Fortior Technology

FU6812/61
MCU Embedded and
Configurable 3-Phase BLDC/PMSM
Motor Controller

Datasheet

Table of Contents

1 System overview

1.1 Features

- Power supply:

  FU6812L:

  Single Supply HV Mode (VCC_MODE=0): VCC= 5~24V

  Dual Supply Mode (VCC_MODE=1), VCC≥VDD5: VCC= 5~36V, VDD5=5V

  Single Supply LV Mode (VCC_MODE=1): VCC=VDD5= 3~ 5.5V

  FU6812N/S/P:

  Single Supply HV Mode: VCC= 5~24V

  Single Supply LV Mode: VCC=VDD5= 3~ 5.5V

  FU6861Q:

  Mode 1: VCC_MODE=0, VCC= 5~24V, VDRV=7~18V

  Mode 2: VCC_MODE=1, VCC=VDD5=3~5.5V, VDRV=7~18V

  FU6861N/L:

  Mode 1: VCC=5~24V, VDRV=7~18V
- Dual core: Motor engine ME and 8051 core. ME realize FOC/BLDC control automatically by hardware; 8051 core is used In parameter configuration and daily transaction processing
- Instruction cycles are mostly 1T or 2T.
- 16KB Flash ROM, with CRC verification, support self-write and code protection
- 256 bytes IRAM, 768 bytes XRAM
- ME: Integrated LPF, proportional integrator (PI), BLDC module,FOC module
- 1T 16 x16 multiplier, 16T 32/16 divider
- 4 priority level interrupt, 15 interrupt sources
- GPIO:

  FU6812L: 34 GPIO

  FU6812N: 20 GPIO

  FU6812S: 12 GPIO

  FU6861Q: 32 GPIO

  FU6861N: 19 GPIO

  FU6861L: 27 GPIO

  FU6812P: 21 GPIO
- Timer:

  2 universal programmable capture timers

  1 support QEP decoding programmable timer

  1 BLDC motor dedicated timer

1 general purpose timer

1 RTC timer

- I2C/SPI/UART interface with DMA support
- Analog peripheral:

  12-bit ADC, 0.9us conversion time, select internal VREF, external VREF for reference voltage

  Number of ADC channels:

  FU6812L: 12 channels

  FU6861Q: 12 channels

  FU6812N: 7 channels

  FU6812S: 5 channels

  FU6861N: 9 channels

  FU6861L: 11 channels

  FU6812P: 9 channels

  Built-in VREF reference for configureable 3V, 4V, 4.5V, VDD5 (FU6812S/P can only choose VDD5 for internal reference.

  Built-in VHALF (1/2 VREF) reference output.

  3 independent operational amplifiers (FU6812N/S and FU6861N only have an independent op amp)

  3-channel analog comparator

  8-bit DAC
- Driver type:

  PWM output (for FU6812L/N/S/P)

  6N Predriver (for FU6861Q/N/L)
- BLDC control supports automatic commutation, wave-by-wave current limiting, support for HALL、BEMF position detection.
- FOC driver supports single/dual/three resistors current sampling (FU6812N/S and FU6861N only support single resistor current sampling).
- FOC driver supports overmodulation
- Clock Source

  System Clock Source: Internal clock 24MHz±2%
- Watchdog Time
- Two-wire FICE protocol for on chip debugging

1.2 Application

Sensorless/Sensor BLDC/PMSM, three -phase/single-phase induction motor, servo motor.

Range hoods, indoor units, ceiling fans, floor fans, vacuum cleaners, hair dryers, industrial fans, water pumps, compressors, electric vehicles, electric tools, aircraft models, etc.

1.3 Description

FU6812/61 series is a high performance motor driver dedicated chip integrated with 8051 kernel and motor control engine (ME). 8051 kernel deals with routine affairs, ME deals with motor real-time transactions, and dual-core collaborative work realizes various high performance motor control. Most of the 8051 kernel instruction cycle is 1T or 2T, chip internal integration of high-speed operational amplifier, comparator, high-speed ADC, multiplier, divider, CRC, SPI, I2C, UART, a variety of TIMER, PWM and other functions, built-in high voltage LDO, BLDC/PMSM motor square wave, SVPWM/SPWM, FOC driver control.

The difference between FU6812 and FU6861 please reference charpter Driver, FU6812 is the PWM output. FU6861 is the output of 6N Predriver.

FU6812 has different packages: FU6812L (LQFP48), FU6812N (QFN32), FU6812S (SSOP24), FU6812P (LQFP32).

FU6812 has different packages: FU6861Q (QFN56), FU6861N (QFN40), FU6861L(LQFP48).

For the convenience of description and differentiation, if specific packages are specified later, it means that this feature is exclusive to corresponding packages; otherwise, it is a common feature of FU6812/61 series chips.

