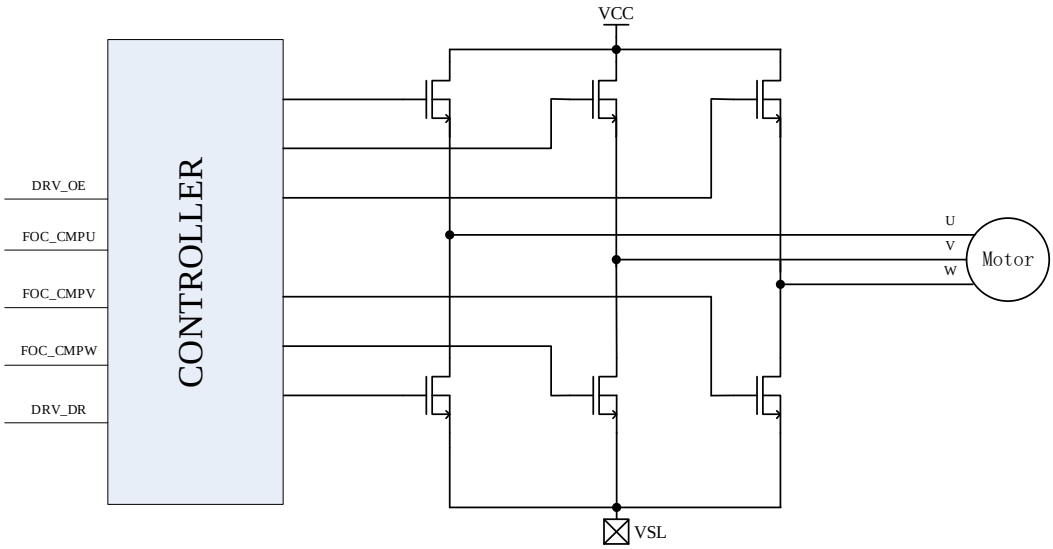


Figure 20-1 Block Diagram of FU6881Q1 Driver Module

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 six-way level signals to control the power MOS after passing through the output control module.

20.1.2 Output Control Module

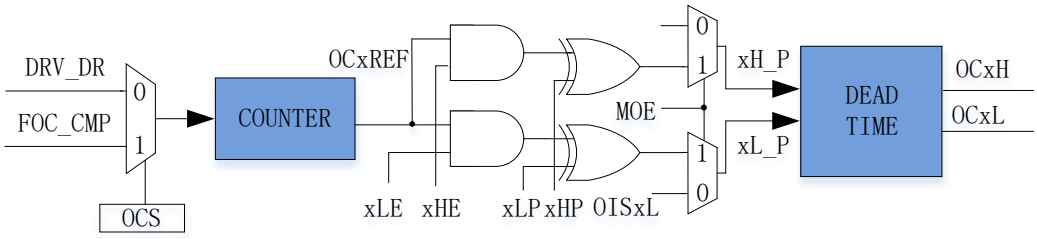


Figure 20-2 Block Diagram of Output Control Module

Before Driver module works, DRV_CR0[MESEL] is set to “1” to select FOC mode or to “0” to select square-wave control mode.

If DRV_CR0[OCS] = 0, comparison value of PWM comes from DRV_DR. Otherwise, it comes from FOC_CMP and U/V/W output signals (OCxREF) are generated. DRV_CMR[xHE], DRV_CMR[xLE], polarity control bits DRV_CMR[xHP] and DRV_CMR[xLP] are configured for logic processing of OCxREF signal. Enabling DRV_OUT[MOE] outputs PWM waveform, otherwise, the idle level. xH_P and xL_P output signals are transferred to the deadtime module to generate OcxH and OcxL signals for PWM drive signals.

20.1.2.1 Count and Compare Module

DRV_CR0[OCS] is configured to select the comparison value of PWM from FOC_CMPU/V/W of FOC module or DRV_DR set by software. The comparison value is sent to the counter for comparison to obtain the 3-phase original PWM signal OCxREF, and DRV_DR is used for motor pre-charging, braking and square-wave control. If DRV__CNTR is smaller than the comparison value, OCxREF outputs high-level signal, and if DRV__CNTR is larger than DRV_DR, OCxREF outputs low-level signal.

When DRV_CR0[OCS] = 1, FOC_CMPU/V/W is compared with the count value to generate the duty cycle OC1REF/OC2REF/OC3REF.

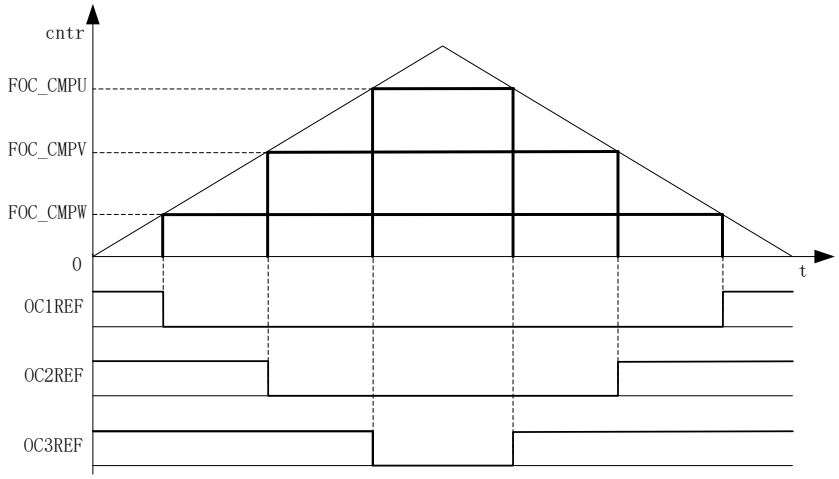


Figure 20-3 PMW Generation

When DRV_CR0[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%.

20.1.2.2 Enable and Polarity of Output Signals

DRV_CMR[xHE] and [xLE] are configured by software to enable high and low sides of the driver, 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_CR0[MESEL] = 0 enables the Square Wave Drive Mode. After Timer1 generates a write timing, the data stored in the corresponding TIM1_DBRx are transferred to the DRV_CMR register.

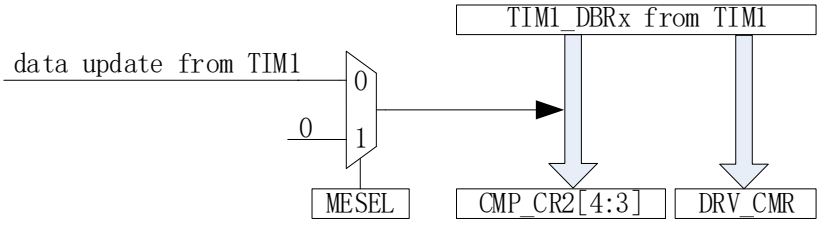


Figure 20-4 Timer1 Automatic Control of DRV_CMR and CMP_CR2[4:3]

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

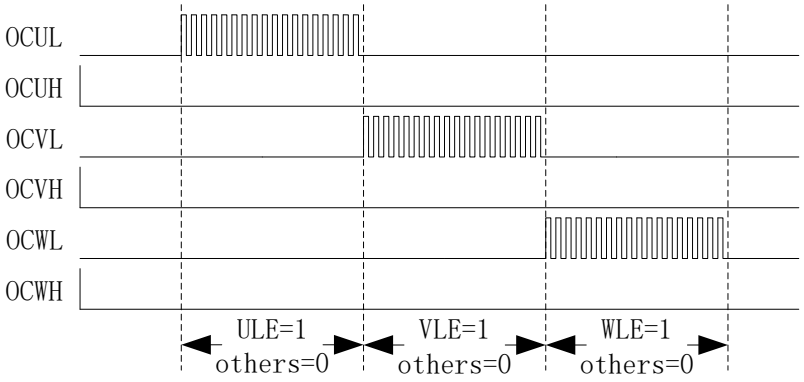


Figure 20-5 Pre-charge Waveform

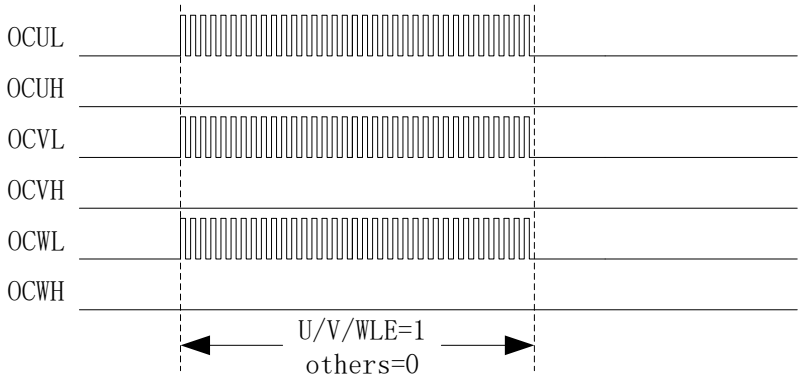


Figure 20-6 Brake Waveform

20.1.2.3 Interrupt

20.1.2.3.1 Compare Match Interrupt

The generation conditions and time for compare match interrupt are configured by DRV_CR1[DCIM] and DRV_COMR respectively. When the timer reaches the value set in DRV_COMR and the conditions set by DRV_CR1[DCIM] are met, a compare match interrupt is generated and the interrupt flag INT_SR2[DCIF] is set to “1” by hardware.

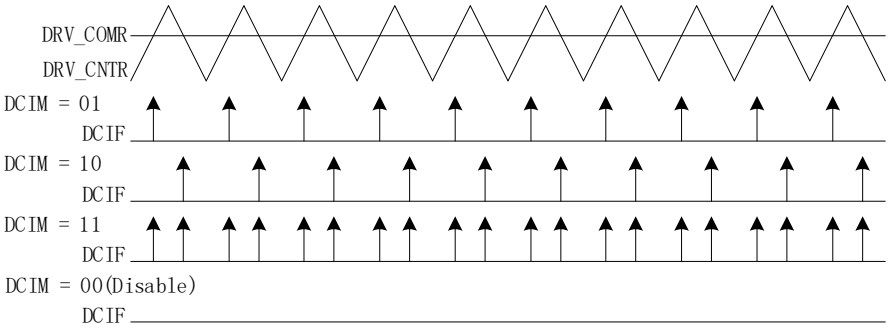


Figure 20-7 Driver Compare Match Interrupt

20.2 Driver Registers

20.2.1 DRV DTR (0x4060)

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

Bit	Name	Description
[7:1]	RSV	Reserved
[0]	DTSEL	Deadtime Selection 0: Disable 1: 4 clock cycles

20.2.2 DRV_CR0 (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) It is used to switch motor rotation directions, and valid in both square-wave drive and FOC drive modes. In sensorless FOC mode, setting this bit changes motor rotation. In 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

20.2.3 DRV_CR1 (0x4061)

Bit	7	6	5	4	3	2	1	0
Name	ANARDY	DRV_MD	RSV			DCIP	DCIM	
Type	R	R/W	-	-	-	R/W	R/W	R/W
Reset	0	0	-	-	-	0	0	0

Bit	Name	Description
[7]	ANARDY	Internal Circuit Ready Flag 0: Not ready 1: Ready
[6]	DRV_MD	Counting Mode 0: Center-aligned Mode 1: Sawtooth Wave Mode (FOC disabled)
[5:3]	RSV	Reserved
[2]	DCIP	Number of PWM cycles to generate a Compare Match Interrupt 0: 1 PWM cycle 1: 2 PWM cycles
[1:0]	DCIM	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. Note: See INT_SR2 (0xF4) for Driver Compare Match Interrupt Flag.

20.2.4 DRV_OUT (0xF8)

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

Bit	Name	Description
[7]	MOE	Main Output Enable This bit selects the sources for high and low sides of the driver of 3-phase output signals. It can be set to “1” and “0” by software. When bus current protection occurs (see section 29.1.1.1), it is automatically cleared to “0” to turn off the output. 0: Disabled, 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: Enabled, with output sourced from the comparison value of the timer.
[6]	RSV	Reserved
[5]	OISWL	Output idle level of WL See descriptions on OISUH register.

[4]	OISWH	Output idle level of WH 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

20.2.5 DRV_CMR (0x405C, 0x405D)

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

Bit	**Name**	**Description**
[15:12]	RSV	Reserved
[11]	WHP	High-side Polarity Control of Phase W 0: Active High 1: Active Low
[10]	WLP	Low-side Polarity Control of Phase W 0: Active High 1: Active Low
[9]	VHP	High-side Polarity Control of Phase-V 0: Active High 1: Active Low
[8]	VLP	Low-side Polarity Control of Phase-V 0: Active High 1: Active Low
[7]	UHP	High-side Polarity Control of Phase-U 0: Active High 1: Active Low
[6]	ULP	Low-side Polarity Control of Phase-U 0: Active High 1: Active Low
[5]	WHE	High-side Output Enable of Phase-W 0: Disable 1: Enable
[4]	WLE	Low-side Output Enable of Phase W 0: Disable 1: Enable
[3]	VHE	High-side Output Enable of Phase-V 0: Disable 1: Enable

[2]	VLE	Low-side Output Enable of Phase-V 0: Disable 1: Enable
[1]	UHE	High-side Output Enable of Phase-U 0: Disable 1: Enable
[0]	ULE	Low-side Output Enable of Phase-U 0: Disable 1: Enable

Notes:

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

20.2.6 DRV_ARR (0x405E, 0x405F)

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

Bit	**Name**	**Description**
[15:14]	RSV	Reserved
[13:0]	DRV_ARR	Timer reload value, which determines PWM frequency (center-aligned) Driver timer up-counts from 0 to DRV_ARR/2 - 1 and an overflow event occurs. Then it down-counts to 0. Calculation formula: $f_{carrier}$ = 40MHz/DRV_ARR DRV_ARR value is calculated using 40MHz clock rate, which falls within the range [0,16383]. Note: The LSB is always 0, and a write of "1" is meaningless.

20.2.7 DRV_COMR (0x405A, 0x405B)

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

Bit	**Name**	**Description**
[15:12]	RSV	Reserved

[11:0]	DRV_COMR	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. The clock rate for the calculation is 10MHz. Duty cycle at the match point = DRV_COMR*4/DRV_ARR*100% DRV_COMR value is calculated using 10MHz clock rate, which falls within the range [0, 4095].

20.2.8 DRV_DR (0x4058, 0x4059)

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

Bit	**Name**	**Description**
[15:14]	RSV	Reserved
[13:0]	DRV_DR	PWM Duty Cycle Setting in Software Duty cycle = DRV_DR/DRV_ARR*100% DRV_DR value is calculated using 40MHz clock rate, which falls within the range [0,16383]. Note: When this register is used as a comparison source, PWM is referenced to high side of the driver and a deadtime is inserted in the complementary output of the low side of driver.

20.2.9 DRV__CNTR (0x4066, 0x4067)

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

Bit	**Name**	**Description**
[15:12]	RSV	Reserved
[11:0]	DRV__CNTR	Count Value of Counter The clock rate for the calculation is 10MHz, and Driver duty cycle = DRV__CNTR*4/DRV_ARR*100% Range [0, 4095] Note: DRV__CNTR register is writable only when DRV_CR0[DRVEN] = 1.