1.4 Functional block diagram

1.4.1 FU6812L functional block diagram

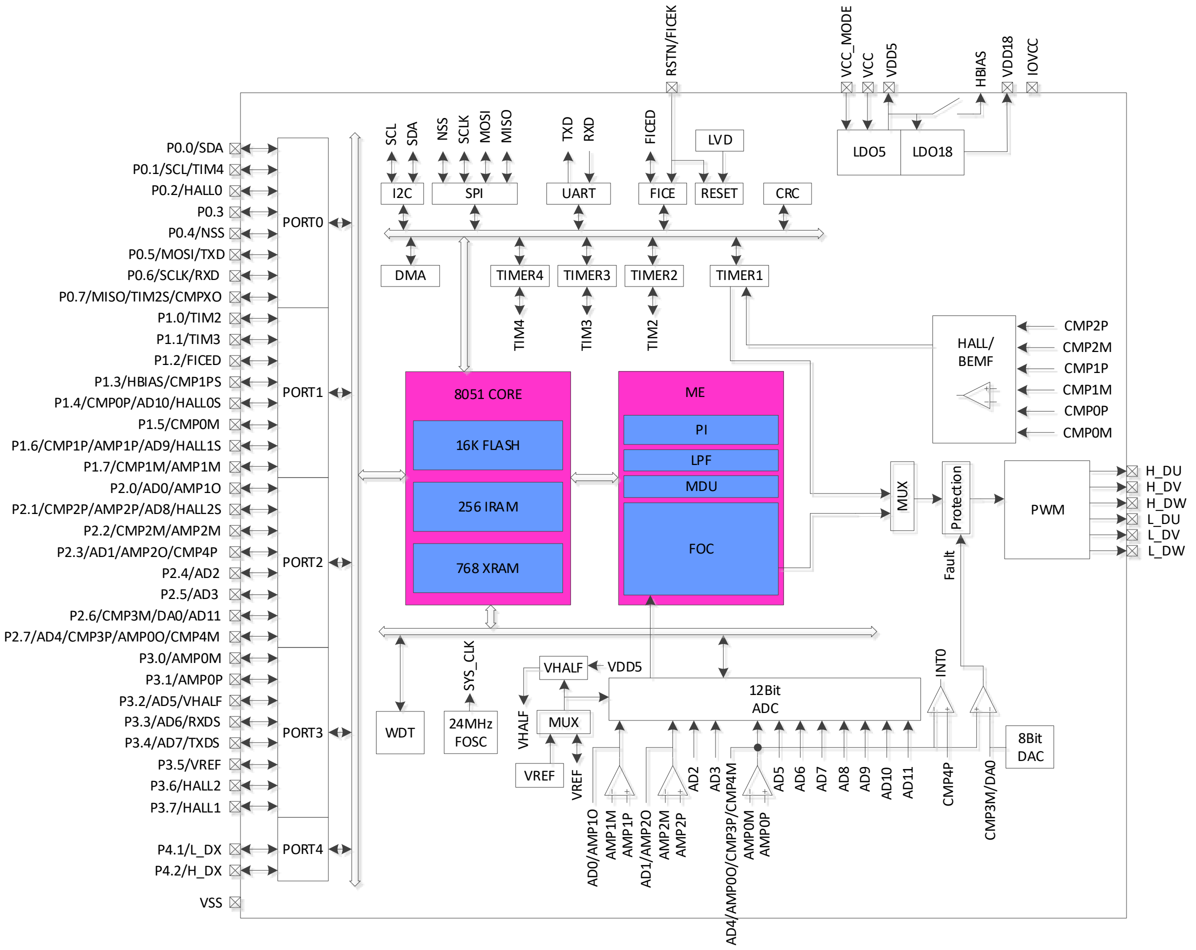


Figure 1-1 Functional block diagram of FU6812L

1.4.2 FU6812N functional block diagram

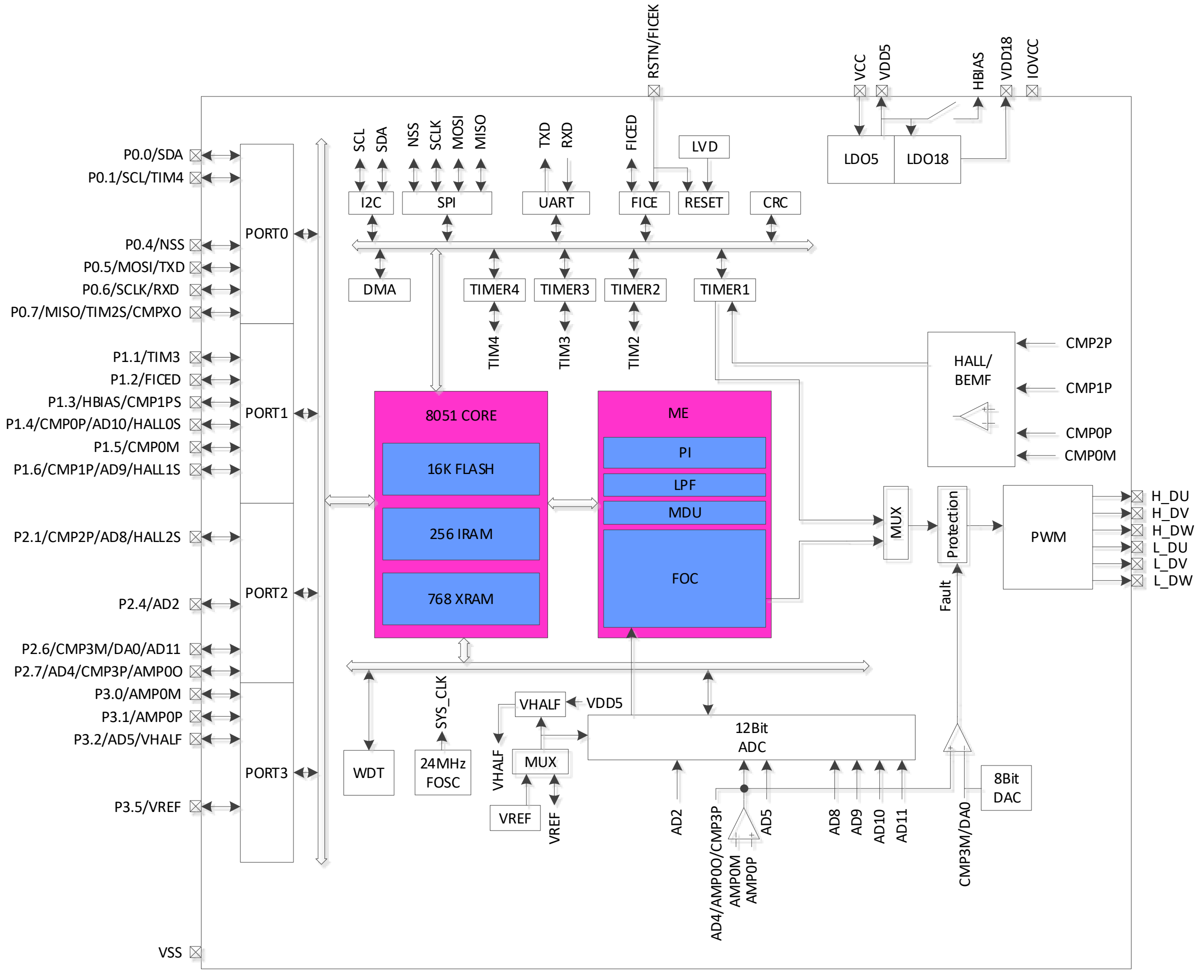


Figure 1-2 Functional block diagram of FU6812N

1.4.3 FU6812S functional block diagram

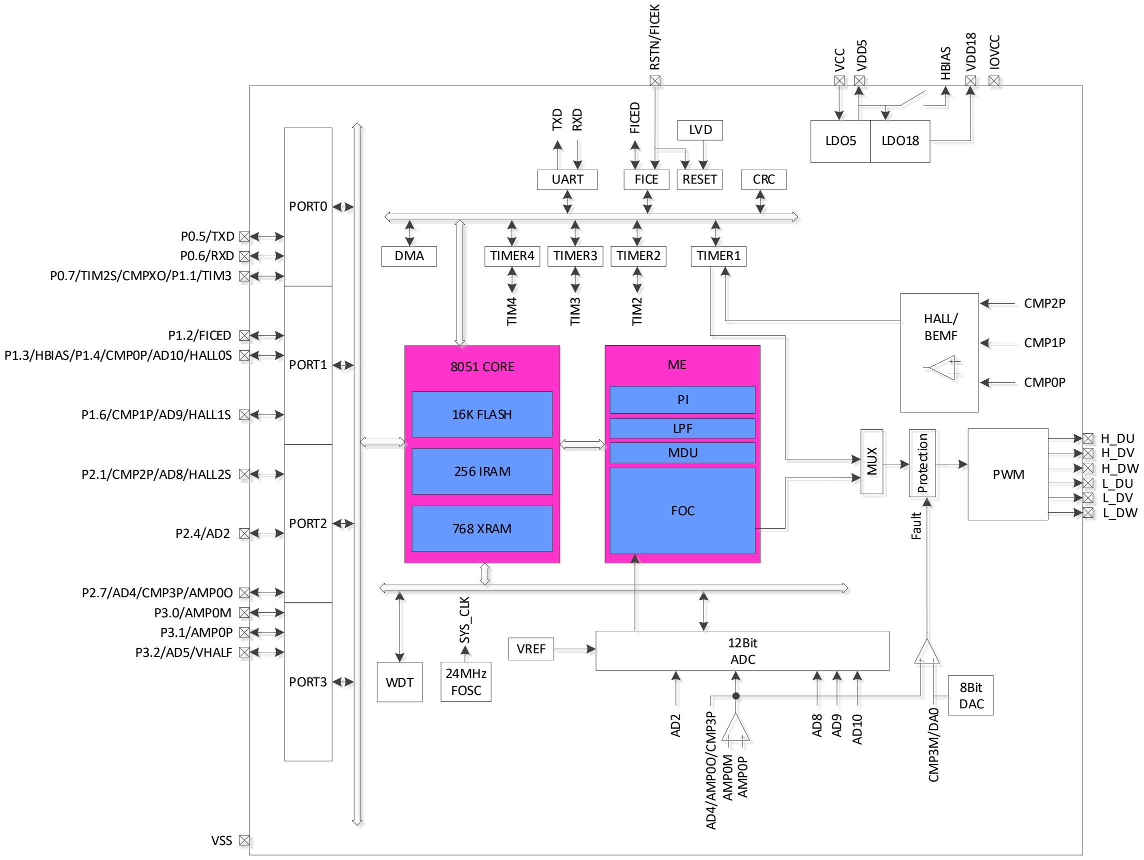


Figure 1-3 Functional block diagram of FU6812S

1.4.4 FU6861Q functional block diagram

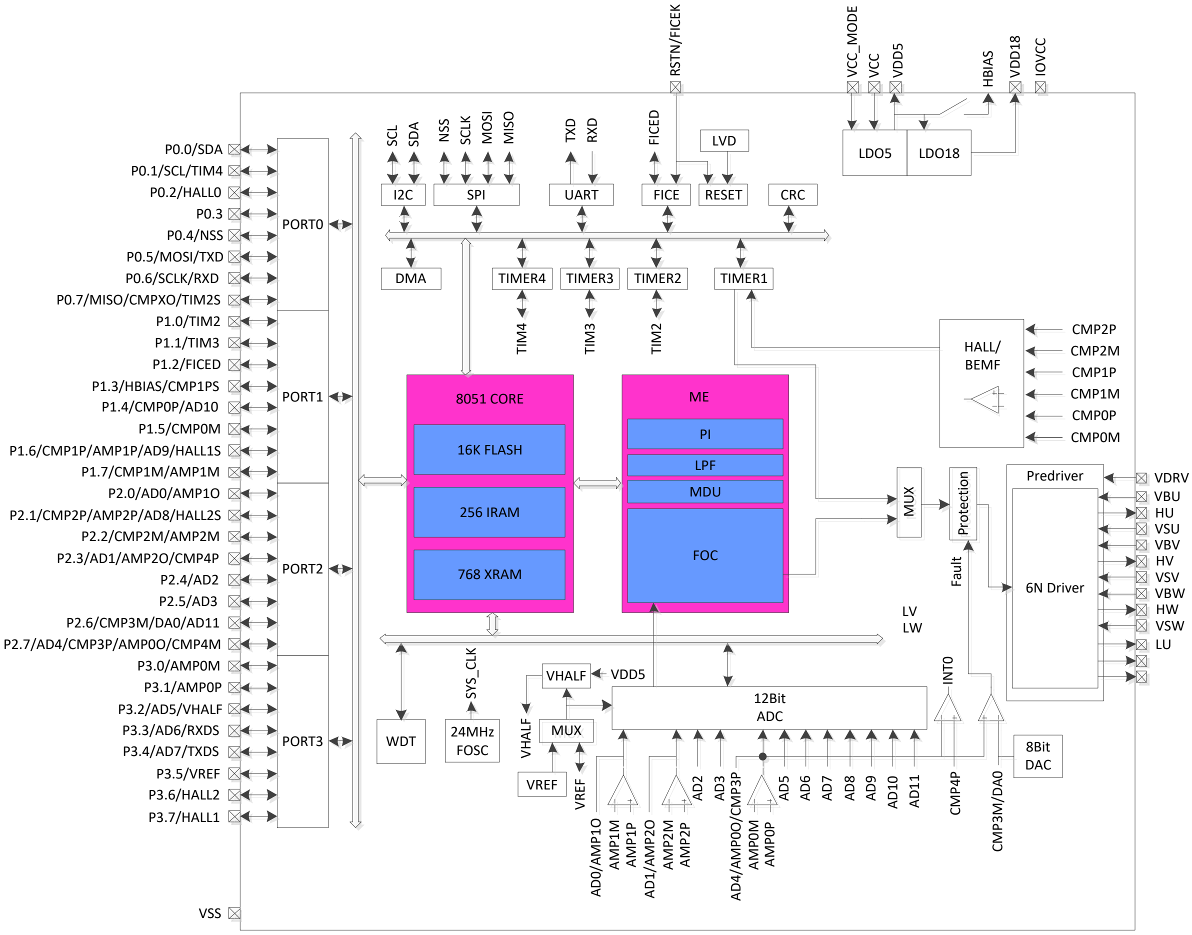


Figure 1-4 Functional block diagram of FU6861Q

1.4.5 FU6861N functional block diagram

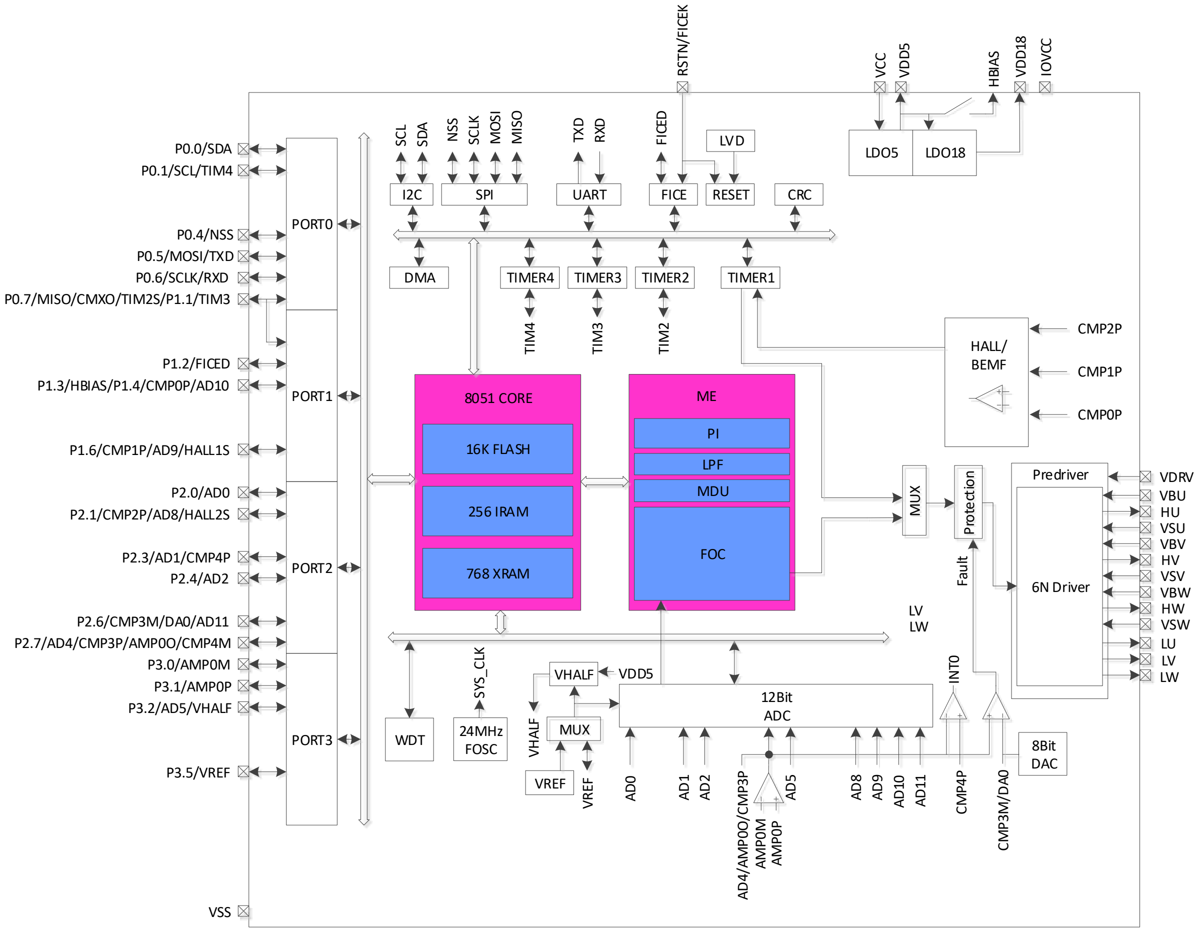


Figure 1-5 Functional block diagram of FU6861N

1.4.6 FU6861L functional block diagram

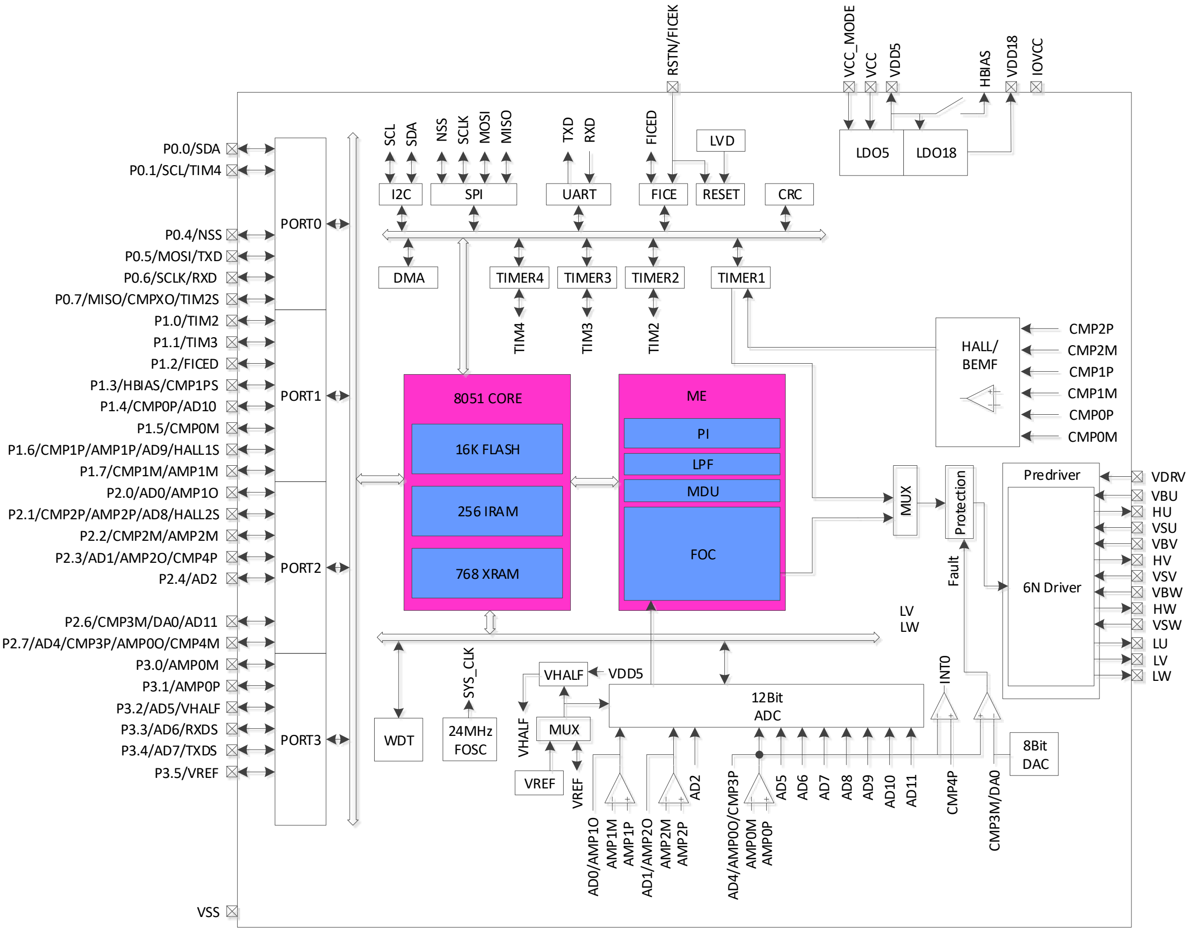


Figure 1-6 Functional block diagram of FU6861L

1.4.7 FU6812P functional block diagram

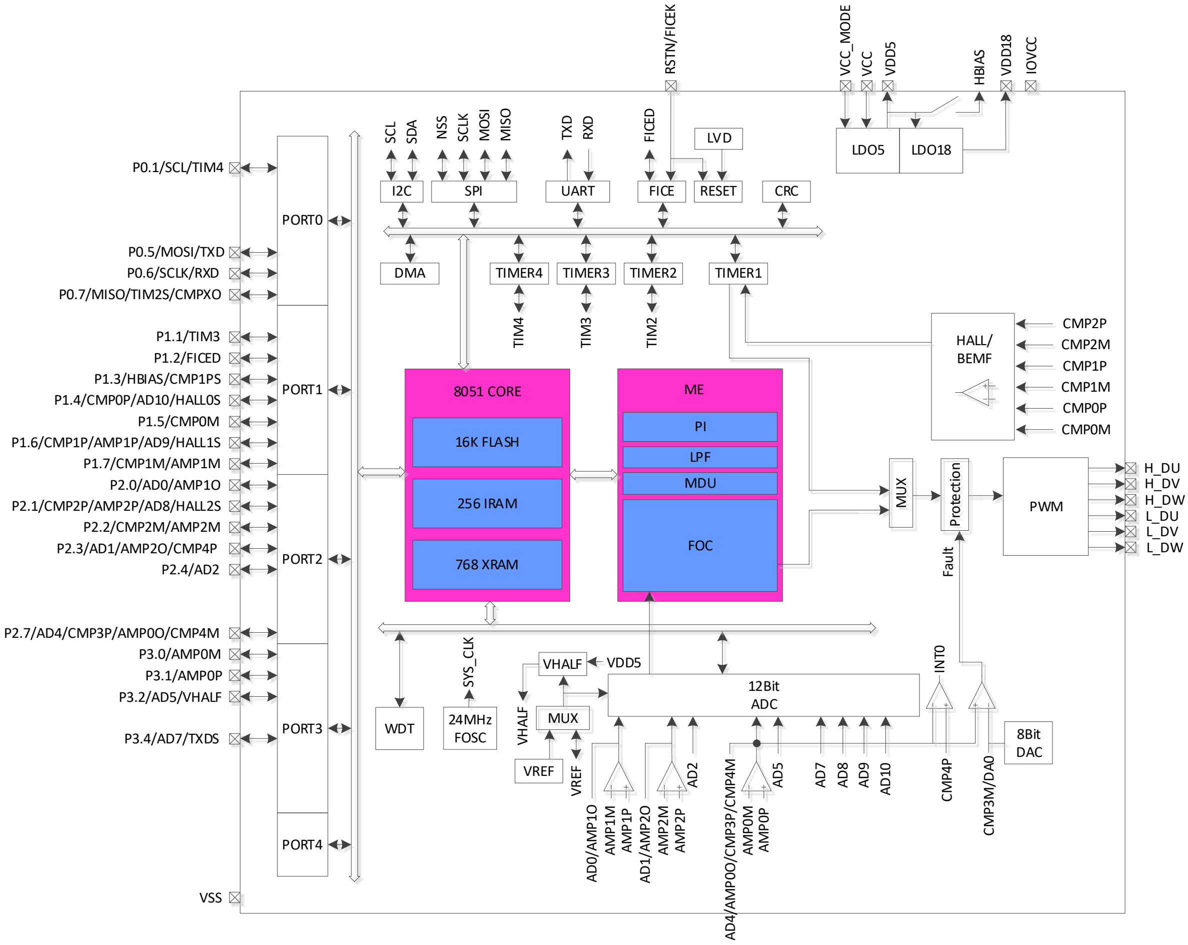


Figure 1-7 Functional block diagram of FU6812P

1.5 Memory organization

The internal storage space is divided into program memory and data memory, which are addressed independently.

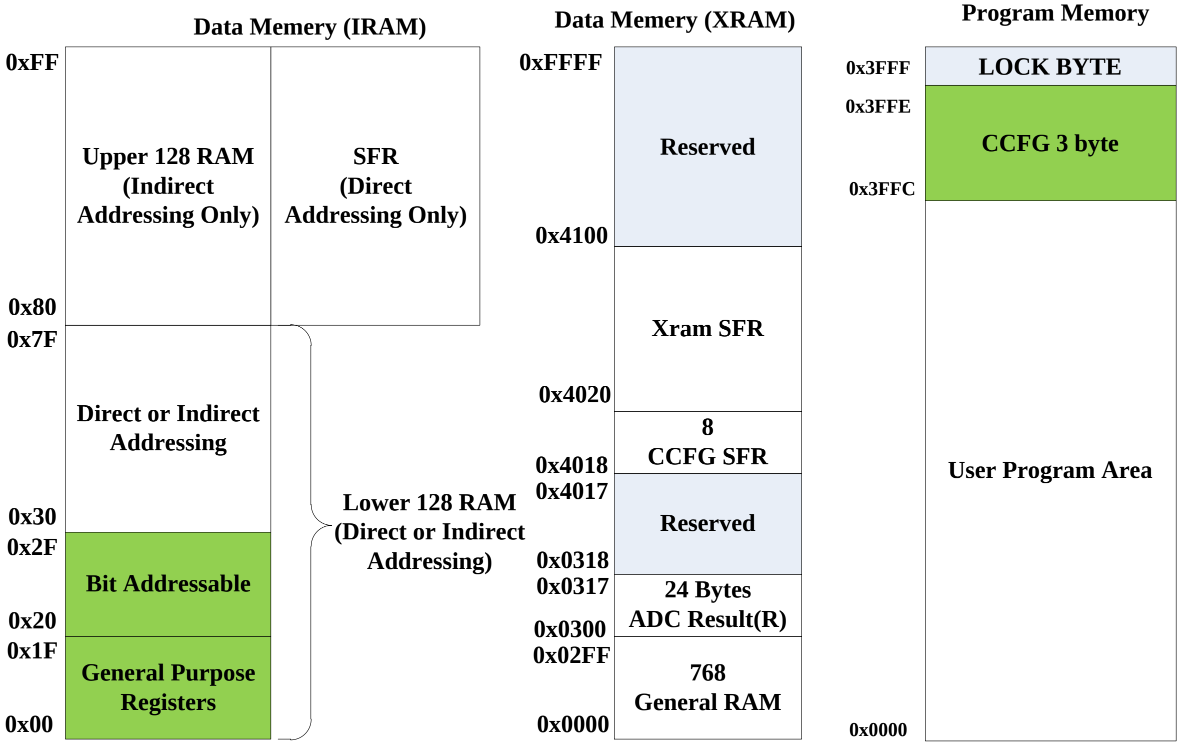


Figure 1-8 Memory space allocation

1.5.1 Program memory

The addressable range of instruction space is 0x0000~0x3FFF. After reset, the CPU starts execution at 0x0000.

1.5.2 Data memory

Data space is divided into external data memory and internal data memory&SFRs.

The external data space is only accessible through the MOVX instruction, with the range 0x0000-0x02FF.

The internal data space is shown in Figure 1-. 0x00-0x1F contains 4 groups registers , each group contains 8 registers; 0x20~0x2F 16Bytes support bit addressing operation; 0x30-0x7F supports direct and indirect addressing; 0x80-0xFF indirectly accesses RAM space and directly accesses SFRs.The stack space is located in the internal data space.

1.5.3 SFR

Table 1-1 Special function register (SFR) address mapping

Addr	0 (8)	1 (9)	2 (A)	3 (B)	4 (C)	5 (D)	6 (E)	7 (F)
0xF8	DRV_OUT	PI_LPF_CR			P0_OE	P1_OE	P2_OE	P3_OE
0xF0	B		PI_KIL	PI_KIH	PI_UKMAXL	PI_UKMAXH	PI_UKMINL	PI_UKMINH
0xE8	P4	P4_OE	PI_EKL	PI_EKH	PI_UKL	PI_UKH	PI_KPL	PI_KPH
0xE0	ACC						LPF_YL	LPF_YH
0xD8	IP3	EVT_FILT	CMP_CR2	LVSR	CMP_CR3	LPF_K	LPF_XL	LPF_XH
0xD0	PSW	P1_IE	P1_IF	P2_IE	P2_IF	CMP_CR0	CMP_CR1	CMP_SR
0xC8	IP2	RST_SR	MDU_MBL	MDU_MBH	MDU_DB0	MDU_DB1		
0xC0	IP1	MDU_CR	MDU_MAL	MDU_MAH	MDU_DA0	MDU_DA1	MDU_DA2	MDU_DA3
0xB8	IP0							
0xB0	P3							
0xA8	IE	TIM2_CR1	TIM2__CNTRL	TIM2__CNTRH	TIM2__DRL	TIM2__DRH	TIM2__ARRL	TIM2__ARRH
0xA0	P2	TIM2_CR0	TIM3__CNTRL	TIM3__CNTRH	TIM3__DRL	TIM3__DRH	TIM3__ARRL	TIM3__ARRH
0x98	UT_CR	UT_DR	UT_BAUDL	UT_BAUDH	TIM3_CR0	TIM3_CR1	TIM4_CR0	TIM4_CR1
0x90	P1		TIM4__CNTRL	TIM4__CNTRH	TIM4__DRL	TIM4__DRH	TIM4__ARRL	TIM4__ARRH
0x88	TCON							
0x80	P0	SP	DPL	DPH	FLA_KEY	FLA_CR		PCON

Notes.

1. Bit is addressable for the SFR when the lower 4 bit is 0 or 8
2. Registers with double underscores need to use a variable to read their value. If the register is read directly, the value is not correct

1.5.4 XSFR

Table 1-2 Extends the special function register (XSFR) address mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x40d8	FOC__POWH	FOC__POWL						
0x40d0	FOC__EALPH	FOC__EALPL	FOC__EBETH	FOC__EBETL	FOC__EOMEH	FOC__EOMEL	FOC__ESQUH	FOC__ESQUL
0x40c8	FOC__IBH	FOC__IBL	FOC__IAH	FOC__IAL	FOC__THETAH	FOC__THETAL	FOC__ETHETAH	FOC__ETHETAL
0x40c0	FOC__IBETH	FOC__IBETL	FOC__VBETH	FOC__VBETL	FOC__VALPH	FOC__VALPL	FOC__ICH	FOC__ICL
0x40b8	FOC__UDH	FOC__UDL	FOC__UQH	FOC__UQL	FOC__IDH	FOC__IDL	FOC__IQH	FOC__IQL
0x40b0	FOC_DMAXH	FOC_DMAXL	FOC_DMINH	FOC_DMINL	FOC_QMAXH	FOC_QMAXL	FOC_QMINH	FOC_QMINL
0x40a8	FOC__RTHESTEPH	FOC__RTHESTEPL	FOC_RTHEACCH	FOC_RTHEACCL	FOC_RTHECNT	FOC_THECOR/CMP_SAMR	FOC_THECOMPH	FOC_THECOMPL
0x40a0	FOC_CR1	FOC_CR2	FOC_TSMIN	FOC_TGLI	FOC_TBLO	FOC_TRGDLY	FOC_CSOH	FOC_CSOL
0x4098	FOC__UDCFLTH/ TIM1__ITRIPH	FOC__UDCFLTL/ TIM1__ITRIPL						
0x4090	FOC_IDREFH	FOC_IDREFL	FOC_IQREFH	FOC_IQREFL	FOC_DQKPH	FOC_DQKPL	FOC_DQKIH	FOC_DQKIL
0x4088	FOC_EK3H/ TIM1__RARRH	FOC_EK3L/ TIM1__RARRL	FOC_EK4H/ TIM1__RCNTRH	FOC_EK4L/ TIM1__RCNTRL	FOC_EK1H	FOC_EK1L	FOC_EK2H	FOC_EK2L
0x4080	FOC_FBASEH/ TIM1_DBR7H	FOC_FBASEL/ TIM1_DBR7L	FOC_EFREQACCH/ TIM1__BCNTRH	FOC_EFREQACCL/ TIM1__BCNTRL	FOC_EFREQMINH/ TIM1__BCCRH	FOC_EFRQMINL/ TIM1__BCCRL	FOC_EFREQHOLDH/ TIM1__BARRH	FOC_EFREQHOLDL/ TIM1__BARRL
0x4078	FOC_KSLIDEH/ TIM1_DBR3H	FOC_KSLIDEL/ TIM1_DBR3L	FOC_EKLPFMINH/ TIM1_DBR4H	FOC_EKLPFMINL/ TIM1_DBR4L	FOC_EBMFKH/ TIM1_DBR5H	FOC_EBMFKL/ TIM1_DBR5L	FOC_OMEKLPFH/ TIM1_DBR6H	FOC_OMEKLPFL/ TIM1_DBR6L
0x4070	TIM1_BCORH	TIM1_BCORL			FOC_EKPH/ TIM1_DBR1H	FOC_EKPL/ TIM1_DBR1L	FOC_EKIH/ TIM1_DBR2H	FOC_EKIL/ TIM1_DBR2L
0x4068	TIM1_CR0	TIM1_CR1	TIM1_CR2	TIM1_CR3	TIM1_CR4	TIM1_ IER	TIM1_SR	
0x4060	DRV_DTR	DRV_SR	DRV_CR		SYST_ARRH	SYST_ARRL		
0x4058	DRV_DRH	DRV_DRL	DRV_COMRH	DRV_COMRL	DRV_CMRH	DRV_CMRL	DRV_ARRH	DRV_ARRL
0x4050	P1_AN	P2_AN	P3_AN	P0_PU	P1_PU	P2_PU	P3_PU	P4_PU
0x4048				DAC_DR	PH_SEL		AMP_CR	VREF_VHALF_CR
0x4040								
0x4038	ADC_SCYC	ADC_CR	DMA0_CR0	DMA1_CR0	DMA0_CR1H	DMA0_CR1L	DMA1_CR1H	DMA1_CR1L
0x4030	SPI_CR0	SPI_CR1	SPI_CLK	SPI_DR		DAC_CR	ADC_MASK_SYSCH	ADC_MASK_SYSCL
0x4028	I2C_CR	I2C_ID	I2C_DR	I2C_SR				
0x4020		CRC_DIN	CRC_CR	CRC_DR	CRC_BEG	CRC_CNT	WDT_CR	WDT_REL
0x4018	CCFG7	CCFG6	CCFG5	CCFG4	CCFG3	CCFG2	CCFG1	CCFG0
0x0310	AD8_DRH	AD8_DRL	AD9_DRH	AD9_DRL	AD10_DRH	AD10_DRL	AD11_DRH	AD11_DRL
0x0308	AD4_DRH	AD4_DRL	AD5_DRH	AD5_DRL	AD6_DRH	AD6_DRL	AD7_DRH	AD7_DRL
0x0300	AD0_DRH	AD0_DRL	AD1_DRH	AD1_DRL	AD2_DRH	AD2_DRL	AD3_DRH	AD3_DRL

Notes

1. Registers with double underscores require a variable to read their value. If the register is read directly, the value is not correct.
2. The chip SFR is divided into two parts, one is mapped in the SFR region of the internal data space and the other is mapped in the external data space.