21 WDT

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

After being enabled, WDT starts counting from 0. When reaching 0xFFFC, it 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 WDT rolls over to WDT_ARR and restarts counting.

21.1 WDT Notes

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

21.2 WDT Operations

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

21.3 WDT Registers

21.3.1 WDT_CR (0x4026)

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

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

21.3.2 WDT_ARR (0x4027)

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

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

21.3.3 CCFG1 (0x401E)

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

Bit	Name	Description
[7]	LVW_EN_B	VCC Low Voltage Warning 0: Disable 1: Enable
[6]	LVWIE	LVW Detection Interrupt Enable 0: Disable 1: Enable Note: See INT_SR3 (0xF5) for LVW Interrupt Flag.
[5]	WDT_EN	WDT Enable 0: Disable 1: Enable
[4:0]	RSV	Reserved

22 RTC

22.1 RTC Functional Block Diagram

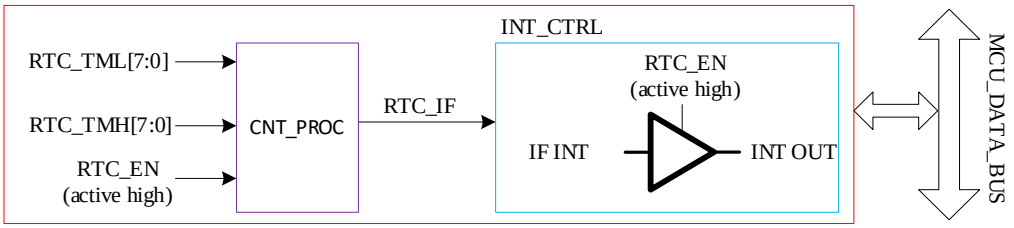


Figure 22-1 RTC Functional Block Diagram

22.2 RTC Operations

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

22.3 RTC Registers

22.3.1 RTC_TM (0x402C, 0x402D)

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

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

22.3.2 RTC_STA (0x402E)

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

Bit	**Name**	**Description**
[7]	RTC_EN	RTC Enable 0: Disable 1: Enable
[6:5]	RSV	Reserved

[4]	ISOSCEN	Internal Slow Clock Enable 0: Disable 1: Enable
[3:0]	RSV	Reserved

22.4 Clock Calibration

22.4.1 Introduction

Clock calibration is a feature that uses internal slow clock to calibrate the internal fast clock. 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.

22.4.2 Clock Calibration Registers

22.4.2.1 CAL_CR0 (0x4044)/CAL_CR1 (0x4045)

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

Bit	**Name**	**Description**
[15]	CAL_STA/ CAL_BUSY	Clock Calibration Enable Read: 0: Calibration is completed. 1: Calibration is in progress. Write: 0: No effect 1: Clock Calibration starts.
[14:13]	RSV	Reserved
[12:0]	CAL_ARR	Calibration Counts The count values of the fast clock to continuously count eight slow clock cycles Note: 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).

23 IO

23.1 IO Introduction

FU6881Q1 has up to 16 GPIO pins, including P0.0 ~ P0.7 and P1.0 ~ P1.7.

23.2 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.

- The enable bits of pull-up resistors and pull-down resistors are configured to "1". See 23.3.4 P0_PU (0x4053) ~ 23.3.5 P1_PU (0x4054) for port pins and registers.
- See section 5.3 GPIO Electrical Characteristics for the values of pull-up resistors and pull-down resistors.
- The relevant bits of P1_AN register are configured to "1". See 23.3.3 P1_AN (0x4050) 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 register.

23.3 IO Registers

23.3.1 P0_OE (0xFC)

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

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

23.3.2 P1_OE (0xFD)

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

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

23.3.3 P1_AN (0x4050)

Bit	7	6	5	4	3	2	1	0
Name	RSV	P1_AN			HBMOD	RSV	ODE1	ODE0
Type	-	R/W	R/W	R/W	R/W	-	R/W	R/W
Reset	-	0	0	0	0	-	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6:4]	P1_AN	P1.4 ~ P1.6 Analog Mode Enable 0: Disable 1: Enable
[3]	HBMOD	P1.3 mode configuration, which determines the functional mode of P1.3 in combination with P1_OE[3], as shown in Table 23-1. Table 23-1 P1.3 Mode Setting

HBMOD	P1_OE[3]	P1.3 Pin Mode
0	0	Digital Input
0	1	Digital Output
1	0	Analog Mode
1	1	Digital enhanced drive output mode. The maximum output current of high level output can be up to 20mA for Hall bias power supply. The drive mode of low level output is the same as that of the digital output mode.

[2]	RSV	Reserved
[1]	ODE1	P0.3 Open Drain Enable 0: Disable 1: Enable
[0]	ODE0	P0.2 Open Drain Enable 0: Disable 1: Enable

23.3.4 P0_PU (0x4053)

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

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

23.3.5 P1_PU (0x4054)

Bit	7	6	5	4	3	2	1	0
Name	P1_PU[7:2]						P11HV_EN	P10HV_EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:2]	P1_PU[7:2]	P1.2 ~ P1.7 Pull-up Resistor Enable 0: Disable 1: Enable

[1]	P11HV_EN	P1.1 High Voltage Input Enable 0: Disable. The reversed level is the same as that of general-purposed IO. See section 5.3 GPIO Electrical Characteristics. 1: Enable. The reversed level is related to VCC. See section 5.3 GPIO Electrical Characteristics. Note: P1.1 can withstand a voltage up to VCC regardless of the level of this bit.
[0]	P10HV_EN	P1.0 High Voltage Input Enable 0: Disable 1: Enable

23.3.6 Px_PL (0x4052)

Bit	7	6	5	4	3	2	1	0
Name	P17_PL	P01_PL	RSV					
Type	R/W	R/W	-	-	-	-	-	-
Reset	0	0	-	-	-	-	-	-

Bit	Name	Description
[7]	P17_PL	P1.7 Pull-down Resistor Enable 0: Disable 1: Enable
[6]	P01_PL	P0.1 Pull-down Resistor Enable 0: Disable 1: Enable
[5:0]	RSV	Reserved

23.3.7 PH_SEL (0x404C)

Bit	7	6	5	4	3	2	1	0
Name	SPITMOD	UARTEN	RSV	T4SEL	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	Port multiplexed as RXD, TXD and UART enabled 0: Disable 1: P0.0 and P0.1, P0.2 and P0.3, P1.2 and P1.7 pins multiplexed as RXD and TXD pins respectively and UART enabled
[5]	RSV	Reserved
[4]	T4SEL	Port pins multiplexed as Timer4 or Timer4S 0: Disable 1: P0.1 or P0.0 multiplexed as Timer4 I/O pins.
[3]	T3SEL	Port pins multiplexed as Timer3 or Timer3S 0: Disable 1: P1.1 or P0.1 pin multiplexed as Timer3 I/O pins
[2]	T2SEL	Port pins multiplexed as Timer2 0: Disable 1: P1.0 pin multiplexed as Timer2 I/O pins
[1]	T2SSEL	Port pins multiplexed as Timer2S 0: Disable 1: P0.7 pin multiplexed as Timer2 I/O pins
[0]	RSV	Reserved

23.3.8 PH_SEL1 (0x404D)

Bit	7	6	5	4	3	2	1	0
Name	UARTCH1	UARTCH0	RSV			T4CT	RSV	T3CT
Type	R/W	R/W	-	-	-	R/W	-	R/W
Reset	0	0	-	-	-	0	-	0

Bit	Name	Description
[7:6]	UARTCH	UART Function Switching 00: P0.0 serving as RXD, and P0.1 pin as TXD (P0.0 is an I/O pin of single-wire mode) 01: P0.2 serving as RXD, and P0.3 pin as TXD (P0.2 is an I/O pin of single-wire mode) 1X: P1.2 serving as TXD, and P1.7 pin as RXD (P1.2 is an I/O pin in single-wire mode)
[5:3]	RSV	Reserved
[2]	T4CT	Timer4 Function Switching 0: Timer4 I/O pin switched to P0.1 1: Timer4 I/O pin switched to P0.0
[1]	RSV	Reserved
[0]	T3CT	Timer3 Function Switching 0: Timer3 I/O pin switched to P1.1 1: Timer3 I/O pin switched to P0.1

23.3.9 P0 (0x80)

Port output register P0/1 supports read and write access. RMW commands are used to access the register value (see Table 23-2 for RMW commands), and other commands are used to access PORT pin.

Bit	7	6	5	4	3	2	1	0
Name	GP07	GP06	GP05	GP04	GP03	GP02	GP01	GP00
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP07	GP07
[6]	GP06	GP06
[5]	GP05	GP05
[4]	GP04	GP04
[3]	GP03	GP03
[2]	GP02	GP02
[1]	GP01	GP01
[0]	GP00	GP00

23.3.10 P1 (0x90)

Bit	7	6	5	4	3	2	1	0
Name	GP17	GP16	GP15	GP14	GP13	GP12	GP11	GP10
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP17	GP17
[6]	GP16	GP16
[5]	GP15	GP15
[4]	GP14	GP14
[3]	GP13	GP13
[2]	GP12	GP12
[1]	GP11	GP11
[0]	GP10	GP10

Table 23-2 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”

24 Temperature Sensor

FU6881Q1 integrates temperature sensor. TSD_CR[TSEN_EN] is configured as "1" to enable temperature sensor, read the value in register, and view the table to obtain current internal temperature of the chip. The sensed temperature ranges from -40°C ~150°C, with its mapping code shown in Table 24-1.

Table 24-1 Mapping between Output Codes and Temperature Levels

TSEN_DR	Temperature (℃)	TSEN_DR	Temperature (℃)
0	≤ -40	1000000	67
1	-39	1000001	68
10	-38	1000010	69
11	-37	1000011	70
100	-36	1000100	71
101	-35	1000101	72
110	-34	1000110	74
111	-33	1000111	75
1000	-32	1001000	76
1001	-31	1001001	77
1010	-30	1001010	78
1011	-29	1001011	79
1100	-28	1001100	81
1101	-26	1001101	82
1110	-25	1001110	83
1111	-24	1001111	84
10000	-23	1010000	85
10001	-22	1010001	87
10010	-20	1010010	88
10011	-19	1010011	89
10100	-18	1010100	90
10101	-16	1010101	92
10110	-15	1010110	93
10111	-14	1010111	94
11000	-12	1011000	96
11001	-11	1011001	97
11010	-9	1011010	98
11011	-8	1011011	100
11100	-6	1011100	101
11101	-5	1011101	102
11110	-3	1011110	104
11111	-2	1011111	105
100000	0	1100000	106
100001	1	1100001	108
100010	3	1100010	109
100011	5	1100011	111
100100	6	1100100	112
100101	8	1100101	114
100110	10	1100110	115
100111	12	1100111	117
101000	14	1101000	118

TSEN_DR	Temperature (℃)	TSEN_DR	Temperature (℃)
101001	15	1101001	119
101010	17	1101010	121
101011	19	1101011	123
101100	21	1101100	124
101101	23	1101101	126
101110	25	1101110	127
101111	27	1101111	129
110000	29	1110000	130
110001	32	1110001	132
110010	34	1110010	133
110011	36	1110011	135
110100	38	1110100	137
110101	40	1110101	138
110110	43	1110110	140
110111	45	1110111	142
111000	48	1111000	143
111001	50	1111001	145
111010	53	1111010	147
111011	55	1111011	148
111100	58	1111100	150
111101	60	1111101	152
111110	63	1111110	153
111111	66	1111111	≥ 155

24.1 Temperature Sensor Registers

24.1.1 TSD_CR (0x402F)

Bit	7	6	5	4	3	2	1	0
Name	TSDEN	TSEN_EN	TSEN_HYS	RSV	TSDADJ			
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]	TSDEN	TSD Feature Enable 0: Disable 1: Enable
[6]	TSEN_EN	Temperature Sensor Enable 0: Disable 1: Enable
[5]	TSEN_HYS	Filter width of temperature sensor output 0: Filter width is 1 code value 1: Filter width is 2 code value
[4]	RSV	Reserved
[3:0]	TSDADJ	TSD Temperature (measured junction temperature of the chip) 0000: 72℃ 0001: 77℃ 0010: 82℃ 0011: 87℃ 0100: 92℃ 0101: 97℃ 0110: 102℃ 0111: 108℃ 1000: 114℃ 1001: 120℃ 1010: 126℃ 1011: 132℃ 1100: 139℃ 1101: 146℃ 1110: 152℃ 1111: 159℃

24.1.2 TSEN_DR (0x4048)

Bit	7	6	5	4	3	2	1	0
Name	RSV	TSEN_DR						
Type	-	R	R	R	R	R	R	R
Reset	-	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6:0]	TSEN_DR	Data of Temperature Sensor Output Read the register value, and find out the temperature of the chip according to Table 24-1.

25 ADC

25.1 ADC Introduction

The ADC module is a 12-bit successive approximation register ADC with 11 channels inside. The sampling mode supports sequential sampling (that is, from ADC Channel 0 to ADC channel 11 in sequence) and triggered sampling (including FOC triggered sampling mode, Timer1 triggered sampling mode and LIN auto-addressing triggered sampling mode). The results of sequential sampling are stored in ADCx_DR (x = 0 ~ 11) in a right-aligned or left-second-high-aligned format. The result of triggered sampling is sent to FOC module or Timer1 module, instead of ADCx_DR, for motor control or LIN module for auto-addressing. The relevant registers of the FOC module, Timer1 module or LIN module are always left-second-high-aligned to store the triggered sample results. Triggered sampling is done automatically by hardware, and sequential sampling is controlled by software. The priority of triggered sampling is higher than that of sequential sampling. If both triggered sampling and sequential sampling are applied at the same time, the triggered sampling is performed first, and then sequential sampling mode.

The clock source of ADC sampling is at a rate of 10MHz, and the sampling time is set by ADC_SCYC1 and ADC_SCYC2. See ADC Electrical Characteristics for sampling time and conversion time.