2 Pin configuration and functions

2.1 FU6812 LQFP48 pin definition

Table 2-1 FU6812L Pin Definition

Name	FU6812 LQFP48	Types	Function
P2.2 / CMP2M / AMP2M	1	DB / AI / AI	GPIO P2.2 BEMF2 minus input AMP2 minus input
P2.3 / AD1 / AMP2O / CMP4P	2	DB / AI / AO / AI	GPIO P2.3 ADC channel 1 input is used to collect the signal amplified by phase current 2 AMP2 output CMP4 positive input
P2.4 / AD2	3	DB / AI	GPIO P2.4 ADC channel 2 input, bus voltage signal input
P2.5 / AD3	4	DB / AI	GPIO P2.5 ADC channel 3 input
P2.6 / CMP3M / DA0 / AD11	5	DB / AI / AO / AI	GPIO P2.6 Overcurrent reference signal input, minus input of CMP3 Internal DAC voltage outputs without buffer ADC channel 11 input
P2.7 / AD4 / CMP3P / AMP0O / CMP4M	6	DB / AI / AI / AO / AI	GPIO P2.7 ADC channel 4 input is used to collect amplified bus current signal The positive input of CMP3 is connected with the bus current sampling signal for detecting over current AMP0 output, output the voltage after amplifing the bus current CMP4 minus input
P3.0 / AMP0M	7	DB / AI	GPIO P3.0 AMP0 minus input for amplifying bus current signal
P3.1 / AMP0P	8	DB / AI	GPIO P3.1 AMP0 positive input for amplifying bus current signal
P3.2 / AD5 / VHALF	9	DB / AI / AO	GPIO P3.2 Overtemperature signal input, ADC channel 5 input 1/2 VDD5 or 1/2 VREF reference output, an external 1uF capacitance is required
P3.3 / AD6 / RXD2	10	DB / AI / DI	GPIO P3.3 ADC channel 6 input UART2 data receiver after function transfer

Name	FU6812 LQFP48	Types	Function
P3.4 / AD7 / TXD2	11	DB AI / DO	GPIO P3.4 Analog speed control input, AD channel 7 input UART2 data transmitter after function transfer
P3.5 / VREF	12	DB / AI	GPIO P3.5 ADC external reference voltage input or internal VREF output, an external 1~4.7uF capacitance is required
VSS	13	P	GND
IOVCC	14	P	GPIO power input, 3~ 5.5v, and connected to an 1~10uF capacitance to the ground, IOVCC is less than or equal to VDD5, only P3.7~6, P0.x, P1.1~0, P4.x, H_DU, H_DV, H_DW, L_DU, L_DV, L_DW with IOVCC power supply, the rest of the GPIO still use VDD5 power supply
P3.6 / HALL2	15	DB / DI	GPIO P3.6 HALL2 logic level input
P3.7 / HALL1	16	DB / DI	GPIO P3.7 HALL1 logic level input
P0.0 / SDA	17	DB / DB	GPIO P0.0 I2C SDA, collector open circuit output, can be configured with 5k pull-up resistance
P0.1 / TIM4 / SCL	18	DB / DB / DB	GPIO P0.1 Timer4 captures mode input I2C SCL clock, collector open circuit output, can be configured with 5k pull-up resistance
P0.2 / HALL0	19	DB / DI	GPIO P0.2 HALL0 logic level input
P0.3	20	DB	GPIO P0.3
P0.4 / NSS	21	DB / DB	GPIO P0.4 SPI's port of choice
P0.5 / TXD / MOSI	22	DB / DO / DB	GPIO P0.5 UART1 data transmitter before function transfer SPI_MOSI, master mode output or slave mode input
P0.6 / RXD / SCLK	23	DB / DI / DB	GPIO P0.6 UART1 data receiver before function transfer SPI interface clock CLK
P0.7 / MISO / CMPXO / TIM2S	24	DB / DB / DO / DB	GPIO P0.7 SPI_MISO, master mode input or slave mode output The test pin of comparator outputs Timer2 captures mode input or PWM mode output after function transfer

Name	FU6812 LQFP48	Types	Function
P1.0 / TIM2	25	DB/ DB	GPIO P1.0 Timer2 captures mode input or PWM mode output before function transfer
P1.1 / TIM3	26	DB/ DB	GPIO P1.1 Timer3 captures mode input
P4.1 / L_DX	27	DB/ DO	GPIO P4.1 PWM output at the lower side of phase X
P4.2 / H_DX	28	DB	GPIO P4.2 PWM output on the upper side of phase X
L_DU	29	DO	PWM output at the lower side of phase U
L_DV	30	DO	PWM output at the lower side of phase V
L_DW	31	DO	PWM output at the lower side of phase W
H_DU	32	DO	PWM output on the upper side of phase U
H_DV	33	DO	PWM output on the upper side of phase V
H_DW	34	DO	PWM output on the upper side of phase W
VCC	35	P	Power input, the voltage range is determined by the power mode VCC_MODE, with an external 10uF or larger filter capacitance. 1. Single-power high-voltage mode: VCC_MODE=0, external power supply from VCC input 5~24V, VDD5 from internal LDO; 2. Single-power low-voltage mode: VCC_MODE=1 (that is, connected with VDD5), the external power supply from VDD5 input 3~ 5.5V, while VCC connect to VDD5; 3. Dual power mode: VCC_MODE=1 (that is, connected with VDD5), external power supply 1 input 5~36V from VCC, external power supply 2 input 5V from VDD5
VSS	36	P	GND
VDD5	37	P	Medium voltage power supply input or internal 5V LDO output power supply, determined by VCC_MODE, power connection method please refer to the description of VCC pin, an external 1~4.7uF capacitance is required. When VCC_MODE=0, the internal LDO outputs 5V power supply; When VCC_MODE=1, 3~ 5.5V power supply is injected from the outside
VCC_MODE	38	DI	Input power mode control, the control mode refers to the VCC pin function description
RSTN / FICEK	39	DI/ DI	External reset input, built-in pull up resistance, schmidt input FICE debugging interface clock port

Name	FU6812 LQFP48	Types	Function
VDD18	40	P	1.8V LDO output power, an external 1~4.7uF capacitor is required
P1.2 / FICED	41	DB/ DB	GPIO P1.2 FICE data port
P1.3 / HBIAS / CMP1PS	42	DB/ DO/ AI	GPIO P1.3 HALL bias power supply, connects to VDD5 by a switch CMP1 second mode positive input
P1.4 / CMP0P / AD10 / HALL0S	43	DB/ AI/ AI/ DI	GPIO P1.4 BEMF0 positive input ADC channel 10 input HALL0 logic level input after function transfer
P1.5 / CMP0M	44	DB / AI	GPIO P1.5 BEMF0 minus input
P1.6 / CMP1P / AMP1P / AD9 / HALL1S	45	DB / AI / AI / AI / DI	GPIO P1.6 BEMF1 positive input AMP1 positive input, connected with phase current 1 voltage signal input ADC channel 9 input HALL1 logic level input after function transfer
P1.7 / CMP1M / AMP1M	46	DB / AI / AI	GPIO P1.7 BEMF1 minus input AMP1 minus input
P2.0 / AD0 / AMP1O	47	DB / AI / AO	GPIO P2.0 ADC channel 0 input is used to collect the amplified phase current 1 signal AMP1 output
P2.1 / CMP2P / AMP2P / AD8 / HALL2S	48	DB / AI / AI / AI / DI	GPIO P2.1 BEMF2 positive input AMP2 positive input, phase current 2 voltage signal input ADC channel 8 input HALL2 logic level input after function transfer

Notes:

IO type description:

DI = Digital input,

DO = Digital output,

DB = Digital bidirectional,

AI = Analog input,

AO = Analog output,

P = Power supply

2.2 FU6812L package-LQFP48

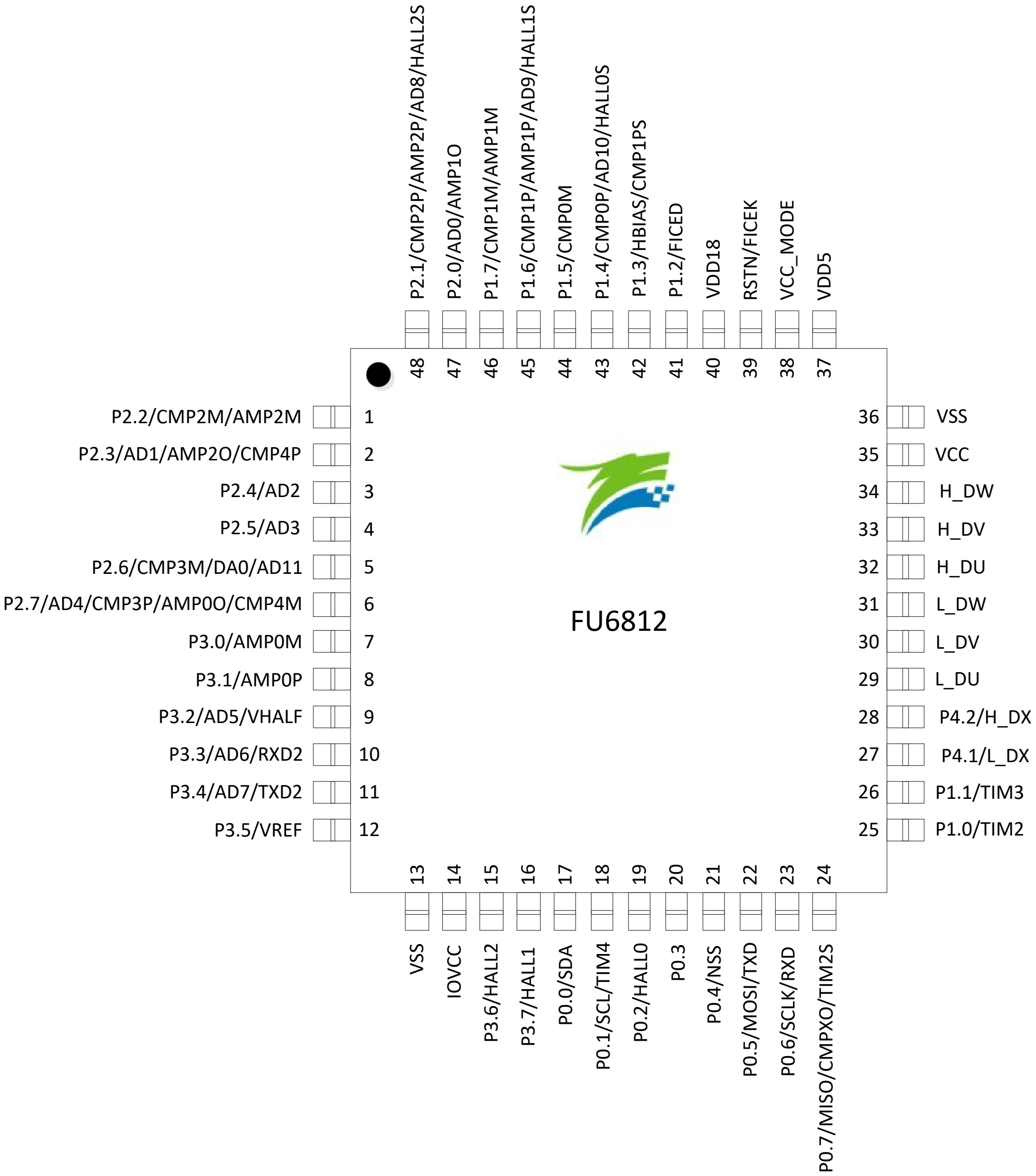


Figure 2-1 FU6812 package-LQFP48

2.3 FU6812 QFN32 pin definition

Table 2-2 FU6812N Pin Definition

Name	FU6812 QFN32	Types	Function
P2.1 / CMP2P / AD8 / HALL2S	1	DB / AI / AI / DI	GPIO P2.1 BEMF2 positive input ADC channel 8 input HALL2 logic level input after function transfer
P2.4 / AD2	2	DB / AI	GPIO P2.4 ADC channel 2 input, bus voltage signal input
P2.6 / CMP3M / DA0 / AD11	3	DB / AI / AO / AI	GPIO P2.6 Overcurrent reference signal input, minus input of CMP3 Internal DAC voltage outputs without buffer ADC channel 11 input
P2.7 / AD4 / CMP3P / AMP0O	4	DB / AI / AI / AO	GPIO P2.7 ADC channel 4 input is used to collect amplified bus current signal CMP3 positive input , connect with the bus current sampling signal for detecting over current AMP0 output, output the voltage after amplifing the bus current
P3.0 / AMP0M	5	DB / AI	GPIO P3.0 AMP0 minus input for amplifying bus current signal
P3.1 / AMP0P	6	DB / AI	GPIO P3.1 AMP0 positive input for amplifying bus current signal
P3.2 / AD5 / VHALF	7	DB / AI / AO	GPIO P3.2 Overtemperature signal input,ADC channel 5 input 1/2 VDD5 or 1/2 VREF reference output, an external 1uF capacitance is required
P3.5 / VREF	8	DB / AI	GPIO P3.5 ADC external reference voltage input or internal VREF output, an external 1~4.7uF capacitance is required
IOVCC	9	P	GPIO power input, 3~ 5.5V, and connected with an 1~10uF capacitance to the ground, IOVCC is less than or equal to VDD5, only P0.x, P1.1, H_DU, H_DV, H_DW, L_DU, L_DV, L_DW with IOVCC power supply, the rest of the GPIO still use VDD5 power supply
P0.0 / SDA	10	DB / DB	GPIO P0.0 I2C SDA, collector open circuit output, can be configured with 5k pull-up resistance

Name	FU6812 QFN32	Types	Function
P0.1 / TIM4 / SCL	11	DB / DB / DB	GPIO P0.1 Timer4 captures mode input I2C SCL clock, collector open circuit output, can be configured with 5k pull-up resistance
P0.4 / NSS	12	DB / DB	GPIO P0.4 SPI's port of choice
P0.5 / TXD / MOSI	13	DB / DO / DB	GPIO P0.5 UART1 data transmitter before function transfer SPI_MOSI, master mode output or slave mode input
P0.6 / RXD / SCLK	14	DB / DI / DB	GPIO P0.6 UART1 data receiver before function transfer SPI interface clock CLK
P0.7 / MISO / CMPXO / TIM2S	15	DB / DB / DO / DB	GPIO P0.7 SPI_MISO, master mode input or slave mode output The test pin of the CMP outputs Timer2 captures mode input or PWM mode output after function transferr
P1.1 / TIM3	16	DB / DB	GPIO P1.1 Timer3 captures mode input
L_DU	17	DO	PWM output at the lower side of phase U
L_DV	18	DO	PWM output at the lower side of phase V
L_DW	19	DO	PWM output at the lower side of phase W
H_DU	20	DO	PWM output at the upper side of phase U
H_DV	21	DO	PWM output at the upper side of phase V
H_DW	22	DO	PWM output at the upper side of phase W
VCC	23	P	Power input, an external 10uF or greater filter capacitance needed. 1. Single-power high-voltage mode: External power supply from VCC input 5~24V, VDD5 from the internal LDO; 2. Single-power low-voltage mode: External power supply from VDD5 input 3~ 5.5v, at the same time VCC and VDD5 short;
VSS	24	P	GND
VDD5	25	P	Medium voltage power supply input or internal 5V LDO output power, power connection method please refer to the description of VCC pin, an external a 1~4.7uF capacitance is required. Medium voltage power input or internal 5V LDO output power, connected with a 1~4.7uF capacitor. When VCC> is 5.5V, VDD5 outputs 5V. When VCC is 3~ 5.5V, VCC and VDD5 are shorted together.

Name	FU6812 QFN32	Types	Function
RSTN / FICEK	26	DI / DI	External reset input, built-in pull up resistance, schmidt input FICE debugging interface clock port
VDD18	27	P	1.8V LDO output power, an external 1~4.7uF capacitor is required
P1.2 / FICED	28	DB / DB	GPIO P1.2 FICE data port
P1.3 / HBIAS / CMP1PS	29	DB / DO / AI	GPIO P1.3 HALL bias power supply, connects to VDD5 by a switch CMP1 second mode positive input
P1.4 / CMP0P / AD10 / HALL0S	30	DB / AI / AI / DI	GPIO P1.4 BEMF0 positive input ADC channel 10 input HALL0 logic level input after function transfer
P1.5 / CMP0M	31	DB / AI	GPIO P1.5 BEMF0 minus input
P1.6 / CMP1P / AD9 / HALL1S	32	DB / AI / AI / DI	GPIO P1.6 BEMF1 positive input ADC channel 9 input HALL1 logic level input after function transfer

Notes:

IO type description:

DI = Digital input,

DO = Digital output,

DB = Digital bidirectional

AI = Analog input,

AO = Analog output,

P = Power supply

2.4 FU6812N package-QFN32

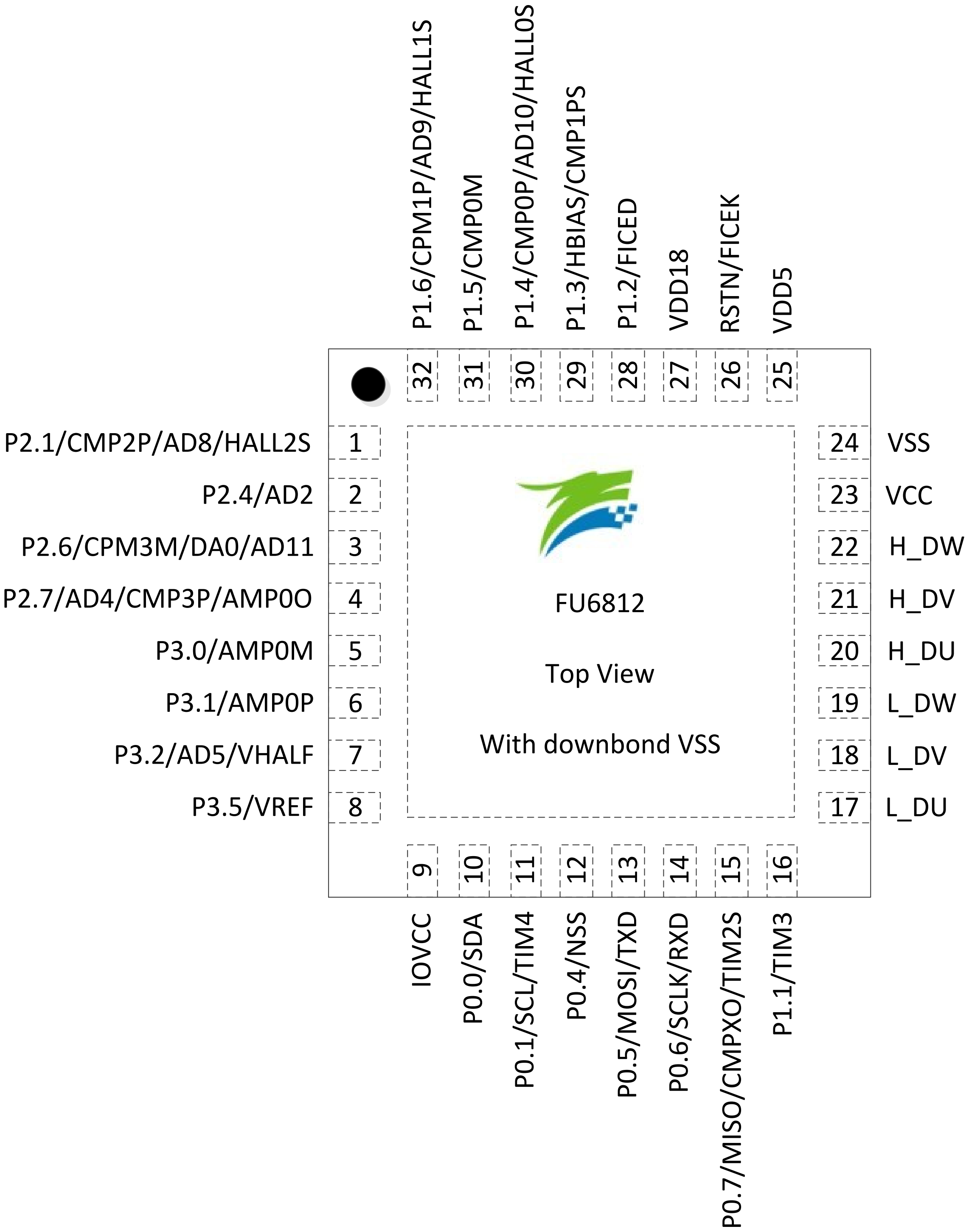


Figure 2-2 FU6812 package-QFN32

2.5 FU6861 QFN56 pin definition

Table 2-3 FU6861Q Pin Definition

Name	FU6861 QFN56	Types	Function
VSU	1	P	6N Predriver U phase input for U phase upper side bootstrap ground reference
HU	2	DO	6N Predriver U phase upper side PWM output
VBU	3	P	6N Predriver U phase upper side bootstrap power supply
VSV	4	P	6N Predriver V phase input, for V phase upper side bootstrap ground reference
HV	5	DO	6N Predriver V phase upper PWM output
VBV	6	P	6N Predriver V phase upper side bootstrap power supply
VSW	7	P	6N Predriver W phase input for the ground end reference of the upper side bootstrap of W phase
HW	8	DO	6N Predriver W phase upper PWM output
VBW	9	P	6N Predriver W phase upper side bootstrap power supply
VCC	10	P	Power input, the voltage range is determined by the power mode VCC_MODE, with an external 10uF or larger filter capacitance. 1. Single-power high-voltage mode: VCC_MODE=0, external power supply from VCC input 5~24V, VDD5 from internal LDO; 2. Single-power low-voltage mode: VCC_MODE=1 (that is, connected with VDD5), the external power supply from VDD5 input 3~ 5.5V, while VCC and VDD5 short; 3. Dual power mode: VCC_MODE=1 (that is, connected with VDD5), external power supply 1 input 5~36V from VCC, external power supply 2 input 5V from VDD5
VSS	11	P	Digital GND
VDD5	12	P	Medium voltage power supply input or internal 5V LDO output power supply, determined by VCC_MODE, power connection method please refer to the description of VCC pin, an external connection 1~4.7uF capacitance is required. When VCC_MODE=0, the internal LDO outputs 5V power supply; When VCC_MODE=1, 3~ 5.5V power supply is injected from the outside
VCC_MODE	13	DI	Input power mode control, the control mode refers to the VCC pin function description
RSTN / FICEK	14	DI / DI	External reset input, built-in pull up resistance, schmidt input FICE debugging interface clock port

Name	FU6861 QFN56	Types	Function
VDD18	15	P	1.8V LDO output power, an external 1~4.7uF capacitor is required
AVSS	16	P	Analog GND
P1.2 / FICED	17	DB / DB	GPIO P1.2 FICE data porta
P1.3 / HBIAS / CMP1PS	18	DB / DO / AI	GPIO P1.3 HALL bias power supply, which connect to VDD5 by a switch CMP1 second mode positive input
P1.4 / CMP0P / AD10 / HALL0S	19	DB / AI / AI / DI	GPIO P1.4 BEMF0 positive input ADC channel 10 input HALL0 logic level input after function transfer
P1.5 / CMP0M	20	DB / AI	GPIO P1.5 BEMF0 negative input
P1.6 / CMP1P / AMP1P / AD9 / HALL1S	21	DB / AI / AI / AI / DI	GPIO P1.6 BEMF1 positive input AMP1 positive input port, connected with phase current 1 voltage signal input ADC channel 9 input HALL1 logic level input after function transfer
P1.7 / CMP1M / AMP1M	22	DB / AI / AI	GPIO P1.7 BEMF1 minus input AMP1 minus input port
P2.0 / AD0 / AMP1O	23	DB / AI / AO	GPIO P2.0 ADC channel 0 input is used to collect the amplified phase current 1 signal AMP1 output
P2.1 / CMP2P / AMP2P / AD8 / HALL2S	24	DB / AI / AI / AI / DI	GPIO P2.1 BEMF2 positive input AMP2 positive input, phase current 2 voltage signal input ADC channel 8 input HALL2 logic level input after function transfer
P2.2 / CMP2M / AMP2M	25	DB / AI / AI	GPIO P2.2 BEMF2 minus input AMP2 minus input
P2.3 / AD1 / AMP2O / CMP4P	26	DB / AI / AO / AI	GPIO P2.3 ADC channel 1 input, which is used to collect the amplified phase current 2 signal AMP2 output port CMP4 positive input
P2.4 / AD2	27	DB / AI	GPIO P2.4 ADC channel 2 input, bus voltage signal input

Name	FU6861 QFN56	Types	Function
P2.5 / AD3	28	DB / AI	GPIO P2.5 ADC channel 3 input
P2.6 / CMP3M / DA0 / AD11	29	DB / AI / AO / AI	GPIO P2.6 Overcurrent reference signal input, the minus input end of CMP3 Internal DAC voltage outputs without buffer ADC channel 11 input
P2.7 / AD4 / CMP3P / AMP0O / CMP4M	30	DB / AI / AI / AO / AI	GPIO P2.7 ADC channel 4 input,which is used to collect amplified bus current signal The positive input of CMP3, which can connecte with the bus current sampling signal for detecting over current AMP0 output, output the voltage after amplifying the bus current CMP4 minus input
P3.0 / AMP0M	31	DB / AI	GPIO P3.0 AMP0 minus input for amplifying bus current signal
P3.1 / AMP0P	32	DB / AI	GPIO P3.1 AMP0 positive input for amplifying bus current signal
P3.2 / AD5 / VHALF	33	DB / AI / AO	GPIO P3.2 Overtemperature signal input,ADC channel 5 input 1/2 VDD5 or 1/2 VREF reference output, external 1uF capacitance
P3.3 / AD6 / RXD2	34	DB / AI / DI	GPIO P3.3 ADC channel 6 input UART2 data receiver after function transfer
P3.4 / AD7 / TXD2	35	DB AI / DO	GPIO P3.4 Analog speed control input, AD channel 7 input UART2 data transmitter after function transfer
P3.5 / VREF	36	DB / AI	GPIO P3.5 ADC external reference voltage input or internal VREF output, an external 1~4.7uF capacitance is required
VSS	37	P	GND
IOVCC	38	P	GPIO power input, 3~ 5.5v, which should connect with a 1~10uF capacitance to the ground, IOVCC is less than or equal to VDD5, only P3.7~6, P0.x, P1.1~0 with IOVCC power supply, the rest of the GPIO is still powered by VDD5
P3.6 / HALL2	39	DB / DI	GPIO P3.6 HALL2 logic level input
P3.7 / HALL1	40	DB / DI	GPIO P3.7 HALL1 logic level input
P0.0 / SDA	41	DB / DB	GPIO P0.0 I2C SDA, collector open circuit output, can be configured with 5k pull-up resistance