25.2 ADC Block Diagram

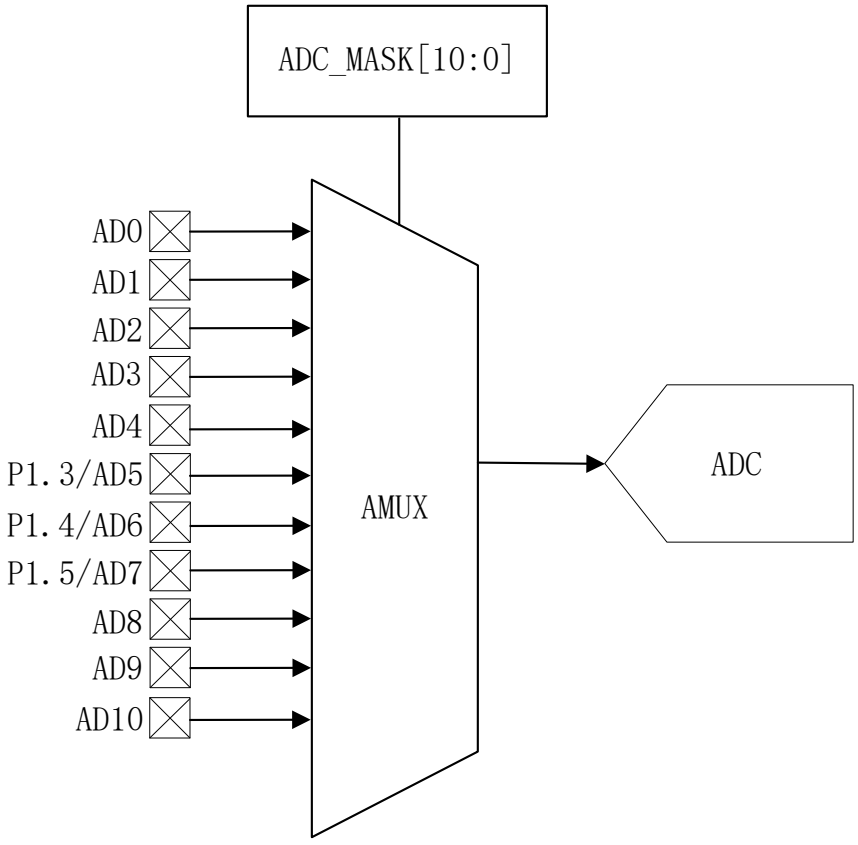


Figure 25-1 ADC Multiplexer Block Diagram

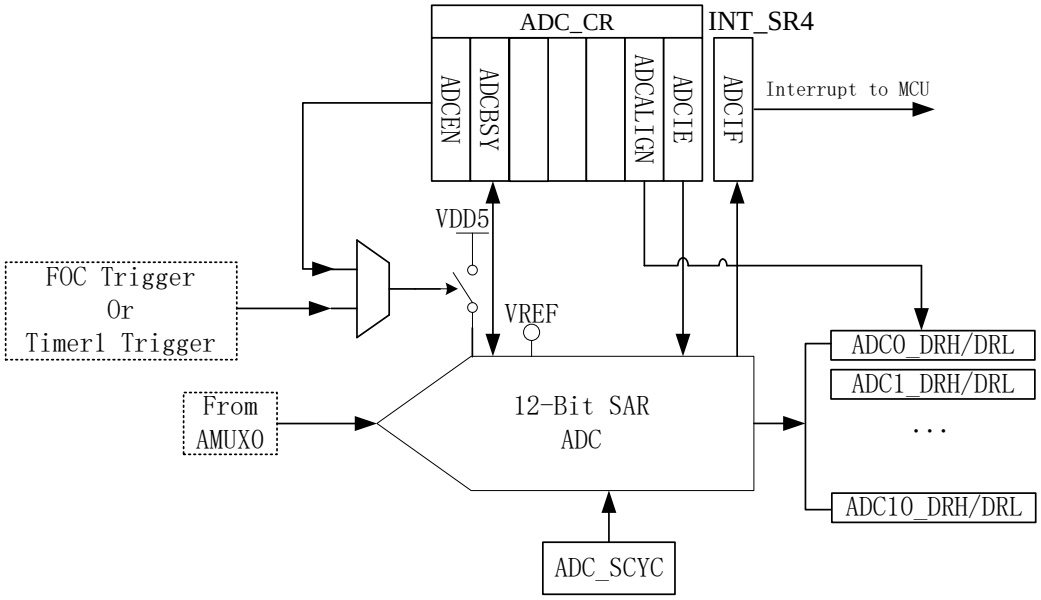


Figure 25-2 ADC Functional Block Diagram

25.3 ADC Operations

25.3.1 Sequential Sampling Mode

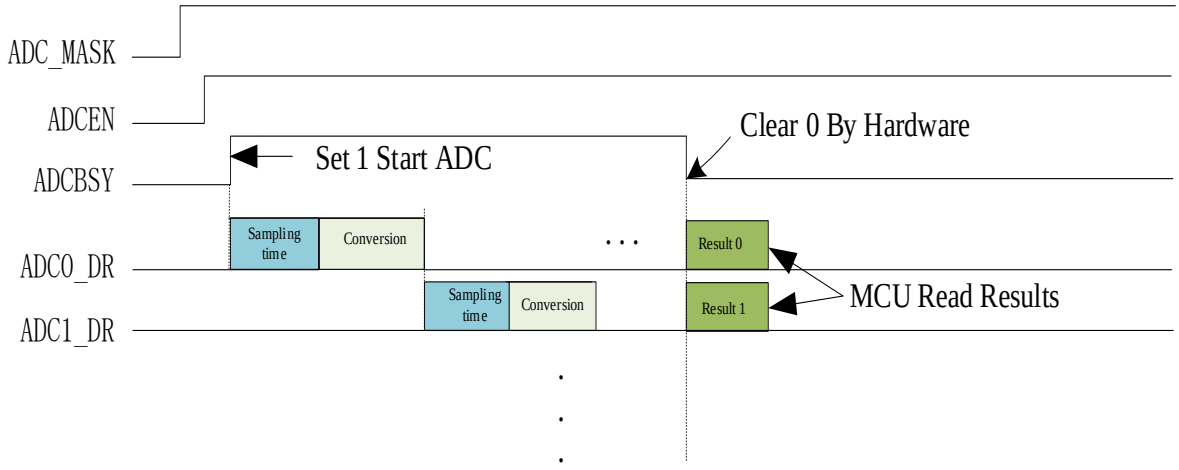


Figure 25-3 ADC Sequential Sampling Timing

ADC operations:

1. Set the appropriate ADC VREF;
2. Configure ADC_MASK to enable the corresponding channel required to sample;
3. Configure ADC_SCYC (minimum value is 3) to select the sampling period of each channel;
4. Configure ADC_CR[ADCEN] = 1 to enable ADC;
5. Configure ADC_CR[ADCBSY] = 1 to start ADC;
6. When ADC_CR[ADCBSY] = 0, ADC results are read by ADCx_DR.

Note: The ADC conversion sequence is from low to high 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).

25.3.2 Triggered Sampling Mode

When FOC module is enabled, ADC channel 0/1/2/4 can be used to FOC triggered sampling. Channel 2 is selected by FOC_CR0[UCSEL] for bus voltage triggered sampling. In single-shunt current sampling mode, channel 4 is used for itrip sampling. In dual-shunt current sampling mode, channel 0 is used for ia sampling and channel 1 for ib sampling. In triple-shunt current sampling mode, channel 0 is used for is ia sampling, channel 1 for ib sampling, and channel 4 for ic sampling.

When Timer1 is enabled, channel 4 is used for bus current sampling. The TIM1_CR3[T1TIS] is configured to select ADC as the input source of position detection. Channel 10 is used for U-phase voltage sampling, channel 9 for V-phase voltage sampling, and channel 8 for W-phase voltage sampling.

25.3.3 Output Data Format

Registers ADCx_DRH and ADCx_DRL contain the high-order bits and the low-order bits of ADC sampling results. Data can be right-aligned or left-second-high-aligned by configuring ADC_CR[ADCALIGN]. When input voltage ranges from 0 to VREF, the relation between the input voltage and result data is shown in Table 25-1. The bits, which are not used in ADCx_DRH and ADCx_DRL, are set to “0”.

Table 25-1 Relation between Output Voltage and Result Data

Input Voltage	Right-aligned	Left-second-high-aligned
0	0x0000	0x0000
VREF/2	0x0800	0x4000
VREF	0x0FFF	0x7FF8

25.4 ADC Registers

25.4.1 ADC_CR (0x4039)

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

Bit	Name	Description
[7]	ADCEN	ADC Enable 0: Disable 1: Enable
[6]	ADCBSY	ADC Start & ADC Busy Flag Read: 0: ADC Idle 1: ADC Busy Write: 0: No effect 1: ADC conversion starts Note: Writing “1” 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, and ADC result = ADCx_DR[11:0] 1: ADC output is left-second-high-aligned, and ADC result = ADCx_DR[14:3] Note: The results of triggered sampling mode are always left-second-high-aligned.
[1]	ADCIE	ADC Interrupt Enable (excluding triggered sampling mode interrupt) 0: Disable 1: Enable Note: See INT_SR4 (0xF6) for ADC Interrupt Flag.
[0]	RSV	Reserved

25.4.2 ADC_MASK (0x4036, 0x4037)

ADC_MASKH(0x4036)								
Bit	15	14	13	12	11	10	9	8
Name	RSV					CH10EN	CH9EN	CH8EN
Type	-	-	-	-	-	R/W	R/W	R/W
Reset	-	-	-	-	-	0	0	0
ADC_MASKL(0x4037)								
Bit	7	6	5	4	3	2	1	0
Name	CH7EN	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:11]	RSV	Reserved
[10]	CH10EN	ADC channel 10 Enable
[9]	CH9EN	ADC channel 9 Enable
[8]	CH8EN	ADC channel 8 Enable
[7]	CH7EN	ADC channel 7 Enable
[6]	CH6EN	ADC channel 6 Enable
[5]	CH5EN	ADC channel 5 Enable

[4]	CH4EN	ADC channel 4 Enable
[3]	CH3EN	ADC channel 3 Enable
[2]	CH2EN	ADC channel 2 Enable
[1]	CH1EN	ADC channel 1 Enable
[0]	CH0EN	ADC channel 0 Enable

Note: In triggered sampling mode, it is not required to configure ADC_MASK.

25.4.3 ADC_SCYC1 (0x4035)

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

Bit	Name	Description
[7:4]	RSV	Reserved
[3:2]	ADC_SCYCH[1:0]	ADC sampling cycle for ADC channel 8 ~ 10 ADC_SCYCH[1] = 0: The sampling cycle is (ADC_SCYC[9:8]*16 + 12) ADC clock cycles. ADC_SCYCH[1] = 1: The sampling cycle is (ADC_SCYC[9:8]*16) ADC clock cycles.
[1:0]	RSV	Reserved

25.4.4 ADC_SCYC2 (0x4038)

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

Bit	Name	Description
[7:4]	ADC_SCYC[7:4]	ADC sampling cycle for ADC channel 5 ~ 7 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 ~ 4 The sampling cycle is ADC_SCYC[3:0] ADC clock cycles.

25.4.5 ADC0_DR (0x0FD8, 0x0FD9)

ADC0_DRH(0x0FD8)								
Bit	15	14	13	12	11	10	9	8
Name	ADC0_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC0_DRL(0x0FD9)								
Bit	7	6	5	4	3	2	1	0
Name	ADC0_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC0_DR	The conversion results of ADC channel 0 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.6 ADC1_DR (0x0FDA, 0x0FDB)

ADC1_DRH(0x0FDA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC1_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC1_DRL(0x0FDB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC1_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC1_DR	The conversion results of ADC channel 1 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.7 ADC2_DR (0x0FDC, 0x0FDD)

ADC2_DRH(0x0FDC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC2_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC2_DRL(0x0FDD)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC2_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC2_DR	The conversion results of ADC channel 2 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.8 ADC3_DR (0x0FDE, 0x0FDF)

ADC3_DRH(0x0FDE)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC3_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC3_DRL(0x0FDF)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Name	ADC3_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC3_DR	The conversion results of ADC channel 3 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.9 ADC4_DR (0x0FE0, 0x0FE1)

ADC4_DRH(0x0FE0)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC4_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC4_DRL(0x0FE1)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC4_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC4_DR	The conversion results of ADC channel 4 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.10 ADC5_DR (0x0FE2, 0x0FE3)

ADC5_DRH(0x0FE2)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC5_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC5_DRL(0x0FE3)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC5_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC5_DR	The conversion results of ADC channel 5 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.11 ADC6_DR (0x0FE4, 0x0FE5)

ADC6_DRH(0x0FE4)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC6_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC6_DRL(0x0FE5)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC6_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	ADC6_DR	The conversion results of ADC channel 6 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.12 ADC7_DR (0x0FE6, 0x0FE7)

ADC7_DRH(0x0FE6)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC7_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC7_DRL(0x0FE7)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC7_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Description**
[15:0]	ADC7_DR	The conversion results of ADC channel 7 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.13 ADC8_DR (0x0FE8, 0x0FE9)

ADC8_DRH(0x0FE8)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC8_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC8_DRL(0x0FE9)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC8_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC8_DR	The conversion results of ADC channel 8 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.14 ADC9_DR (0x0FEA, 0x0FEB)

ADC9_DRH(0x0FEA)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC9_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC9_DRL(0x0FEB)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC9_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC9_DR	The conversion results of ADC channel 9 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.15 ADC10_DR (0x0FEC, 0x0FED)

ADC10_DRH(0x0FC)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	ADC10_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC10_DRL(0x0FED)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADC10_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC10_DR	The conversion results of ADC channel 10 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

26 DMA

26.1 DMA Instructions

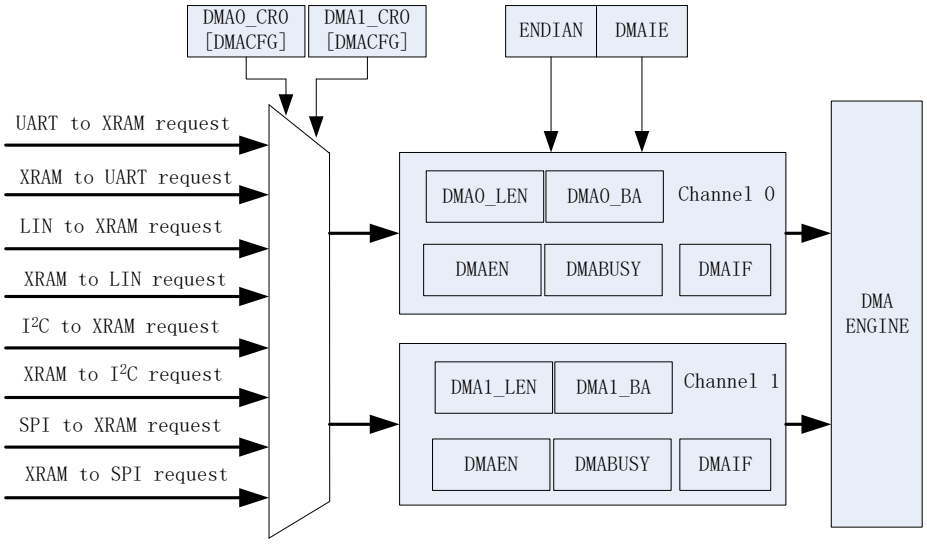


Figure 26-1 DMA Functional Block Diagram

The DMA module is a dual-channel DMA controller, which performs direct data transfer between peripherals (SPI, UART, I²C, LIN) 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 feature can be enabled.

DMA operations are as follows:

1. Configure the peripheral and enable the peripheral, and set input and output channels taken over by DMA by DMAx_CR0[DMACFG];
2. Configure DMA interrupt enable, transfer order, transfer length and XRAM start address. Write “1” to DMAx_CR0[DMAEN] and DMAx_CR0[DMABSY] to start DMA;
3. After data transfer, the interrupt flag bit INT_SR4[DMAxIF] is set to “1” by hardware and it is cleared to “0” by software;
4. Set DMAx_CR0[DMABSY] to “1” to start DMA again.