Name	FU6861 QFN56	Types	Function
P0.1 / TIM4 / SCL	42	DB / DB / DB	GPIO P0.1 Timer4 captures mode input I2C SCL clock, collector open circuit output, can be configured with 5k pull-up resistance
P0.2 / HALL0	43	DB / DI	GPIO P0.2 HALL0 logic level input
P0.3	44	DB	GPIO P0.3
P0.4 / NSS	45	DB / DB	GPIO P0.4 SPI's port of choice
P0.5 / TXD / MOSI	46	DB / DO / DB	GPIO P0.5 UART1 data transmitter before function transfer SPI_MOSI, master mode output or slave mode input
P0.6 / RXD / SCLK	47	DB / DI / DB	GPIO P0.6 UART1 data receiver before function transfer SPI interface clock CLK
P0.7 / MISO / CMPXO / TIM2S	48	DB / DB / DO / DB	GPIO P0.7 SPI_MISO, master mode input or slave mode output The test pin of the comparator outputs Timer2 captures mode input or PWM mode output after function transferr
P1.0 / TIM2	49	DB / DB	GPIO P1.0 Timer2 captures mode input or PWM mode output before function transfer
P1.1 / TIM3	50	DB / DB	GPIO P1.1 Timer3 captures mode input
VDRV	51	P	6N Predriver power supply, 7~18V, an external 1~10uF capacitance is required
VSS	52	P	GND
NC	53		NC Pin, dangling
LU	54	DO	6N Predriver U phase lower side PWM output
LV	55	DO	6N Predriver V phase lower side PWM output
LW	56	DO	6N Predriver W phase lower side PWM output

Notes:

IO type description:

DI = Digital input,

DO = Digital output,

DB = Digital bidirectional

AI = Analog input,

AO = Analog output,

P = Power supply

2.6 FU6861Q package-QFN56

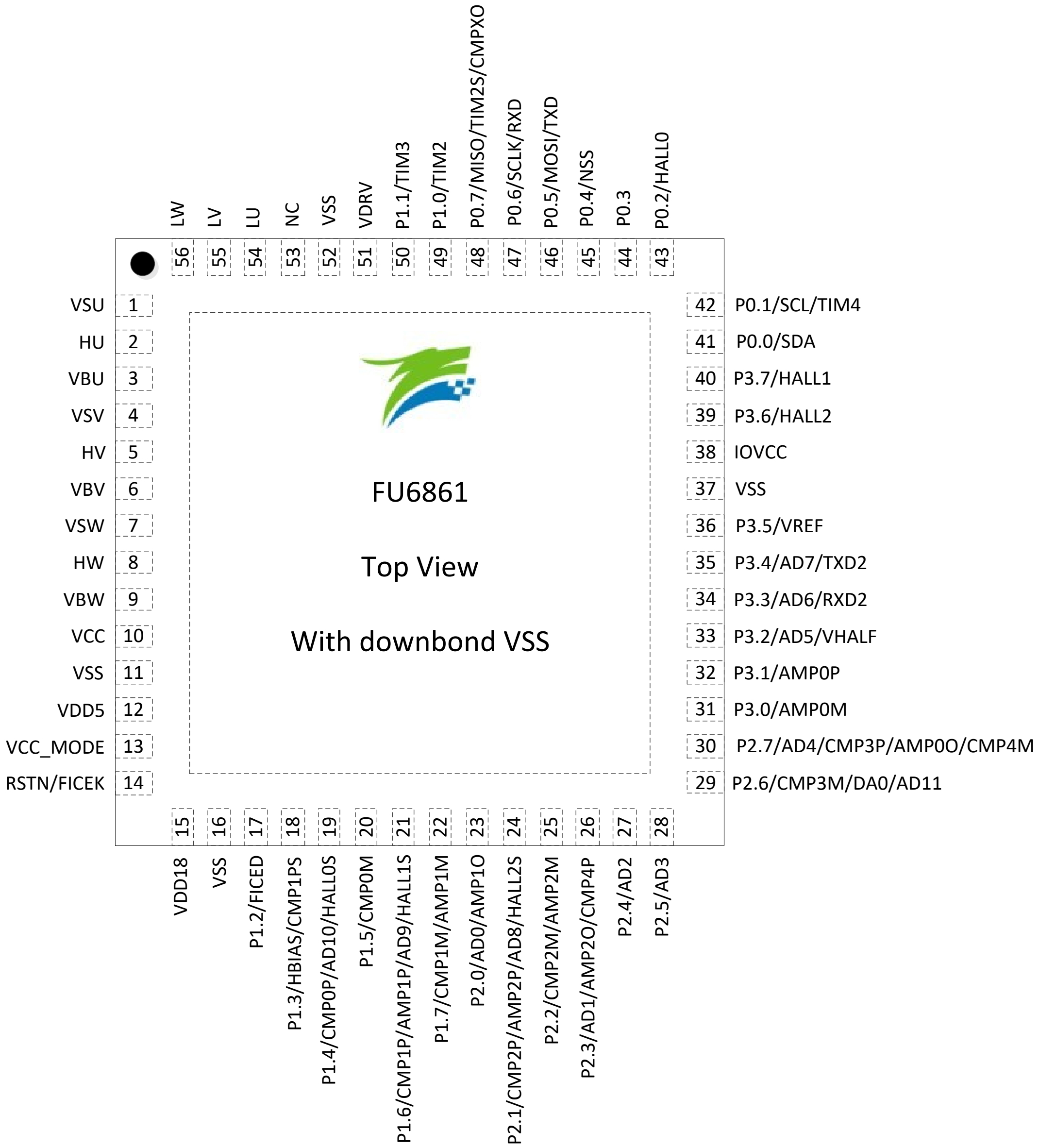


Figure 2-3 FU6861 package-QFN56

2.7 FU6812 SSOP24 pin definition

Table 2-4 FU6812S Pin Definition

Name	FU6812 SSOP24	Types	Function
P0.6 / RXD	1	DB / DI	GPIO P0.6 UART1 data receiver before function transfer
P0.7 / CMPXO / TIM2S / P1.1 / TIM3	2	DB / DO / DB / DB / DB	GPIO P0.7 The test pin of the comparator outputs Timer2 captures mode input or PWM mode output after functiona transferre GPIO P1.1, configurable for external interrupt 1 input Timer3 captures mode input
L_DU	3	DO	PWM output at the lower side of phase U
L_DV	4	DO	PWM output at the lower side of phase V
L_DW	5	DO	PWM output at the lower side of phase W
H_DU	6	DO	PWM output on the upper side of phase U
H_DV	7	DO	PWM output on the upper side of phase V
H_DW	8	DO	PWM output on the upper side of phase W
VCC	9	P	Power input, external 10uF or greater filter capacitance. 1. Single-power high-voltage mode: External power supply from VCC input 5~24V, VDD5 from the internal LDO;
VSS	10	P	Digital GND
VDD5	11	P	Medium voltage power supply input or internal 5V LDO output power, power connection method please refer to the description of VCC pin, an external connection 1~4.7uF capacitance is required.
RSTN / FICEK	12	DI / DI	External reset input, built-in pull up resistance, schmidt input FICE debugging interface clock port
VDD18	13	P	1.8V LDO output power, external 1~ 4.7uF capacitor
P1.2 / FICED	14	DB / DB	GPIO P1.2 FICE data port
P1.3 / HBIAS / P1.4 / CMP0P / AD10 / HALL0S	15	DB / DO / DB / AI / AI / DI	GPIO P1.3 HALL bias power supply, which connect to VDD5 by a switch. GPIO P1.4 BEMF0 positive input ADC channel 10 input HALL0 logic level input after function transfer
P1.6 / CMP1P / AD9 / HALL1S	16	DB / AI / AI / DI	GPIO P1.6 BEMF1 positive input ADC channel 9 input HALL1 logic level input after function transfer

Name	FU6812 SSOP24	Types	Function
P2.1 / CMP2P / AD8 / HALL2S	17	DB / AI / AI / DI	GPIO P2.1 BEMF2 positive input ADC channel 8 input HALL2 logic level input after function transfer
P2.4 / AD2	18	DB / AI	GPIO P2.4 ADC channel 2 input, bus voltage signal input
P2.7 / AD4 / CMP3P / AMP0O	19	DB / AI / AI / AO	GPIO P2.7 ADC channel 4 input, used to collect amplified bus current signal CMP3 positive input, which is connected with the bus current sampling signal for detecting over current AMP0 output, output the voltage after amplifying the bus current
P3.0 / AMP0M	20	DB / AI	GPIO P3.0 AMP0 minus input for amplifying bus current signal
P3.1 / AMP0P	21	DB / AI	GPIO P3.1 AMP0 positive input for amplifying bus current signal
P3.2 / AD5 / VHALF	22	DB / AI / AO	GPIO P3.2 Overtemperature signal input, ADC channel 5 input 1/2 VDD5 or 1/2 VREF reference output, an external 1uF capacitance is required
IOVCC	23	P	IO power input, 3~5.5v, and connect 1~10uF capacitance to the ground, IOVCC is less than or equal to VDD5
P0.5 / TXD	24	DB / DO	GPIO P0.5 UART1 data transmitter before function transfer

Notes:

IO type description:

DI = Digital input,

DO = Digital output,

DB = Digital bidirectional

AI = Analog input,

AO = Analog output,

P = Power supply

2.8 FU6812S package-SSOP24

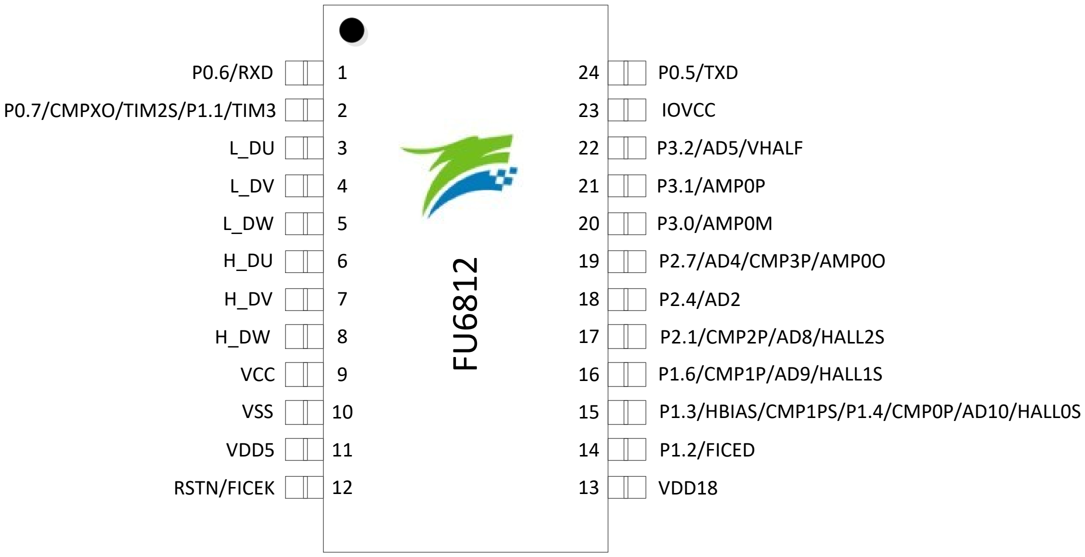


Figure 2-4 FU6812 SSOP24 package

2.9 FU6861N QFN40 pin definition

Table 2-5 FU6861Q Pin Definition

Name	FU6861 QFN40	Types	Function
VSU	1	P	6N Predriver U phase input for U phase upper side bootstrap ground reference
HU,	2	DO	6N Predriver U phase upper side PWM output
VBU	3	P	6N Predriver U phase upper side bootstrap power supply
VSV	4	P	6N Predriver V phase input, for V phase upper side bootstrap ground reference
HV	5	DO	6N Predriver V phase upper side PWM output
VBV	6	P	6N Predriver V phase upper side bootstrap power supply
VSW	7	P	6N Predriver W phase input for the ground end reference of the upper side bootstrap of W phase
HW	8	DO	6N Predriver W phase upper side PWM output
VBW	9	P	6N Predriver W phase upper side bootstrap power supply
VCC	10	P	Power input, the voltage range is determined by the power mode VCC_MODE, with an external 10uF or larger filter capacitance.
VSS	11	P	Digital GND
VDD5	12	P	Medium voltage power supply input or internal 5V LDO output power supply, determined by VCC_MODE, power connection method please refer to the description of VCC pin, an external connection 1~4.7uF capacitance is required.
RSTN / FICEK	13	DI / DI	External reset input, built-in pull up resistance, schmidt input FICE debugging interface clock port
VDD18	14	P	1.8V LDO output power, an external 1~4.7uF capacitor is required
P1.2 / FICED	15	DB / DB	GPIO P1.2 FICE data porta
P1.3 / HBIAS / P1.4 / CMP0P / AD10 / HALL0S	16	DB / DO / DB / AI / AI / DI	GPIO P1.3 HALL bias power supply, which connect to VDD5 by a switch GPIO P1.4, can be configured with external interrupt 1 input, can be configured with 5k pull-up resistance BEMF0 positive input ADC channel 10 input HALL0 logic level input after function transfer
P1.6 / CMP1P / AD9 / HALL1S	17	DB / AI / AI / DI	GPIO P1.6 BEMF1 positive input ADC channel 9 input HALL1 logic level input after function transfer