26.2 DMA Registers

26.2.1 DMA0_CR0 (0x403A)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DMA0IE	ENDIAN	RSV
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	-
Reset	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 Idle 1: Channel 0 Busy Write: 0: No effect 1: Channel 0 starts for data transfer
[5:3]	DMACFG	DMA Channel 0 Peripherals and Transfer Direction Selection 000: From UART to XRAM 001: From XRAM to UART 010: From I^2C to XRAM 011: From XRAM to I^2C 100: From SPI to XRAM 101: From XRAM to SPI 110: From LIN to XRAM 111: From XRAM to LIN Note: It cannot be configured when Channel 0 is busy. The read/write direction of LIN module is determined by LIN_CR[LINRW].
[2]	DMA0IE	DMA Channel 0 Interrupt Enable 0: Disable 1: Enable Note: See INT_SR4 (0xF6) for ADC Interrupt Flag.
[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]	RSV	Reserved

26.2.2 DMA1_CR0 (0x403B)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DBGSW	DBGEN	DMA1IE
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 Read: 0: Channel 1 Idle 1: Channel 1 Busy

		Write: 0: No effect 1: Channel 1 starts for data transfer
[5:3]	DMACFG	DMA Channel 1 Peripherals and Direction Selection 000: From UART to XRAM 001: From XRAM to UART 010: From I^2C to XRAM 011: From XRAM to I^2C 100: From SPI to XRAM 101: From XRAM to SPI 110: From LIN to XRAM 111: From XRAM to LIN Note: It cannot configured when Channel 1 is busy. The read/write direction of LIN module is determined by LIN_CR[LINRW].
[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 Enable DMA works in Debug mode when DMA1_CR0[DMACFG] is set to “101” and DMA1_CR0[DBGEN] to “1”. After SPI is enabled, DMA automatically sends relevant data in the sector defined by DMA1_CR0[DBGSW] via MOSI. DMA1_BA/DMA1_LEN defines the start address and range of the relevant data. 0: Disable 1: Enable Note: DMA Channel 1 Interrupt is automatically disabled in Debug mode.
[0]	DMA0IE	DMA Channel 1 Interrupt Enable 0: Disable 1: Enable Note: See INT_SR4 (0xF6) for ADC Interrupt Flag.

26.2.3 DMA0_LEN (0x403C)

Bit	7	6	5	4	3	2	1	0
Name	RSV		DMA0_LEN					
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	DMA0_LEN	Transfer Length of DMA Channel 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 DMA0_LEN be set to an odd number.

26.2.4 DMA0_BA (0x403E, 0x403F)

DMA0_BAH(0x403E)								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name	RSV				DMA0_BA[11:8]			
Type	-	-	-	-	R/W	R/W	R/W	R/W
Reset	-	-	-	-	0	0	0	0
DMA0_BAL(0x403F)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Name	DMA0_BA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:0]	DMA0_BA	Start address of data transfer by DMA Channel 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: DMA0_BA[11:0] ~ (DMA0_BA[11:0] + DMA0_LEN[5:0])

26.2.5 DMA1_LEN (0x403D)

Bit	7	6	5	4	3	2	1	0
Name	RSV		DMA1_LEN					
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	DMA1_LEN	Transfer length of DMA Channel 1 Read: The number of the bytes 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 DMA0_CR0[ENDIAN] = 1 (low bytes are received or transmitted first), it is recommended that DMA1_LEN be set to an odd number.

26.2.6 DMA1_BA (0x4040, 0x4041)

DMA1_BAH(0x4040)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DMA1_BA[11:8]			
Type	-	-	-	-	R/W	R/W	R/W	R/W
Reset	-	-	-	-	0	0	0	0
DMA1_BAL(0x4041)								
Bit	7	6	5	4	3	2	1	0
Name	DMA1_BA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:12]	RSV	Reserved
[11:0]	DMA1_BA	Start address of data transfer by DMA Channel 1 Start address of XRAM data transfer by DMA Channel 1 It cannot be configured when Channel 1 is busy. Note: XRAM address space of Channel 1 data transfer: DMA1_BA[11:0] ~ (DMA1_BA[11:0] + DMA1_LEN[5:0])

Note: When I²C is selected as DMA channel peripherals (including from I²C to XRAM and from XRAM to I²C) , START + Address interrupt of I²C communication still requires to be cleared to “0” by MCU software. In I²C slave mode, if STOP is received, I2C_SR[I2CSTP] = 0 is configured to clear I²C interrupt and restart the DMA transfer.

27 VREF

27.1 VREF Instructions

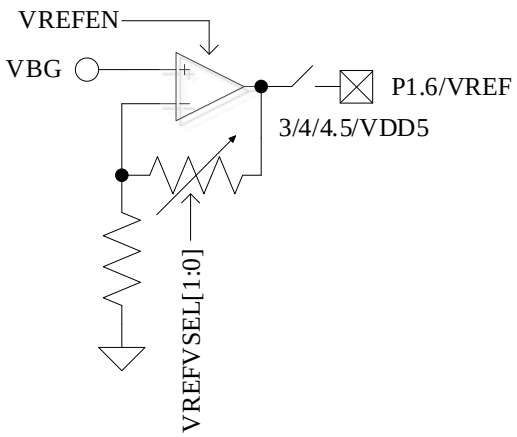


Figure 27-1 I/O Pins of VREF Module

The input and output ports of the VREF module are shown in Figure 27-1. VREF is the voltage reference generation block that provides internal voltage reference to ADC and DAC module. VBG is the voltage supplied by the chip internally.

VREF is enabled when VREF_CR[VREFEN] is set to “1”. The output voltage is selected by configuring VREF_CR[VREFVSEL]. When P1_AN[6] = 1 and P1_OE[6] = 1, VREF is output to P1.6 pin.

27.2 VREF Register

27.2.1 VREF_CR (0x404F)

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

Bit	Name	Description
[7:6]	VREFVSEL	VREF Module Output Voltage Selection 00: 4.5V 01: VDD5 10: 3V 11: 4V
[5]	RSV	Reserved
[4]	VREFEN	VREF Module Enable Bit 0: Disable. P1_AN[6] is set to “1”, and external VREF is input from P1.6 pin 1: Enable. P1_AN[6] is set to “1”, and internal VREF is output to P1.6 pin. A 1μF ~ 4.7μF external capacitor is added to improve the stability of VREF.
[3:0]	RSV	Reserved

28 Current Sampling

28.1 Current Sampling Instructions

The chip integrates current sampling module to sample and amplify U/V/W-phase current and bus current. The sampled current is sent to ADC module for conversion. When AMP_CR0[AMP1EN] = 1, current sampling module samples U-phase current. When AMP_CR0[AMP2EN] = 1, current sampling module samples V-phase current. When AMP_CR0[AMP3EN] = 1, current sampling module samples W-phase current. When AMP_CR0[AMP4EN] = 1, current sampling module samples bus current. The amplification gain is configured by AMP_CR1 register. CALOE = 1 shall be configured for initialization after current sampling is enabled (AMPxEN = 1).

28.2 Current Sampling Registers

28.2.1 AMP_CR0 (0x404E)

Bit	7	6	5	4	3	2	1	0
Name	CALOE	RSV			AMP3EN	AMP2EN	AMP1EN	AMP4EN
Type	R/W	-	-	-	R/W	R/W	R/W	R/W
Reset	0	-	-	-	0	0	0	0

Bit	Name	Description
[7]	CALOE	Current Sampling Initialization 0: Normal operation 1: Current sampling initialization starts
[6:4]	RSV	Reserved
[3]	AMP3EN	W-phase Current Sampling Enable 0: Disable 1: Enable
[2]	AMP2EN	V-phase Current Sampling Enable 0: Disable 1: Enable
[1]	AMP1EN	U-phase Current Sampling Enable 0: Disable 1: Enable
[0]	AMP4EN	Bus Current Sampling Enable 0: Disable 1: Enable

28.2.2 AMP_CR1 (0x4034)

Bit	7	6	5	4	3	2	1	0
Name	IBUS_GAIN			AMP4_ISCALE		RSV	AMP_GAIN	
Type	R/W	R/W	R/W	-	-	-	R/W	R/W
Reset	0	0	0	-	-	-	0	0

Bit	Name	Description
[7:5]	IBUS_GAIN	Amplification Gain for Bus Current See descriptions on AMP_CR1[AMP_GAIN]. AMP_CR1 (0x4034)
[4:3]	AMP4_ISCALE	Amplification Gain for FOC Current 00: 8x 01: 4x 10: 2x 11: No amplification is implemented.
[2]	RSV	Reserved
[1:0]	AMP_GAIN	Amplification Gain for U/V/W-phase Current 00: 8x 01: 12x 10: 16x 11: 20x

29 Comparators

29.1 Comparator Operations

29.1.1 Comparator CMP3

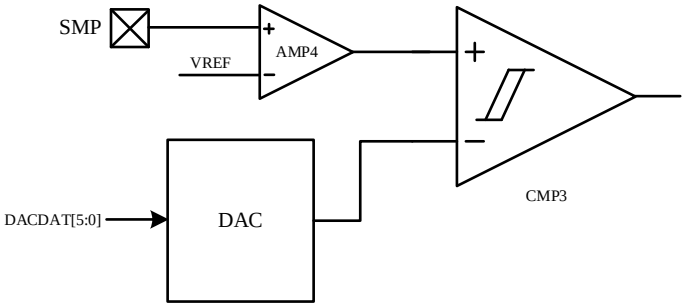


Figure 29-1 CMP3 I/O Pins

29.1.1.1 Over-current Protection (OCP) on Bus

When an over-current protection signal is generated on bus, DRV_OUT[MOE] is automatically cleared to 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 on bus is selected by configuring EVT_FILT[MOEMD] ≠ 00, namely CMP3 interrupt or external interrupt INT0. TCON[IT0] bit is programmed to select the trigger edge of the external interrupt INT0 which generates an OCP signal. At this time, the source of OCP interrupt is INT0. When CMP_CR0[CMP3IM] = 11, the OCP signal is generated on the raising edge of CMP3. At this time, the source of OCP interrupt is CMP3.

Configuring EVT_FILT[EFDIV] enables the filtering of interrupt signals for OCP on bus, and programming EVT_FILT[EFDIV] = 01/10/11 selects filter width of 4/9/19 clock cycles. When the filtering feature is enabled, the filtered signal is delayed by 4/9/19 clock cycles compared to the signal before filtering.

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 an OCP event occurs, DRV_OUT[MOE] is set to “1” by hardware after it has been cleared to “0” for a period of time, so that the motor drive is automatically restored. When CMP_CR0[CMP3IM] = 11, DRV_OUT[MOE] is cleared to “0” on the rising edge of CMP3OUT to protect the motor. When EVT_FILT[MOEMD] = 10, the outputs are automatically turned off upon an OCP interrupt. DRV_OUT[MOE] is enabled automatically upon Driver timer overflow/underflow events or after 10μs to restore motor drive. When EVT_FILT[MOEMD] = 11, the outputs are automatically turned off upon an OCP

interrupt, DRV_OUT[MOE] is enabled automatically upon Driver timer overflow/underflow events or after 5μs to restore motor drive.

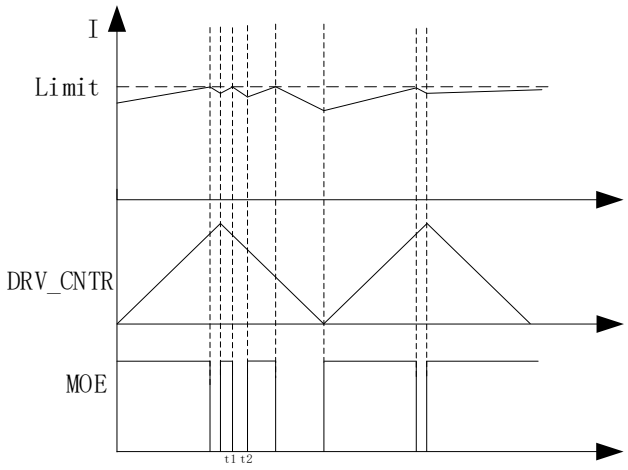


Figure 29-2 Cycle-by-cycle Current Limiting Waveform (t2 - t1 = 10μs) when EVT_FILT[MOEMD] = 10

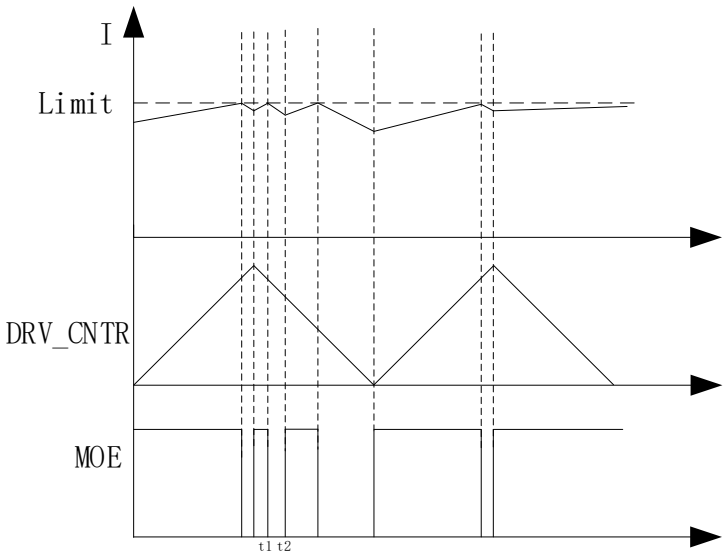


Figure 29-3 Cycle-by-cycle Current Limiting Waveform (t2 - t1 = 5μs) when EVT_FILT[MOEMD] = 11

29.1.2 Comparator Group (CMPG)

Comparator Group (CMPG) is a collection of CMP0, CMP1 and CMP2, with multiple comparison modes for different applications.

When CMP_CR2[CMP0MOD] = 01, CMPG works in the mode of three comparators with built-in resistors. It is used for BEMF detection of the internal resistors at virtual neutral point. The number of comparators operating in this mode is defined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0, CMP1 and CMP2 work simultaneously, which is the recommended configuration. When

CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works. When CMP_CR2[CMP0SEL] = 11, only CMP2 works.

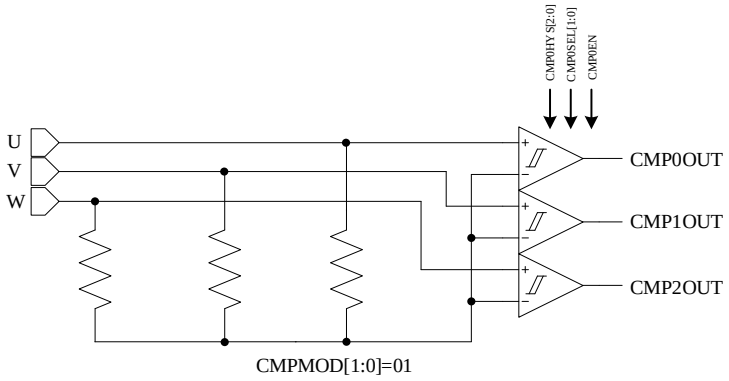


Figure 29-4 CMPG Mode with Built-in Three Comparators and Resistors

(without Function Switching)

When CMP_CR2[CMP0MOD] = 10, CMPG mode with two comparators is selected for motor speed detection. The I/O pins are shown in Figure 29-5. The positive inputs of the two comparators are connected together to U-phase, and the negative inputs are connected to V-phase and W-phase respectively. The outputs are CMP0OUT and CMP1OUT respectively. The number of comparators in this mode is defined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works.

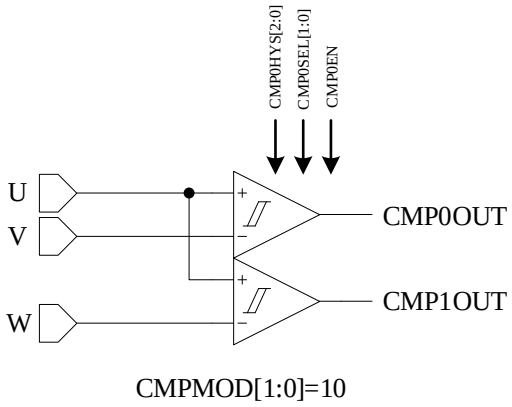


Figure 29-5 Dual-comparator Mode

The output signals of CMP0/CMP1/CMP2 are sent to Timer1 after filtering and sampling modules.

29.1.3 Comparator Sampling

The comparator sampling feature is mainly used for the square-wave control and RSD, which eliminates the switching interference from driving circuit. See section 16.1.2.3 for square-wave control and section 17.1.7.1 for RSD.

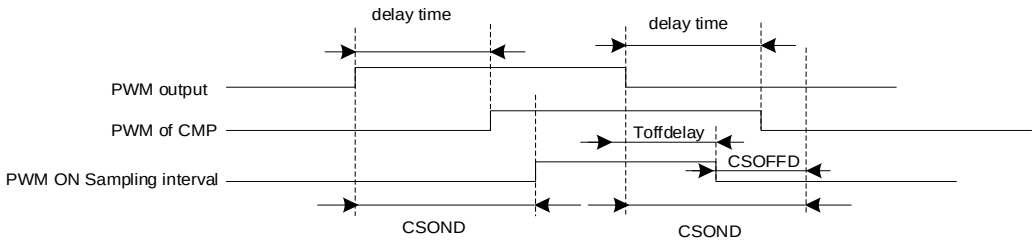


Figure 29-6 PWM ON Sampling Mode

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, ON/OFF speed of power IC, and input delay and hysteresis settings of the comparator. As shown in Figure 29-6, the delay time is from the chip output to the comparator output. When high-level sampling is performed, the sampling interval shall be enveloped by actual high-level output of the comparator. First, the sampling ON-delayed time CMP_SAMR[CSOND] is set to overcome output delay and avoid ringing due to PWM 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. The sampling OFF-lead time CMP_SAMR[CSOFFD] is set to stop sampling Toffdelay after the PWM output falling edge, where Toffdelay = CMP_SAMR[CSOND] – CMP_SAMR[CSOFFD]. By configuring CMP_SAMR[CSOND] and CMP_SAMR[CSOFDD], the sampling interval can be located in the high-level interval of the actual output of the comparator.

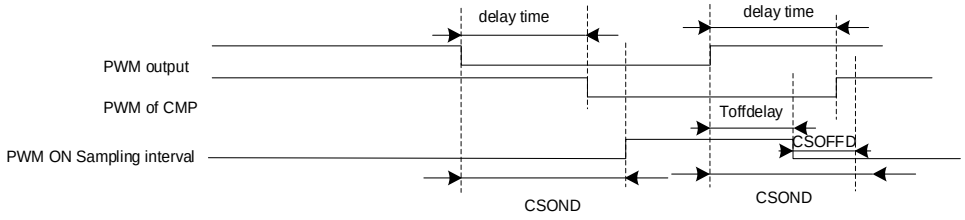


Figure 29-7 PWM OFF Sampling Mode

Similarly, when low-level sampling is performed, the sampling ON-delayed time CMP_SAMR[CSOND] and the sampling OFF-lead time CMP_SAMR[CSOFFD] are set reasonably to ensure that the actual sampling interval is located in the low-level output interval of the comparator.

Method for measuring the delay of PWM output to comparator: Set CMP_CR3[SAMSEL] = 00 to disable the comparator sampling delay feature. Set CMP_CR3[CMPSEL] to select the corresponding comparator output to test pin P0.1. 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 P0.1 or select function switching to P3.4.

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

Bit	Name	Description
[7:6]	CMP3IM	CMP3 Interrupt Mode 00: No interrupt is generated. 01: An interrupt is generated upon rising edge. 10: An interrupt is generated upon falling edge. 11: When a rising edge is detected, DRV_OUT[MOE] is cleared to “0”, and the interrupt event flag bit INT_SR3[CMP3IF] is set to “1”. However, the interrupt is disabled. (Note: In the Cycle-by-cycle Current Limiting mode, EVT_FILT[MOEMD] must be set to 10/11).
[5:4]	CMP2IM	CMP2 Interrupt Mode See descriptions on CMP_CR0[CMP0IM].
[3:2]	CMP1IM	CMP1 Interrupt Mode See descriptions on CMP_CR0[CMP0IM]
[1:0]	CMP0IM	CMP0 Interrupt Mode 00: No interrupt 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	RSV		CMP3MOD	CMP3EN	CMP3HYS	RSV	CMP0HYS	
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]	CMP3MOD	CMP3 Mode Selection 0: Reserved 1: Feeds into the comparator after operation amplifier. See Figure 29-1.
[4]	CMP3EN	CMP3 Enable 0: Disable 1: Enable
[3]	CMP3HYS	CMP3 Hysteresis Voltage Selection 0: No hysteresis 1: 9mv
[2]	RSV	Reserved
[1:0]	CMP0HYS	CMP0/1/2 Hysteresis Voltage Selection: 00: No hysteresis 01: ± 3mV 10: + 6mV 11: ± 12mV

29.2.3 CMP_CR2 (0xDA)

Bit	7	6	5	4	3	2	1	0
Name	RSV	CMP0MOD		CMP0SEL		RSV		CMP0EN
Type	-	R/W	R/W	R/W	R/W	-	-	R/W
Reset	-	0	0	0	0	-	-	0

Bit	Name	Description
[7]	RSV	Reserved
[6:5]	CMP0MOD	CMPG Mode Setting 00: Reserved 01: BEMF detection for the internal resistors at virtual neutral point 10: Dual-comparator mode. Phase is compared with V phase or W phase. 11: Reserved
[4:3]	CMP0SEL	CMPG 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 CMPG Port and CMP_CR2[CMP0MOD] Combination

CMP_CR2 [CMP0MOD]	CMP_CR2 [CMP0SEL]	Description
01	00	CMP0 works in three-comparator polling mode. The positive input is automatically switched among U/V/W-phase in turn, and the negative input is connected to the neutral point of built-in BEMF resistor. The output results are transferred to CMP0_OUT, CMP1_OUT, and CMP2_OUT respectively. See Figure 29-4.
	01	CMP0 selects the port combination corresponding to CMP0. The positive input is connected to U-phase, and negative input is connected to the neutral point of built-in BEMF resistor. The output results are transferred to CMP0_OUT.
	10	CMP0 selects the port combination corresponding to CMP1. The positive input is connected to V-phase, and negative input is connected to the neutral point of built-in BEMF resistor. The output results are transferred to CMP1_OUT.
	11	CMP0 selects the port combination corresponding to CMP2. The positive input is connected to W-phase, and negative input is connected to the neutral point of built-in BEMF resistor. The output results are transferred to CMP2_OUT.
10	00	CMP0 works in dual-comparator polling mode. The positive input is connected to U-phase, and the negative input is automatically switched between V-phase and W-phase in turn. The output results are transferred to CMP0_OUT and CMP1_OUT respectively. See Figure 29-5.
	01	CMP0 selects the port combination corresponding to CMP0. The positive input is connected to U-phase, and negative input is connected to V-phase. The output results are transferred to CMP0_OUT.
	10	CMP0 selects the port combination corresponding to CMP1. The positive input is connected to U-phase, and negative input is connected to V-phase. The output results are transferred to CMP1_OUT.

[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 P1.5 pin 00: Debug output is disabled 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 Output signals of one selected comparator to P0.1, which can be switched to P3.4. 000: No output 001: CMP0 010: CMP1 011: CMP2 100: CMP3 101: Reserved 110: Reserved 111: Omega Start Flag (Estimator Output Angle Flag, see section 14.1.8.1 for details)

29.2.5 CMP_CR4 (0xE1)

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

Bit	Name	Description
[7:3]	RSV	Reserved
[2]	FAEN	Filtered Signal Sampling Coefficient Scale-up Enable With it enabled, the base clock rates of TIM1_CR3[T1INM] and CMP_SAMR are scaled up by 4 times. 0: Disable 1: Enable
[1: 0]	RSV	Reserved

29.2.6 CMP_SAMR (0x40AD)

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

Bit	Name	Description
[7:4]	CSOND	CMP0/CMP1/CMP2 ON-delayed Sampling Time When PWM module switches from OFF to ON or from ON to OFF, turn-on/off of the power device affects input signal of the comparator. In this case, CMP_SAMR[CSOND] is configured to delay the sampling of CMP0/CMP1/CMP2. The on-delayed sampling time can be multiplied by 4 times by setting CMP_CR4[FAEN]. CMP_CR4[FAEN] = 0: ON-delayed sampling time = 8*CMP_SAMR[CSOND]*T CMP_CR4[FAEN] = 1: ON-delayed sampling time = 32*CMP_SAMR[CSOND]*T Notes: ■ CMP_SAMR[CSOND] value must be greater than or equal to CMP_SAMR[CSOFFD] value. ■ See section Sampling for BLDC drive application. ■ See section RSD Comparator Sampling for RSD application.
[3:0]	CSOFFD	CMP0/CMP1/CMP2 OFF-lead Sampling Time CMP_SAMR[CSOND] is configured to end the sampling CMP_SAMR[CSOND] - CMP_SAMR[CSOFFD] after the back edge of PWM output to ensure sampling interval is enveloped by the PWM interval. OFF-lead sampling time can be multiplied by 4 times by setting CMP_CR4[FAEN]. CMP_CR4[FAEN] = 0: OFF-lead sampling time = 8*CMP_SAMR[CSOFFD]*T CMP_CR4[FAEN] = 1: OFF-lead sampling time = 32*CMP_SAMR[CSOFFD]*T Notes: ■ CMP_SAMR[CSOND] value must be greater than or equal to CMP_SAMR[CSOFFD] value. ■ See section Sampling for BLDC drive application. ■ See section RSD Comparator Sampling for RSD application.

29.2.7 CMP_SR (0xD7)

Bit	7	6	5	4	3	2	1	0
Name	RSV				CMP3OUT	CMP2OUT	CMP1OUT	CMP0OUT
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0

Bit	Name	Description
[7:4]	RSV	Reserved
[3]	CMP3OUT	CMP3 comparison result
[2]	CMP2OUT	CMP2 comparison result
[1]	CMP1OUT	CMP1 comparison result
[0]	CMP0OUT	CMP0 comparison result

29.2.8 HALL_CR (0xE2)

Bit	7	6	5	4	3	2	1	0
Name	RSV	HALL_IE	RSV		HALLSEL	HALL2	HALL1	HALL0
Type	-	R/W	-	-	R/W	R/W	R/W	R/W
Reset	-	0	-	-	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	HALL_IE	Hall Interrupt Enable 0: Disable 1: Enable Note: See INT_SR3 (0xF5) for Hall Interrupt Flag
[5:4]	RSV	Reserved
[3]	HALLSEL	Hall Input Selection 0: P1.4/P0.5/P0.6 1: P1.4/P0.2/P0.3
[2]	HALL2	Hall2 Level 0: Hall2 level = 0 1: Hall2 level = 1
[1]	HALL1	Hall1 Level 0: Hall1 level = 0 1: Hall1 level = 1
[0]	HALL0	Hall0 Level 0: Hall0 level = 0 1: Hall0 level = 1

29.2.9 LCP_DR (0x404A)

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

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	DACDAT	Voltage Settings for CMP3. It works with CMP3MOD and VREF_CR[VREFEN] must be set to “1”. When VREF = 4.5V and CMP_CR1[CMP3MOD] = 0, the regulating voltage ranges from 35mV to 589mV and the step is 8.79mV. When VREF = 4.5V and CMP_CR1[CMP3MOD] = 1, the regulating voltage ranges from 2.285V to 4.5V and the step is 35.16mV.

29.2.10 EVT_FILT (0xD9)

Bit	7	6	5	4	3	2	1	0
Name	OVPF	OCPF	RSV	MOEMD		RSV	EFDIV	
Type	R	R	-	R/W	R/W	-	R/W	R/W
Reset	0	0	-	0	0	-	0	0

Bit	Name	Description
[7]	OVPF	OVP Flag Bit 0: OVP is not triggered. 1: OVP is triggered.
[6]	OCPF	OCP Flag Bit 0: OCP is not triggered. 1: OCP is triggered.
[5]	RSV	Reserved
[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 Driver timer overflow/underflow events or after 10μs (for square-wave drive). 11: MOE is automatically cleared and enabled automatically upon Driver timer overflow/underflow events or after 5μs (for square-wave drive).
[2]	RSV	Reserved
[1:0]	EFDIV	Filter Width for Current Protection 00: Not to filter 01: 4 system clock cycles 10: 9 system clock cycles 11: 19 system clock cycles

30 Over-voltage Protection and Over-current Protection

30.1 Over-voltage Protection (OVP)

The chip integrates OVP module. When bus voltage exceeds the specified value, the chip automatically turns off the output to prevent MOS from being damaged.

Configuring OVCP_CR[OVPDIS] = 0 enables the OVP feature. When OVP_CR[OVPVSEL] is set to “0”, over-voltage protection occurs if the bus voltage is higher than 24V. When OVP_CR[OVPVSEL] is set to “1”, over-voltage protection occurs if the bus voltage is higher than 30V. In these cases, EVIT_FILT[OVPF] is set to “1” and DRV_OUT[MOE] is automatically cleared to “0” by the hardware.

30.2 Over-current Protection (OCP)

The chip integrates 2A OCP module. When an over-current event occurs, the chip automatically turns off the output to prevent MOS from being damaged.

Configuring OVCP_CR[OCPDIS] = 0 enables the OCP feature. When an over-voltage event occurs, EVIT_FILT[OCPF] is set to “1” and DRV_OUT[MOE] is automatically cleared to “0” by the hardware.

For the input signal, configuring OVCP_CR[OCPFSEL] = 00/01/10/11 selects the filtering width as 0/4/9/19 clock cycle. The filtered signal is delayed by 0/4/9/19 clock cycle than the original.

30.3 OVP and OCP Registers

30.3.1.1 OVCP_CR (0xE3)

Bit	7	6	5	4	3	2	1	0
Name	RSV	OVPVSEL	OVPIE	OVPDIS	OCPFSEL		OCPIE	OCPDIS
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]	OVPVSEL	OVP Voltage Selection 0: 24V 1: 30V
[5]	OVPIE	OVP Interrupt Enable 0: Disable 1: Enable Note: See INT_SR4 (0xF6) for OVP Interrupt Flag.
[4]	OVPDIS	OVP Disable 0: Turn on 1: Turn off
[3:2]	OCPFSEL	Filter Width for OCP 00: Not to filter 01: 4 system clock cycles 10: 9 system clock cycles 11: 19 system clock cycles
[1]	OCPIE	OCP Interrupt Enable 0: Disable 1: Enable Note: See INT_SR4 (0xF6) for OCP Interrupt Flag.
[0]	OCPDIS	OCP Disable 0: Turn on 1: Turn off

31 Power Supply

31.1 Low Voltage Detection (LVD)

31.1.1 LVD Introduction

The low voltage detection includes low voltage warning and low voltage reset.