Name	FU6861 QFN40	Types	Function
P2.0 / AD0 /	18	DB / AI /	GPIO P2.0 ADC channel 0 input is used to sample the amplified phase current 1 signal
P2.1 / CMP2P / AD8 / HALL2S	19	DB / AI / AI / DI	GPIO P2.1 BEMF2 positive input ADC channel 8 input HALL2 logic level input after function transfer
P2.3 / AD1 / CMP4P	20	DB / AI / AI	GPIO P2.3 ADC channel 1 input, which is used to sample the amplified phase current 2 signal CMP4 positive input
P2.4 / AD2	21	DB / AI	GPIO P2.4 ADC channel 2 input, bus voltage signal input
P2.6 / CMP3M / DA0 / AD11	22	DB / AI / AO / AI	GPIO P2.6 Overcurrent reference signal input, the minus input of CMP3 Internal DAC voltage outputs without buffer ADC channel 11 input
P2.7 / AD4 / CMP3P / AMP0O / CMP4M	23	DB / AI / AI / AO / AI	GPIO P2.7 ADC channel 4 input,which is used to sample amplified bus current signal The positive input of CMP3, which can connecte with the bus current sampling signal for detecting over current AMP0 output, output the voltage after amplifying the bus current CMP4 minus input
P3.0 / AMP0M	24	DB / AI	GPIO P3.0 AMP0 minus input for amplifying bus current signal
P3.1 / AMP0P	25	DB / AI	GPIO P3.1 AMP0 positive input for amplifying bus current signal
P3.2 / AD5 / VHALF	26	DB / AI / AO	GPIO P3.2 Overtemperature signal input, ADC channel 5 input 1/2 VDD5 or 1/2 VREF reference output, an external 1uF capacitance is required
P3.5 / VREF	27	DB / AI	GPIO P3.5 ADC external reference voltage input or internal VREF output, an external 1~4.7uF capacitance is required
VSS	28	P	GND
IOVCC	29	P	GPIO power input, 3~ 5.5v, which should connect with a 1~10uF capacitance to the ground, IOVCC is less than or equal to VDD5

Name	FU6861 QFN40	Types	Function
P0.0 / SDA	30	DB / DB	GPIO P0.0 I2C SDA, collector open circuit output, can be configured with 5k pull-up resistance
P0.1 / TIM4 / SCL	31	DB / DB / DB	GPIO P0.1 Timer4 captures mode input or PWM mode output I2C SCL clock, collector open circuit output, can be configured with 5k pull-up resistance
P0.4 / NSS	32	DB / DB	GPIO P0.4 SPI's port of choice
P0.5 / TXD / MOSI	33	DB / DO / DB	GPIO P0.5 UART1 data transmitter before function transfer SPI_MOSI, master mode output or slave mode input
P0.6 / RXD / SCLK	34	DB / DI / DB	GPIO P0.6 UART1 data receiver before function transfer SPI interface clock CLK
P0.7 / MISO / CMPXO / TIM2S/ P1.1 / TIM3	35	DB / DB / DO / DB/ DB / DB	GPIO P0.7 SPI_MISO, master mode input or slave mode output The test pin of the comparator outputs Timer2 captures mode input or PWM mode output after function transferr GPIO P1.1, configurable for external interrupt 1 input Timer3 captures mode input
VDRV	36	P	6N Predriver power supply, 7~18V, an external 1~10uF capacitance is required
VSS	37	P	Digital GND
LU	38	DO	6N Predriver U phase lower side PWM output
LV	39	DO	6N Predriver V phase lower side PWM output
LW	40	DO	6N Predriver W phase lower side PWM output

Notes:

IO type description:

DI = Digital input,

DO = Digital output,

DB = Digital bidirectional

AI = Analog input,

AO = Analog output,

P = Power supply

2.10 FU6861N package-QFN40

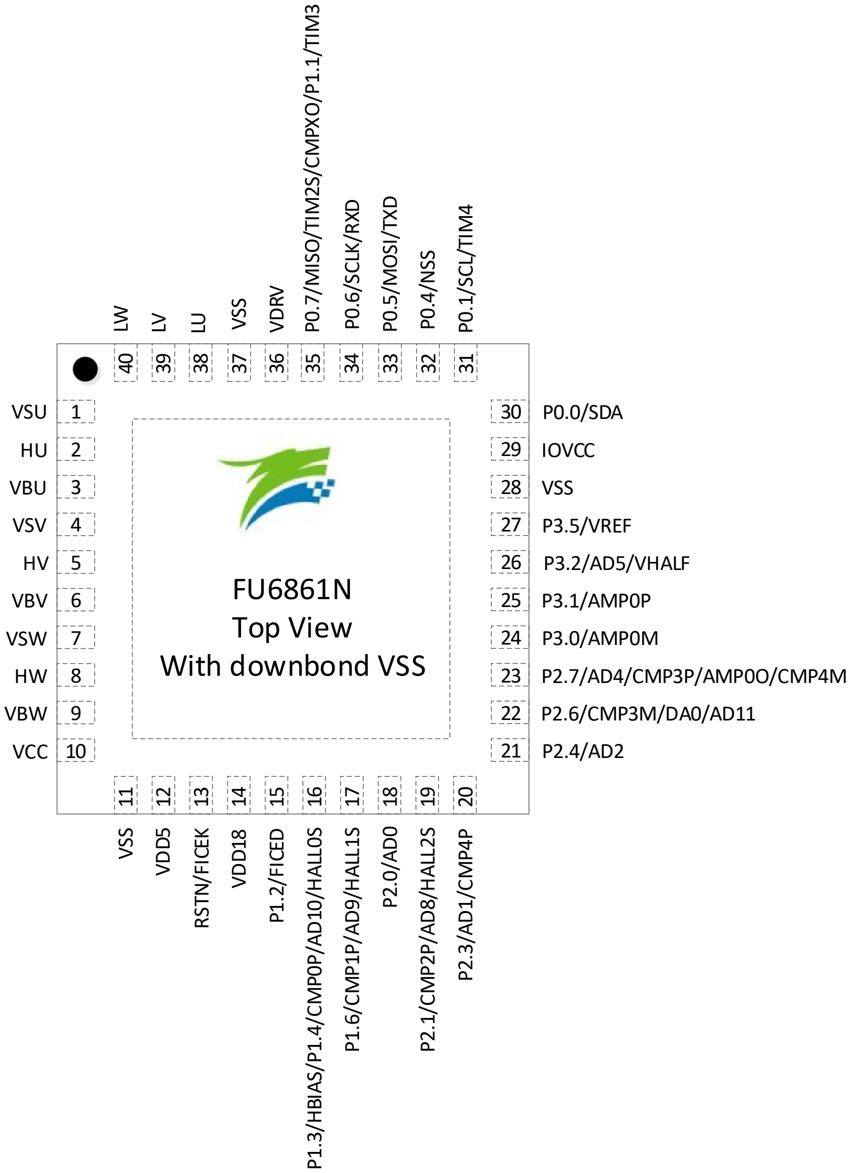


图 2-1 FU6861 QFN40 package

2.11 FU6861L LQFP48 pin definition

Table 2-5 FU6812L Pin Definition

Name	FU6861 LQFP48	Types	Function
P2.3 / AD1 / AMP2O / CMP4P	1	DB / AI / AO / AI	GPIO P2.3 ADC channel 1 input is used to sample the amplified phase current 2 signal AMP2 output CMP4 positive input
P2.4 / AD2	2	DB / AI	GPIO P2.4 ADC channel 2 input, bus voltage signal input
P2.6 / CMP3M / DA0 / AD11	3	DB / AI / AO / AI	GPIO P2.6 Overcurrent reference signal input, CMP3 minus input Internal DAC voltage outputs without buffer ADC channel 11 input
P2.7 / AD4 / CMP3P / AMP0O / CMP4M	4	DB / AI / AI / AO / AI	GPIO P2.7 ADC channel 4 input is used to sample amplified bus current signal The positive input of CMP3 is connected with the bus current sampling signal for detecting over current AMP0 output, outputs the voltage after amplifing the bus current CMP4 minus input
P3.0 / AMP0M	5	DB / AI	GPIO P3.0 AMP0 minus input for amplifying bus current signal
P3.1 / AMP0P	6	DB / AI	GPIO P3.1 AMP0 positive input for amplifying bus current signal
P3.2 / AD5 / VHALF	7	DB / AI / AO	GPIO P3.2 Overtemperature signal input, ADC channel 5 input 1/2 VDD5 or 1/2 VREF reference output, an external 1uF capacitance is required
P3.3 / AD6 / RXDS	8	DB / AI / DI	GPIO P3.3 ADC channel 6 input UART data receiver after function transfer
P3.4 / AD7 / TXDS	9	DB AI / DO	GPIO P3.4 Analog speed control input, AD channel 7 input UART data transmitter after function transfer
P3.5 / VREF	10	DB / AI	GPIO P3.5 ADC external reference voltage input or internal VREF output, an external 1~4.7uF capacitance is required

Name	FU6861 LQFP48	Types	Function
IOVCC	11	P	GPIO power input, 3~ 5.5v, and connected to an 1~10uF capacitance to the ground, IOVCC is less than or equal to VDD5
P0.0 / SDA	12	DB / DB	GPIO P0.0 I2C SDA, collector open circuit output, can be configured with 5k pull-up resistance
P0.1 / TIM4 / SCL	13	DB / DB / DB	GPIO P0.1 Timer4 captures mode input I2C SCL clock, collector open circuit output, can be configured with 5k pull-up resistance
P0.4 / NSS	14	DB / DB	GPIO P0.4 SPI's port of choice
P0.5 / TXD / MOSI	15	DB / DO / DB	GPIO P0.5 UART1 data transmitter before function transfer SPI_MOSI, master mode output or slave mode input
P0.6 / RXD / SCLK	16	DB / DI / DB	GPIO P0.6 UART1 data receiver before function transfer SPI interface clock CLK
P0.7 / MISO / CMPXO / TIM2S	17	DB / DB / DO / DB	GPIO P0.7 SPI_MISO, master mode input or slave mode output The test pin of comparator outputs Timer2 captures mode input or PWM mode output after function transfer
P1.0 / TIM2	18	DB/ DB	GPIO P1.0 Timer2 captures mode input or PWM mode output before function transfer
P1.1 / TIM3	19	DB/ DB	GPIO P1.1 Timer3 captures mode input
VDRV	20	DB/ DO	6N Predriver power supply, 7~18V, an external 1~10uF capacitance is required
VSS	21	P	Digital GND
LU	22	DO	6N Predriver PWM output at the lower side of phase U
LV	23	DO	6N Predriver PWM output at the lower side of phase V
LW	24	DO	6N Predriver PWM output at the lower side of phase W
VSU	25	P	6N Predriver PWM input of phase U, used for U phase upper side bootstrap ground reference
HU	26	DO	6N Predriver PWM output on the upper side of phase U
VBU	27	P	6N Predriver U phase upper side bootstrap power supply
VSV	28	P	6N Predriver PWM input of phase V, used for V phase upper side bootstrap ground reference

Name	FU6861 LQFP48	Types	Function
HV	29	DO	6N Predriver PWM output on the upper side of phase V
VBV	30	P	6N Predriver V phase upper side bootstrap power supply
VSW	31	P	6N Predriver PWM input of phase W, used for W phase upper side bootstrap ground reference
HW	32	DO	6N Predriver PWM output on the upper side of phase W
VBW	33	P	6N Predriver W phase upper side bootstrap power supply
NC	34		NC Pin, dangling
VCC	35	P	Power input, the voltage range is determined by the power mode VCC_MODE, with an external 10uF or larger filter capacitance.
VSS	36	P	Digital GND
VDD5	37	P	Medium voltage power supply input or internal 5V LDO output power supply, determined by VCC_MODE, power connection method please refer to the description of VCC pin, an external 1~4.7uF capacitance is required.
RSTN / FICEK	38	DI/ DI	External reset input, built-in pull up resistance, schmidt input FICE debugging interface clock port
VDD18	39	P	1.8V LDO output power, an external 1~4.7uF capacitor is required
P1.2 / FICED	40	DB/ DB	GPIO P1.2 FICE data port
P1.3 / HBIAS / CMP1PS	41	DB/ DO/ AI	GPIO P1.3 HALL bias power supply, connects to VDD5 by a switch internally CMP1 second mode positive input
P1.4 / CMP0P / AD10 / HALL0S	42	DB/ AI/ AI/ DI	GPIO P1.4 BEMF0 positive input ADC channel 10 input HALL0 logic level input after function transfer
P1.5 / CMP0M	43	DB / AI	GPIO P1.5 BEMF0 minus input
P1.6 / CMP1P / AMP1P / AD9 / HALL1S	44	DB / AI / AI / AI / DI	GPIO P1.6 BEMF1 positive input AMP1 positive input, connected with phase current 1 voltage signal input ADC channel 9 input HALL1 logic level input after function transfer
P1.7 / CMP1M / AMP1M	45	DB / AI / AI	GPIO P1.7 BEMF1 minus input AMP1 minus input