31.1.2 LVD Operations

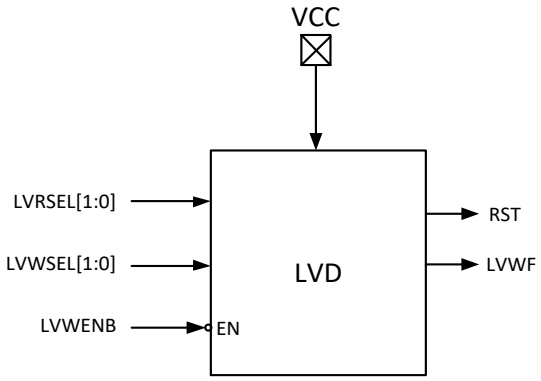


Figure 31-1 LV Detection Module

The operating instructions for LVD are as follows:

- LV warning and LV reset are always enabled by default.
- 6/7/8/9V can be selected for LV warning threshold. When the interrupt feature is enabled, an interrupt is triggered if VCC voltage is lower than the LV warning threshold.
- 3.0/3.5/3.8V can be selected for the LV reset threshold. The chip resets when VCC voltage is lower than the LV reset voltage threshold.

LV warning threshold, interrupt settings and LV reset threshold are configured through the IDE, as shown in Figure 31-2.

LVR Config sets low voltage reset threshold, LVW Interrupt En enables low voltage interrupt, and LVW Config sets low voltage warning threshold.

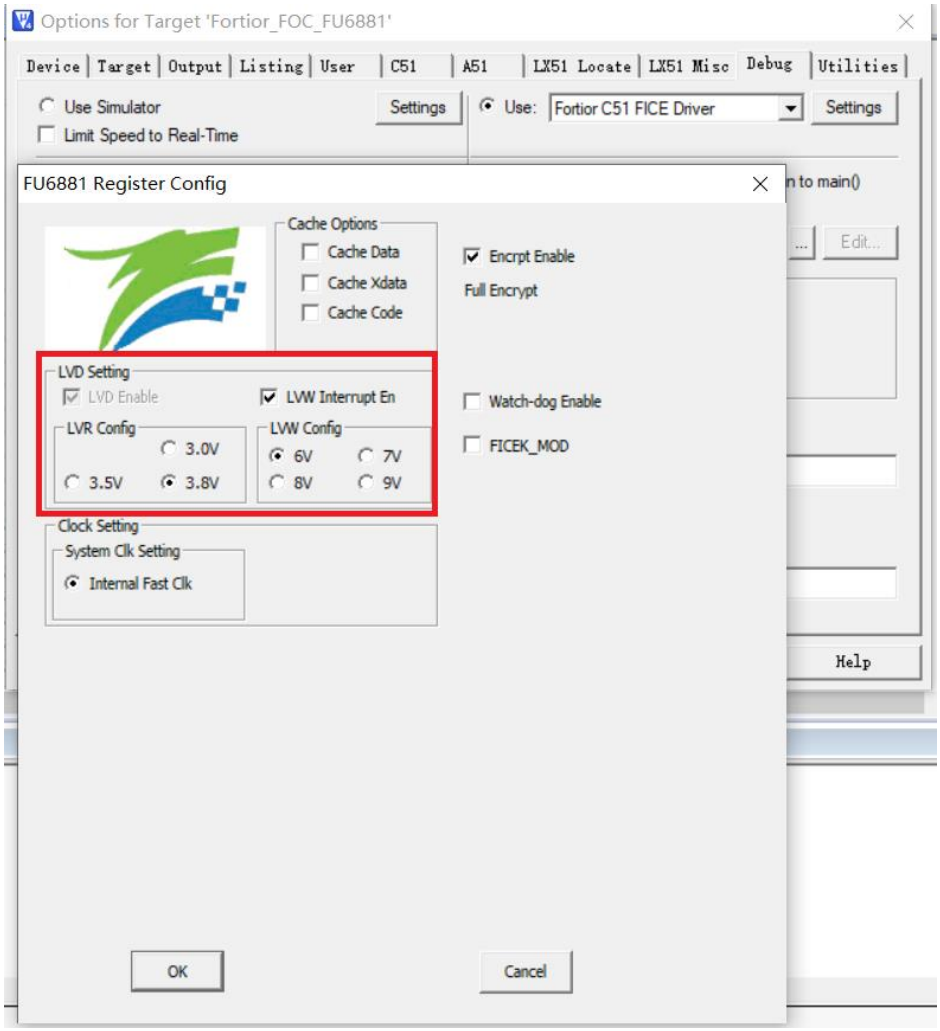


Figure 31-2 Configurations of LV Reset Threshold, LV Interrupt and LV Warning Threshold

31.1.3 LVD Registers

31.1.3.1 LVSR (0xDB)

Bit	7	6	5	4	3	2	1	0
Name	RSV		EXT0CFG			TSDF	LVWF	RSV
Type	-	-	R/W	R/W	R/W	R	R	-
Reset	-	-	0	0	0	0	0	-

Bit	Name	Description
[7:6]	RSV	Reserved
[5:3]	EXT0CFG	INT0 Pin Selection 000: P0.0 001: P0.1 010: P0.2 011: P0.3 100: P0.4 101: P1.0 110: P1.1 111: P1.5
[2]	TSDF	Over-temperature Status Flag 0: Current temperature does not exceed the predefined temperature. 1: Current temperature exceeds the predefined temperature. Note: It works with TSD Interrupt Flag (INT_SR3[6]) .
[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]	RSV	Reserved

32 EEPROM

32.1 EEPROM Introduction

EEPROM is a programmable read-only memory with random read, page write and byte read/write operations. As shown in Figure 32-1, to generate START condition, SDA is changed from high to low while keeping SCL high; and to generate STOP condition, SDA goes from low to high while keeping SCL high. EEPROM includes device address and byte address. As shown in Figure 32-2, the device address consists of a 6-bit ID address and a 1-bit block select bit. The ID address is the slave address, which is 0xA0 by default. B0 is the select bit of 2k block. The byte address is an 8-bit address with the high-order 4 bits being the page address.

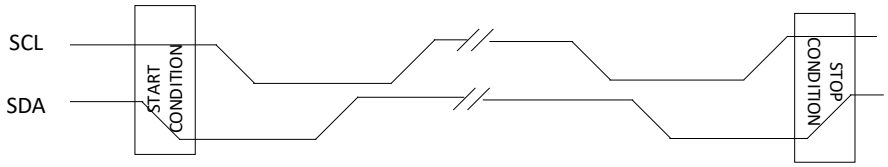


Figure 32-1 START and STOP Conditions of EEPROM

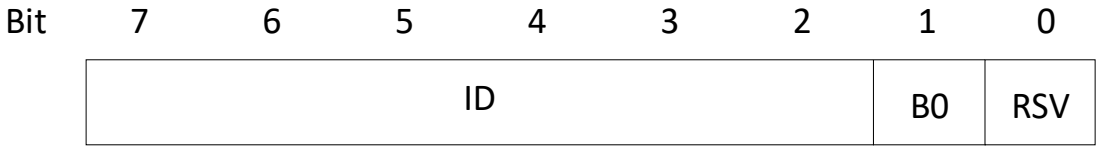


Figure 32-2 Slave Device Address

During the power-on period, the device does not respond to any commands unless the supply voltage (VCC) is higher than the reset threshold voltage. When VCC rises above the reset threshold voltage, the device resets and enters standby mode to avoid unexpected write operations. During the power-off period, the device enters standby mode when VCC drops below the power-on threshold voltage. In addition, if no write operation is being performed, the device enters standby mode after receiving the stop command.

32.2 Main Features

- Support 1MHz and 400KHz modes;
- 2k bits (256 x 8 bits) memory array, organized by 16 pages of 16 bytes each;
- Data read/write via I^2C. In the read mode, the data can be read sequentially at any position of any length. In the write mode, a maximum of 16 bytes can be written and the written data is stored in the page where its byte address resides.
- Self-timed write cycle: 5ms

32.3 Operating Instructions

32.3.1 Read Mode

1. Configure PH_SEL2[I2CFS] = 1 to enable access EEPROM via I²C;
2. Configure I2C_ID[7:0] =0xA0 to match EEPROM device code, and configure I2C_ID[1] to select EEPROM block;
3. Configure I2C_CR[I2CMS] = 1 to select the master mode;
4. Set SCL frequency by I2C_CR[I2CSPD];
5. Configure I2C_SR[DMOD] = 0 to set the write direction;
6. Configure I2C_CR[I2CEN] = 1 to enable I²C;
7. Set I2C_SR[I2CSTA] to "1" to send the START and address. If an ACK signal is received, EEPROM is successfully matched and the device proceeds to the next step. If an NACK signal is received, EEPROM fails to be matched;
8. Configure I2C_DR to send byte addresses. This address is the start address for subsequent read operations;
9. Configure I2C_SR[DMOD] = 1 to set the read direction, keep I2C_ID[7:2] unchanged and set I2C_SR[I2CSTA] to "1" to change the direction for I²C data. EEPROM responds with an ACK to start to read the data;
10. After receiving the data, MCU responds with an NACK to request EEPROM to stop reading the data, or responds with an ACK to request another data transfer where EEPROM increments the byte address by one and transfers the data b. By responding ACK signals multiple times, page reading or sector reading is completed;
11. Stop Communication: The master sends the STOP signal to stop communication.

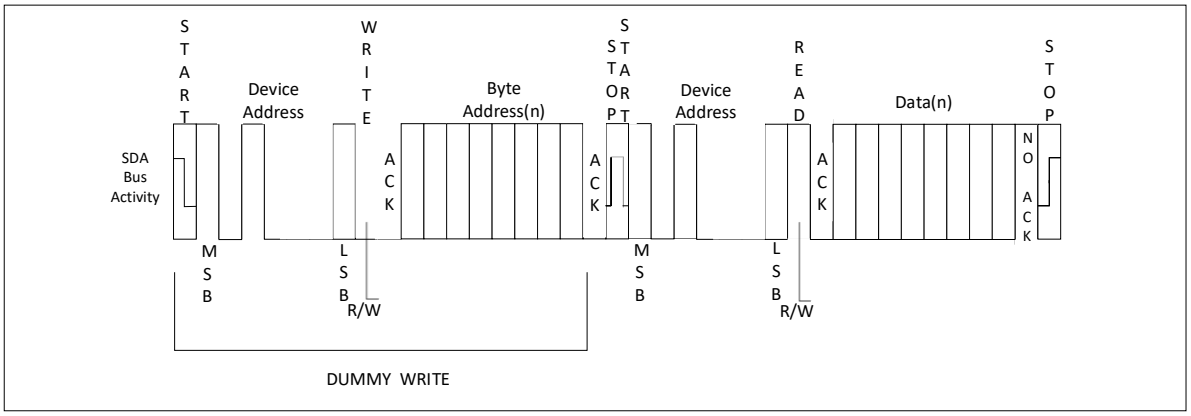


Figure 32-3 Random Read

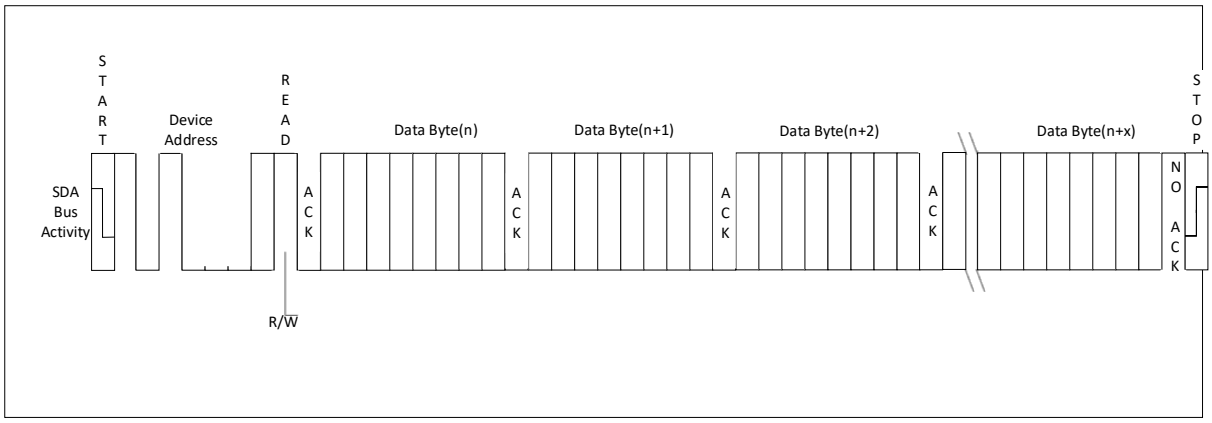


Figure 32-4 Sequential Read

32.3.2 Write Mode

1. Configure PH_SEL2[I2CFS] = 1 to enable access EEPROM via I^2C;
2. Configure I2C_ID[7:0] =0xA0 to match EEPROM device code, and configure I2C_ID[1] to select EEPROM block;
3. Configure I2C_CR[I2CMS] = 1 to select the master mode;
4. Set SCL frequency by I2C_CR[I2CSPD];
5. Configure I2C_SR[DMOD] = 0 to set the write direction;
6. Configure I2C_CR[I2CEN] = 1 to enable I^2C;
7. Set I2C_SR[I2CSTA] to “1” to send the START and address. If an ACK signal is received, EEPROM is successfully matched and the device proceeds to the next step. If an NACK signal is received, EEPROM fails to be matched as EEPROM may be offline or busy.
8. Configure I2C_DR to send byte addresses. This address is the start address for subsequent write operations, and determines the page address to be written.
9. Write the byte data. The data can be sent up to 16 times, and after that EEPROM makes no response. The written data is stored in the page where its byte address resides. For example, after 0x01 is written to the byte address and 16 groups of data are sent, the last byte of data is stored into byte address 0x00 instead of 0x11
10. Stop Communication: The master sends the STOP signal to stop communication. EEPRM also enters the busy state and does not respond to any communication data in 5ms.

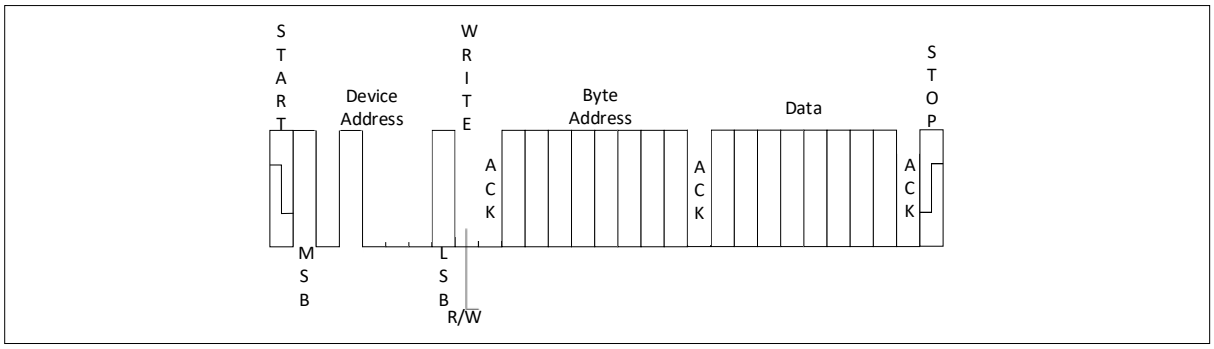


Figure 32-5 Byte write

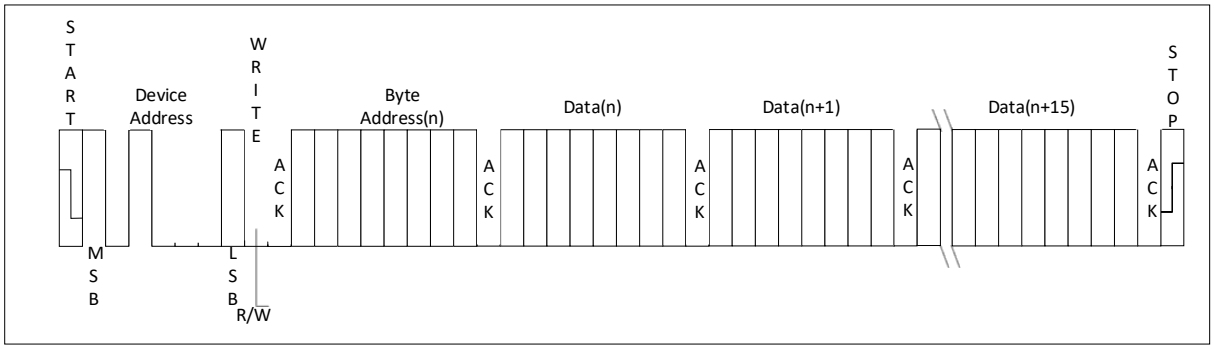


Figure 32-6 Page Write

32.4 EEPROM Registers

32.4.1 I2C_CR (0x4028)

Bit	7	6	5	4	3	2	1	0
Name	I2CEN	I2CMS	EROMVDD5PD	I2CDMANAKINT	I2CDMAAUTO	I2CSPD		I2CIE
Type	R/W	R/W	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]	I2CEN	I^2C Enable The associated GPIO is enabled to switch to I^2C mode, serving as open-drain output. The pull-up setting decides whether to pull I^2C HIGH. 0: Disable 1: Enable
[6]	I2CMS	Master/Slave Mode Selection 0: Slave 1: Master
[5]	EROMVDD5PD	EEPROM 5V Power Supply or Power Down 0: EEPROM 5V Power Supply 1: EEPROM Power Down
[4]	I2CDMANAKINT	An interrupt is generated when the device does not acknowledge (NAK) the received the data during DMA transmission. 0: Disable 1: Enable
[3]	I2CDMAAUTO	Automatically transfer the first byte of data during DMA transmission 0: Disable 1: Enable
[2:1]	I2CSPD	I^2C transfer rate setting, valid only in Master Mode 00: 100kHz 01: 400kHz 10: 1MHz 11: Reserved
[0]	I2CIE	I^2C Interrupt Enable 0: Disable 1: Enable Note: See INT_SR5 (0xF7) for I^2C Interrupt Flag.

Note: See section 9.3.2 for I2C_ID register and section 9.3.3 for I2C_DR register.

32.4.2 PH_SEL2 (0x4049)

Bit	7	6	5	4	3	2	1	0
Name	RSV						I2C_FS	RSV
Type	-	-	-	-	-	-	R/W	-
Reset	-	-	-	-	-	-	0	-

Bit	Name	Description
[7:2]	RSV	Reserved
[1]	I2C_FS	Access EEPROM via I^2C after Function Switching 0: Disable 1: Enable
[0]	RSV	Reserved

33 Flash

33.1 Flash Introduction

The chip provides 32k bytes of Flash space. It supports page erase, page pre-programing and write.

Main features:

- 128 sectors in total, each with a size of 256 bytes
- 16 pages in total, each with 8 sectors
- Last sector (address range: 0x7F00~0x7FFF) cannot be erased at any time
- 120ms~150ms for page erase
- Programming is enabled when FLA_CR [FLAEN] is set to “1”, where page pre-programming, page erase or write and other Flash operations are activated with MOVX instructions.

33.2 Flash Operations

- Flash memory must be unlocked before erase and programming operations. 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.
- CRC results change if Flash memory is rewritten during program execution.
- Page pre-programming must be done before page erase.
- Configuring FLA_CR=0x23 enables page erase, FLA_CR=0x25 enables page pre-programming and FLA_CR=0x21 enables write operations.

Note: All interrupts must be disabled before self-programming to ensure the security of Flash operations and avoid mis-operation of Flash using MOVX instruction during interrupt processing.

33.3 Flash Registers

33.3.1 FLA_CR (0x85)

Bit	7	6	5	4	3	2	1	0
Name	RSV		FLAPAGE	FLAERR	RSV	FLAPRE	FLAERS	FLAEN
Type	-	-	R/W	R	-	R/W	R/W	R/W
Reset	-	-	0	0	-	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	FLAPAGE	Page Operation Enable 0: Disable 1: Enable
[4]	FLAERR	Programming Error Flag 0: Programming or pre-programming succeeds. 1: Programming or pre-programming fails.
[3]	RSV	Reserved
[2]	FLAPRE	Pre-programming Enable 0: Disable 1: Enable Note: FLA_CR[FLAPRE] is valid only when FLA_CR[FLAEN] = 1.
[1]	FLAERS	Erase Enable 0: Disable 1: Enable Note: FLA_CR[FLAERS] is valid only when FLA_CR[FLAEN] = 1.
[0]	FLAEN	Programming Enable 0: Disable 1: Enable

33.3.2 FLA_KEY (0x84)

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

Bit	Name	Description
[7:0]	FLA_KEY	Write: Write “0x5A” and “0x1F” in sequence to unlock Flash operations; Write any value to FLA_CR bit to lock Flash operations.

Bit	7	6	5	4	3	2	1	0
Name	RSV						FLAKSTA	
Type	-	-	-	-	-	-	R	R
Reset	-	-	-	-	-	-	0	0

Bit	Name	Description
[7:2]	RSV	Reserved
[1:0]	FLAKSTA	Read: Flash release status 00: Locked 01: 0x5A is written, waiting for 0x1F 10: Frozen 11: Unlocked

34 CRC

34.1 CRC Functional Block Diagram

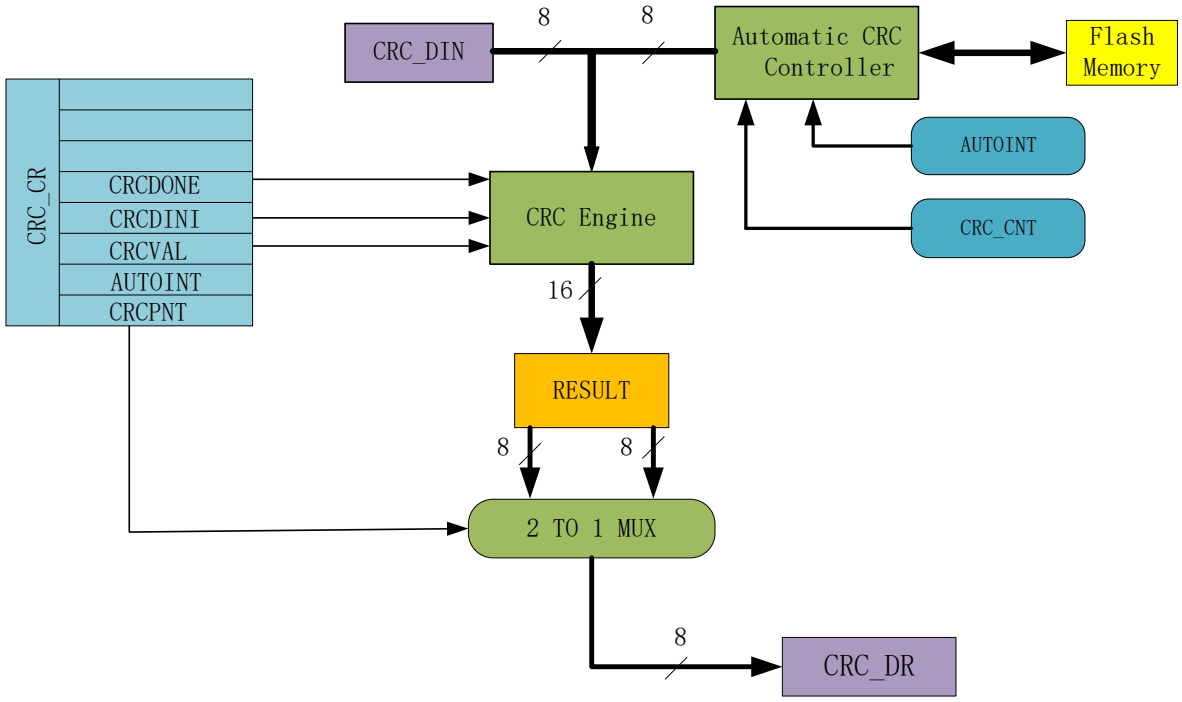


Figure 34-1 CRC Functional Block Diagram

CRC module outputs the result of CRC calculation for any 8-bit data based on a fixed polynomial. As shown in Table 34-1, CRC receives the 8-bit data from CRC_DIN and sends the 16-bit result to the internal register after the calculation is completed. The result can be indirectly accessed through CRC_CR[CRCPNT] and CRC_DR.

34.2 CRC16 Polynomial

The chip uses CRC16/CCITT-FALSE polynomial.

Table 34-1 CRC Criteria and Polynomial

CRC Criteria	Polynomial	Hexadecimal Representation
CRC16-CCITT-FALSE	x^16+x^12+x^5+1	0x1021

34.3 CRC16 Logic Diagram

The schematic diagram of CRC16 is shown in Figure 34-2. The chip implementation is based on parallel algorithm. For each input byte, MCU calculates the results within one system clock cycle.

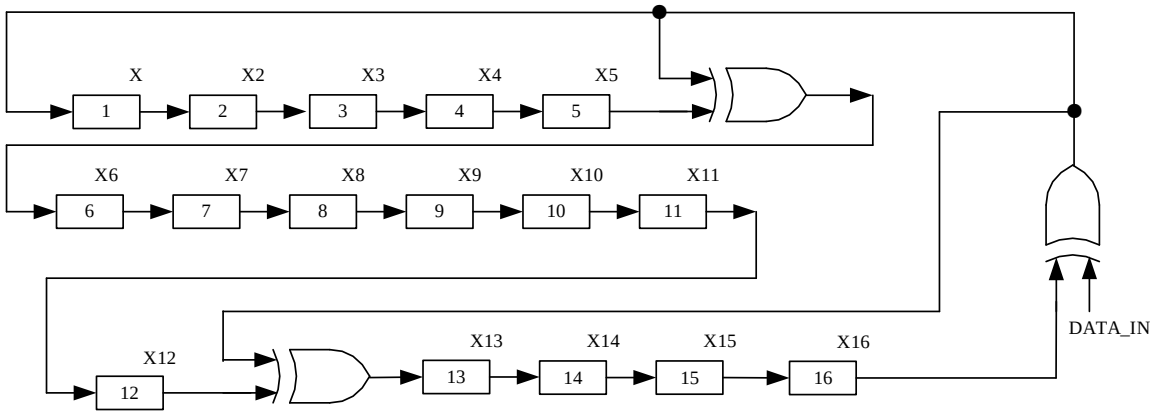


Figure 34-2 CRC16 Schematic Diagram

34.4 CRC Operations

34.4.1 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 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.

34.4.2 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.

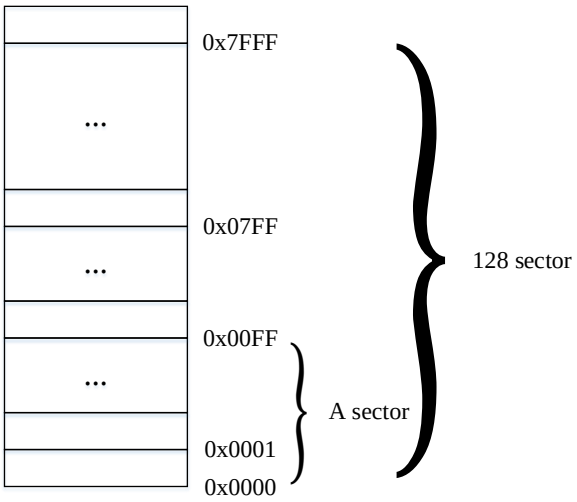


Figure 34-3 ROM Sectors

As shown in Figure 34-3, ROM contains 32k bytes and is divided into 128 sectors, numbered from sector0 to sector127. Each sector contains 256 bytes. For CRC calculation of sectors, the value of CRC_BEG (the starting sector) can be any value falling between 0x00 and 0xFF, including 0x00 and 0x7F. The CRC_CNT (total number of sectors to be calculated) can be any value between 0x00~0x7F, including 0x00 and 0xFF.

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, CRC controller will automatically limits the number of sectors to be calculated. Finally, CRC module only calculates CRC value of the last sector.

34.5 CRC Registers

34.5.1 CRC_CR (0x4022)

Bit	7	6	5	4	3	2	1	0
Name	RSV			CRCDONE	CRCDINI	CRCVAL	AUTOINT	CRCPNT
Type	-	-	-	R	W1	R/W	W1	R/W
Reset	-	-	-	1	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	CRCDONE	CRC Sector Calculation Completion Flag During the calculation, this bit is automatically set to “0” by hardware and the software program stops. In other cases, this bit is automatically set to “1” by the hardware, so the software always returns “1” when reading this bit.
[3]	CRCDINI	CRC Result Initialization Trigger 0: No effect 1: CRC result initialization is triggered.
[2]	CRCVAL	CRC Result Initialization Selection 0: CRC result is initialized to 0x0000. 1: CRC result is initialized to 0xFFFF.
[1]	AUTOINT	CRC Sector Calculation Launch 0: No effect 1: Launch CRC batch calculation See section CRC Calculation of ROM Sector.
[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_CR[AUTOINT] is set to “0” to perform single-byte CRC checksum.

34.5.2 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.

34.5.3 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 or low 8 bits of the CRC result.

34.5.4 CRC_BEG (0x4024)

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

Bit	Name	Description
[7]	RSV	Reserved
[6:0]	CRC_BEG	First ROM Sector Pending Auto CRC Calculation Example: If CRC_BEG is set to “1”, CRC calculation starts from location 1*256 = 256, or rather from the first byte of sector 2.

34.5.5 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.

35 Sleep Mode

35.1 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 35-1.

Table 35-1 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	Flash Deep Sleep. The analog fast clock circuit is disconnected and MCU software shall ensure that ADC, FOC, and driver modules are idle before the chip enters the Sleep Mode. WDT is disabled.	External interrupt, RTC interrupt, Level changes on P1.7 in IO mode, 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_();
_nop_();
```

35.2 Sleep Mode Register

35.2.1 PCON(0x87)

Bit	7	6	5	4	3	2	1	0
Name	RSV		GF3	GF2	GF1	RSV	STOP	IDLE
Type	-	-	R/W	R/W	R/W	-	R/W	R/W
Reset	-	-	0	0	0	-	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	GF3	General-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. This bit is automatically cleared to “0” by hardware after wakeup.
[0]	IDLE	A write of “1” makes the chip enter the standby mode. This bit is automatically cleared to “0” by hardware after wakeup.

Power Consumption Mode PCON[STOP:IDLE]:

00: Normal

01: Standby

1X: Sleep

36 Code Protection

36.1 Introduction

The chip supports Flash space encryption to protect your software intellectual property and avoid unauthorized access. When Flash memory is encrypted, the data inside cannot be read, and data consistency can be verified by CRC module only.

36.2 Operating Instructions

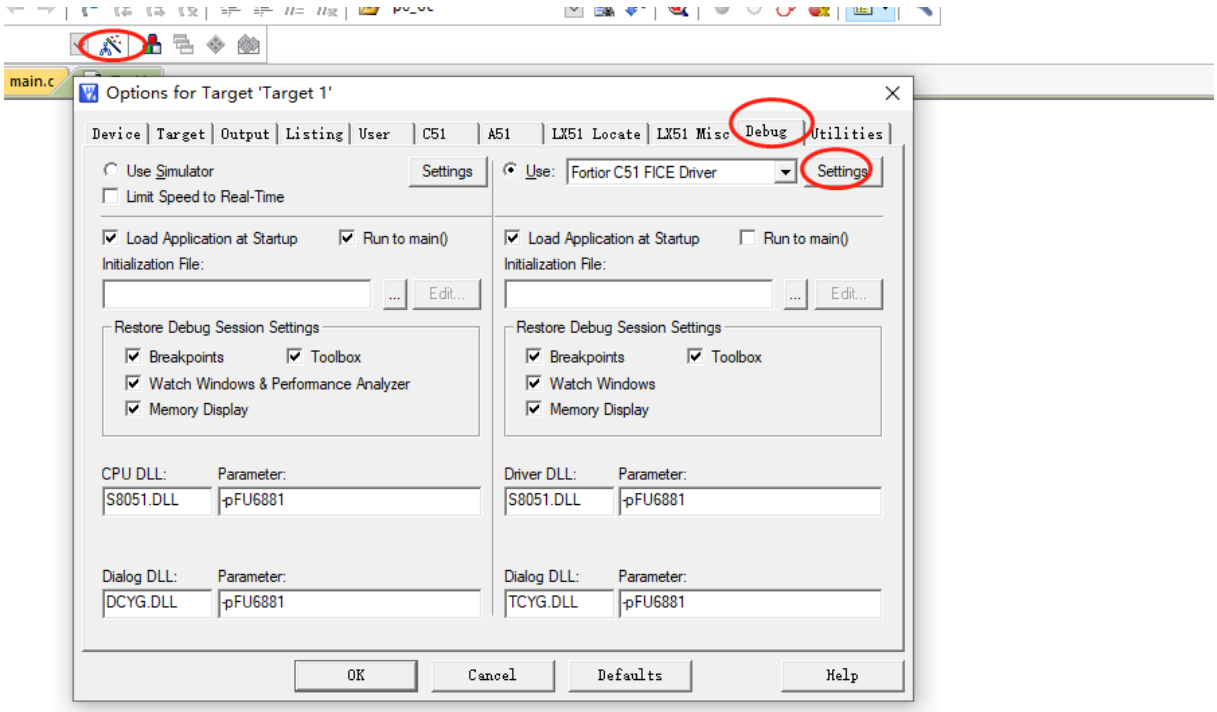


Figure 36-1 Code Protection Configurations

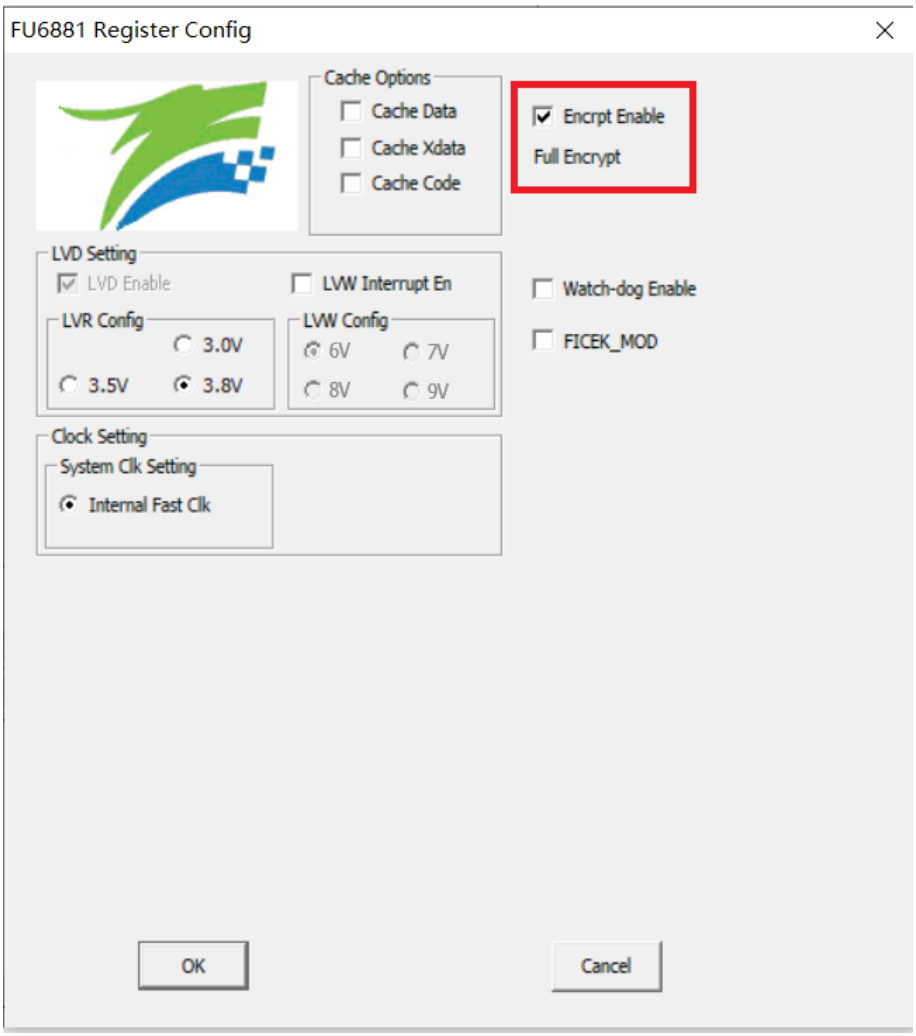


Figure 36-2 Full Code Protection Mode

Operation steps are as follows:

1. Start 8051 IDE, enter Target Options and select Debug tab. As shown in Figure 36-1, click Settings to proceed with the setting;
2. Select the options as shown in Figure 36-2, and click OK. Then compile the project and download it. Get the BIN file and program it to Flash.

37 Revision History

Rev.	Description	Date	Prepared By
V0.2	Preliminary datasheet, translated from Chinese version 0.2	2023/07/26	Eric Deng
V0.3	1. Removed Vds protection from section 1.1 Features; 2. Added LIN-VCC pin in Figure 1-1 FU6881Q1 Functional Block Diagram; 3. Updated Table 2-1 FU6881Q1 QFN40 Pins; 4. Updated Figure 2-1 FU6881Q1 QFN40 Pinout Diagram; 5. Updated Figure 3-1 QFN40_5X5 Package Dimensions; 6. Modified bit [1] "RSV" in of IE (0xA8) in section 6.5.1 as "TSDIE", and added descriptions on this bit; 7. Modified the range of FOC_DMAX in section 13.2.39 FOC_DMAX (0x4078) from [-256,255] to [-128,127]; 8. Corrected "Timer1 supports 6 interrupt sources…" in section 15.1.4 Timer1 Interrupt as "Timer1 supports 7 interrupt sources…"; 9. Modified the description "1*128 = 128" on the bit [6:0] in section 33.5.4 CRC_BEG (0x4024) as 1*256 = 256; 10. Optimized document format.	2023/08/10	Lydia Zhu
V0.4	1. Added section 5 Electrical Characteristics; 2. Modified EEPROM 4K bits as 2k bits; 3. Modified the number of comparator from 4 to 2; 4. Added the description "See section 5.3 GPIO Electrical Characteristics for the values of pull-up resistors and pull-down resistors." in section 23.2 IO Operations; 5. Added the description "See ADC Electrical Characteristics for sampling time and conversion time." In section 25.1 ADC Introduction; 6. Added descriptions on bit [2] in 28.2.2 AMP_CR1 (0x4034); 7. Updated descriptions in section 29.1.1.1 Over-current Protection (OCP) on Bus; 8. Added Table 29-1 Function Description of CMPG Port and CMP_CR2[CMP0MOD] Combination in section 29.2.3 CMP_CR2 (0xDA), and modified the bit [2:1]CMPOCSEL as RSV; 9. Optimized document format.	2023/08/16	Eric Deng
V1.0	Final Datasheet. 1. Deleted descriptions on FG interrupt; 2. Modified DRV_CR(0x4062) as DRV_CR0(0x4062); 3. Updated Figure 1-1 FU6881Q1 Functional Block Diagram and moved TSD functional block to the bus line; 4. Added "VCC to VSS Voltage" with the test condition "Phase-U/V/W output is turned off during chip operation" in Table 5-1 Absolute Maximum Ratings; 5. Added "I_{VCC} Sleep-mode Current" in Table 5-2 Global Electrical Characteristics; 6. Added Note in section 5.6 Operational Amplifier Electrical Characteristics; 7. Modified the enable bit "DMA0_CR0[0]" in Table 7-1 Interrupt Summary as "DMA0_CR1[0]"; 8. Added descriptions on the range of received baud rate in chapter 12 LIN; 9. Modified values of Type and Reset on bit [6], [3] and [2] in section 14.2.1 FOC_CR0 (0x409F); 10. Modified the description "…the target speed into FOC_EFREQMIN and FOC_EFREQACC…" as "…the target speed into FOC_EFREQMIN and FOC_EFREQHOLD…"; 11. Added section 20.2.1 DRV DTR(0x4060);	2023/11/24	Eric Deng

Rev.	Description	Date	Prepared By
	12. Modified the description on bit [1:0] in section 20.2.3 DRV_CR1 (0x4061) “See INT_SR2 (0xF4) for FG Interrupt Flag” as “See INT_SR2 (0xF4) for Driver Compare Match Interrupt Flag”; 13. Modified the name “P1_AN” of bit [7] in section 23.3.3 P1_AN (0x4050) as “RSV”; 14. Modified the description on step 5 in section 32.3.1 Read Mode “Configure I2C_SR[DMOD] = 1 to set the write direction” as “Configure I2C_SR[DMOD] = 0 to set the write direction” and modified the description on step 9 “Configure I2C_SR[DMOD] = 0 to set the read direction” as “Configure I2C_SR[DMOD] = 1 to set the read direction”; 15. Modified the description on step 5 in section 32.3.2 Write Mode “Configure I2C_SR[DMOD] = 0 to set the write direction” as “Configure I2C_SR[DMOD] = 1 to set the write direction”; 16. Deleted descriptions on preliminary datasheet in Cover, Footer and Copyright Notice; 17. Standardized document format.		
V1.1	1. Added step controller registers ST_VABK, ST_BPCOMP, ST_DZCOMP, ST__ETHETA and ST__OMEEST in Table 1-2 XSFR Address Mapping; 2. Deleted FOC_CR1[ANGM] = 1 in section 14.1.8.1.2 Forced Angle of Estimator; 3. Added “The voltage division ratio of the bus voltage is 1/10, and the current sampling resistance is 0.1Ω” in section 14.1.7 Current and Voltage Sampling; 4. Added “The voltage division ratio of the bus voltage is 1/10.” in section 14.2.57 FOC UDCFLT(0x4098,0x4099), and modified the description “The bus voltage is scaled down by 1/6 ….,then bus voltage = 19661/32768*5V*10 = 18V.” as “The bus voltage is scaled down by 1/10 …., then bus voltage = 19661/32768*5V*10 = 30V.”; 5. Modified the bit CPCKSEL in 20.2.3 DRV_CR1 (0X4061) as RSV; 6. Deleted ADCRATIO in Figure 25-2 ADC Functional Block Diagram; 7. Modified EEPROM feature “2k bits (512x8 bits) memory array” as “2k bits (256 × 8 bits) memory array” in section 32.2 Main Features; 8. Modified some descriptions.	2024/03/22	Eric Deng
V1.2	1. Modified MIPS(Peak) in Table 4-1 Model Selections as MHz; 2. Added the parameter VCC to VSS Voltage (test condition: phase-U/V/W output is turned on) in Table 5-1 Absolute Maximum Ratings 3. Modified “VDD” in section 6.4 Low Voltage Detection Reset as “VCC”; 4. Modified “LIN controller complies with the 2.2 Specification (backward compatible)” as “LIN controller complies with ISO 17987 standard” in section 12 LIN, and “bit error” as “bit error/format error”; 5. Updated Figure 29-1 CMP3 I/O Pins; 6. Deleted inapplicable CRC Criteria in Table 34-1 CRC Criteria and Polynomials; 7. Modified some descriptions.	2024/06/12	Eric Deng
V1.3	1. Added AEC-Q100 Qualified (Grade 1) in section 1.1 Features; 2. Added “open drain” to the descriptions on P0.2, P1.0 and P1.1 in section 2.1 FU6881Q1 QFN40 Pins;	2024/09/14	Eric Deng

Rev.	Description	Date	Prepared By
	3. Added “ADCLK2” to sampling time of ADC module; 4. Added section 5.10 LIN Electrical Characteristics and updated descriptions in section 12.1 LIN Introduction; 5. Corrected “TIM1_KFMAX” as “TIM1_KRMAX”; 6. Modified the type “W” of bit [7] T1UPD in section 16.3.6 TIM1_IER (0x406D) as “W1”; 7. Deleted the register T2OPM in Figure 17-1 Output Mode Block Diagram and Figure 17-5 Schematic Diagram of Input Capture Mode 8. Standardized the writing of the registers TIM1_KR and TIM1_KF (section 16 Timer1), TIM2__DR, TIM2__CNTR and TIM2__ARR (section 17 Timer2); 9. Updated descriptions on the bit [1] in section 23.3.5 P1_PU (0x4054); 10. Modified “0.1μF ~ 1μF capacitor” on the bit [VREFEN] in section 27.2.1 VREF_CR(0x404F) as “1μF~4.7μF capacitor”; 11. Updated Figure 31-2 Configurations of LV Reset Threshold, LV Interrupt and LV Warning Threshold and Figure 36-2 Full Code Protection Mode 12. Standardized the document format and modified some descriptions.		

Copyright Notice

Right to make changes — Fortior Technology (Shenzhen) Co., Ltd. reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. The information contained in this manual is provided for the general use by our customers. Our customers shall ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Fortior Technology (Shenzhen) Co., Ltd. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

If there are any differences between the Chinese and the English contents, please take the Chinese version as the standard.

Fortior Technology (Shenzhen) Co., Ltd.
Room 203, 2/F, Building No.11, Keji Central Road 2,
Software Park, High-Tech Industrial Park, Shenzhen, P.R. China 518057
Tel: 0755-26867710
Fax: 0755-26867715
Web: www.fortiortech.com