Name	FU6861 LQFP48	Types	Function
P2.0 / AD0 / AMP1O	46	DB / AI / AO	GPIO P2.0 ADC channel 0 input, used to samle the amplified phase current 1 signal AMP1 output
P2.1 / CMP2P / AMP2P / AD8 / HALL2S	47	DB / AI / AI / AI / DI	GPIO P2.1 BEMF2 positive input AMP2 positive input, connects to phase current 2 voltage signal input ADC channel 8 input HALL2 logic level input after function transfer
P2.2/ CMP2M/ AMP2M	48	DB/ AI/ AI	GPIO P2.2 BEMF2 minus input AMP2 minus input

2.12 FU6861L package-LQFP48

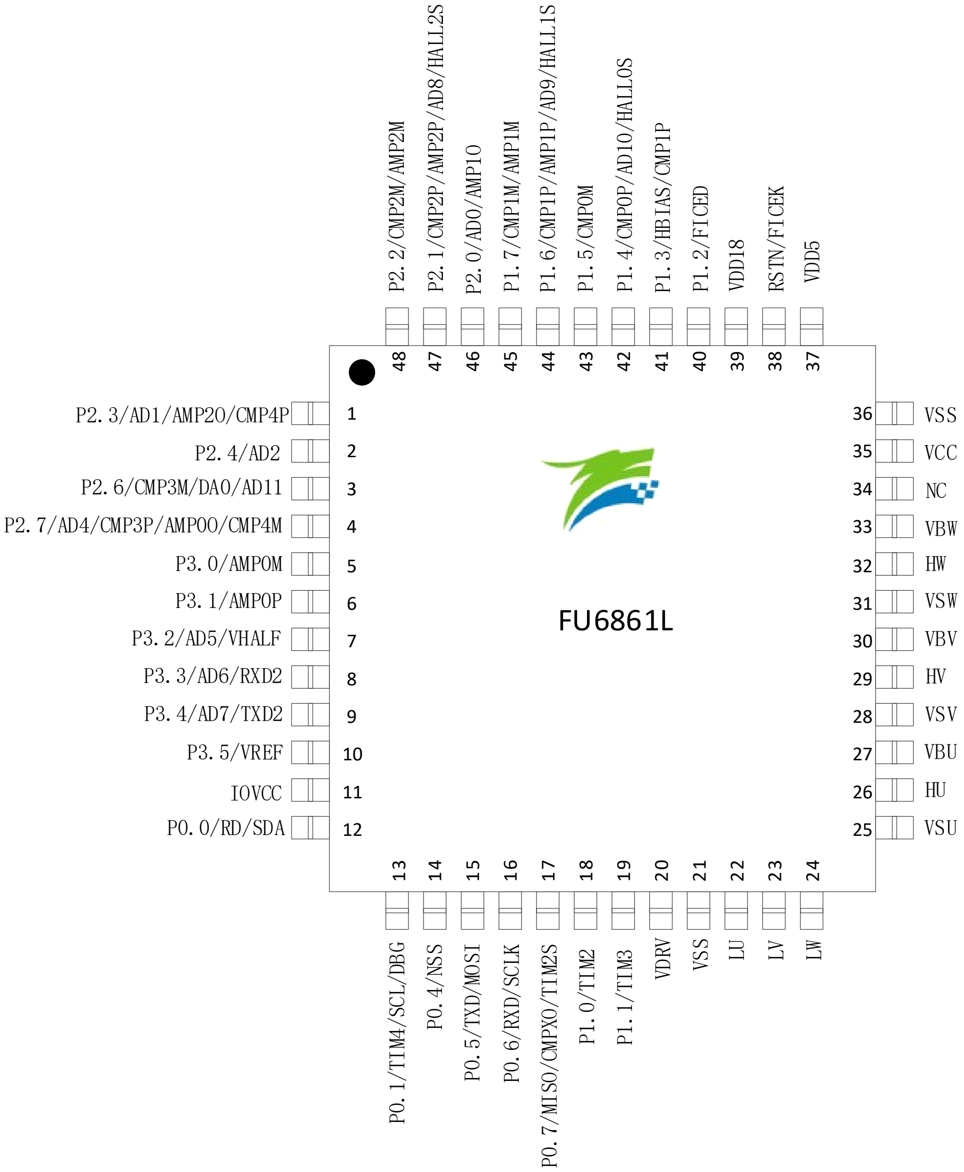


Figure 2-6 FU6861L LQFP48 package

2.13 FU6812P LQFP32 pin definition

Table 2-6 FU6812N Pin Definition

Name	FU6812 LQFP32	Types	Function
P2.1 / CMP2P / AMP2P AD8 / HALL2S	1	DB / AI / AI / AI DI	GPIO P2.1 BEMF2 positive input AMP2 positive input, connects to phase 2 voltage signal input ADC channel 8 input HALL2 logic level input after function transfer
P2.2/ CMP2M/ AMP2M	2	DB/ AI/ AI	GPIO P2.2 BEMF2 minus input AMP2 minus input
P2.3/ AD1/ AMP2O/ CMP4P	3	DB/ AI/ AO/ AI	GPIO P2.3 ADC channel 1 input, used to collect amplified phase current 2 signal AMP2 output CMP4 positive input
P2.4 / AD2	4	DB / AI	GPIO P2.4 ADC channel 2 input, bus voltage signal input
P2.7 / AD4 / CMP3P / AMP0O/ CMP4M	5	DB / AI / AI / AO/ AI	GPIO P2.7 ADC channel 4 input is used to sample amplified bus current signal CMP3 positive input , connect with the bus current sampling signal for detecting over current AMP0 output, output the voltage after amplifing the bus current CMP4 minus input
P3.0 / AMP0M	6	DB / AI	GPIO P3.0 AMP0 minus input for amplifying bus current signal
P3.1 / AMP0P	7	DB / AI	GPIO P3.1 AMP0 positive input for amplifying bus current signal
P3.2 / AD5 / VHALF	8	DB / AI / AO	GPIO P3.2 Overtemperature signal input,ADC channel 5 input 1/2 VDD5 or 1/2 VREF reference output, an external 1uF capacitance is required
P3.4/ AD7/ TXDS	9	DB / AI/ DO	GPIO P3.4 Analog speed control input, AD channel 7 input UART data transmitter after function transfer
IOVCC	10	P	GPIO power input, 3~ 5.5V, and connected with an 1~10uF capacitance to the ground, IOVCC is less than or equal to VDD5

Name	FU6812 LQFP32	Types	Function
P0.1 / TIM4 / SCL	11	DB / DB / DB	GPIO P0.1 Timer4 captures mode input I2C SCL clock, collector open circuit output, can be configured with 5k pull-up resistance
P0.5 / TXD / MOSI	12	DB / DO / DB	GPIO P0.5 UART1 data transmitter before function transfer SPI_MOSI, master mode output or slave mode input
P0.6 / RXD / SCLK	13	DB / DI / DB	GPIO P0.6 UART1 data receiver before function transfer SPI interface clock CLK
P0.7 / MISO / CMPXO / TIM2S/ P1.1 / TIM3	14	DB / DB / DO / DB/ DB / DB	GPIO P0.7 SPI_MISO, master mode input or slave mode output The test pin of the CMP outputs Timer2 captures mode input or PWM mode output after function transferr GPIO P1.1 Timer3 captures mode input
L_DU	15	DO	PWM output at the lower side of phase U
L_DV	16	DO	PWM output at the lower side of phase V
L_DW	17	DO	PWM output at the lower side of phase W
H_DU	18	DO	PWM output at the upper side of phase U
H_DV	19	DO	PWM output at the upper side of phase V
H_DW	20	DO	PWM output at the upper side of phase W
VCC	21	P	Power input, the voltage range is determined by the power mode VCC_MODE, an external 10uF or greater filter capacitance is required.
VSS	22	P	Digital GND
VDD5	23	P	Medium voltage power supply input or internal 5V LDO output power, determinded by VCC_MODE, power connection method please refer to the description of VCC pin, an external a 1~4.7uF capacitance is required.
RSTN / FICEK	24	DI / DI	External reset input, built-in pull up resistance, schmidt input FICE debugging interface clock port
VDD18	25	P	1.8V LDO output power, an external 1~4.7uF capacitor is required
P1.2 / FICED	26	DB / DB	GPIO P1.2 FICE data port
P1.3 / HBIAS / CMP1PS	27	DB / DO / AI	GPIO P1.3 HALL bias power supply, connects to VDD5 by a switch CMP1 second mode positive input

Name	FU6812 LQFP32	Types	Function
P1.4 / CMP0P / AD10 / HALL0S	28	DB / AI / AI / DI	GPIO P1.4 BEMF0 positive input ADC channel 10 input HALL0 logic level input after function transfer
P1.5 / CMP0M	29	DB / AI	GPIO P1.5 BEMF0 minus input
P1.6 / CMP1P / AMP1P/ AD9 / HALL1S	30	DB / AI / AI / AI / DI	GPIO P1.6 BEMF1 positive input AMP1 positive input, connect to phase current 1 voltage signal input ADC channel 9 input HALL1 logic level input after function transfer
P1.7/ CMP1M/ AMP1M	31	DB/ AI/ AI	GPIO P1.7 BEMF1 minus input AMP1 minus input
P2.0/ AD0/ AMP10	32	DB/ AI/ AO	GPIO P2.0 ADC channel 0 input, used to sample amplifyed phase current 1 signal AMP1 output

2.14 FU6812P package-LQFP32

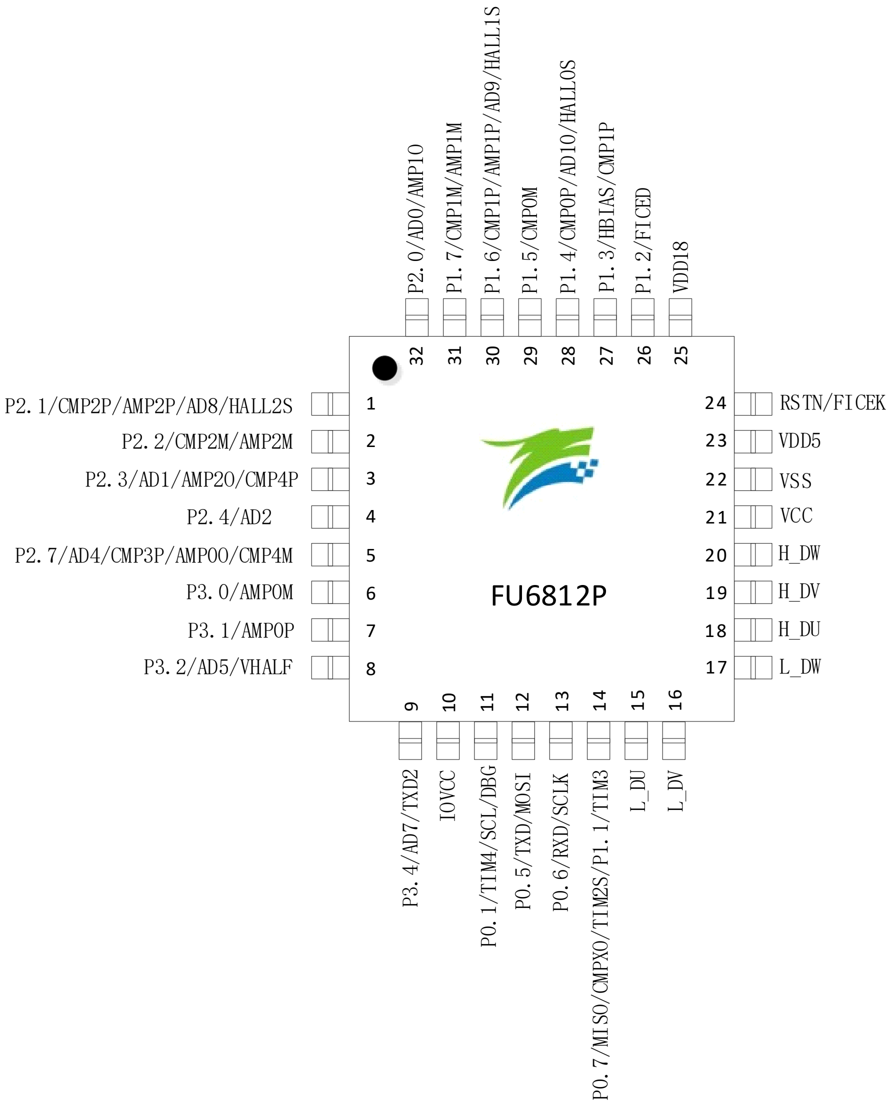


Figure 2-7 FU6812P LQFP32 package

3 Electrical characteristics

3.1 Absolute maximum ratings

Table 3-1 Absolute Maximum Ratings

Parameter	Conditions	Min	Type	Max	Unit
Ambient temperature at work T_A		- 40	-	85	℃
Ambient temperature at work T_A	VCC less than or equal to 12V, Ivcc less than or equal to 30mA	- 40	-	105	℃
Junction temperature at work is T_J		- 40	-	150	℃
Storage temperature		- 65	-	150	℃
VCC versus VSS		-0.3	-	36	V
VDD5/IOVCC versus VSS		-0.3	5	6.5	V
VDRV versus VSS	Only for FU6861	-0.3	-	22	V
VBU, VBV, VBW floating voltage	Only for FU6861	-0.3	-	180	V
VSU, VSV, VSW	Only for FU6861	VBU -22, VBV -22, VBW -22	-	VBU +0.3 VBV +0.3 VBW +0.3	V
HU, HV, HW	Only for FU6861	VSU -0.3 VSV -0.3 VSW -0.3	-	VBU +0.3 VBV +0.3 VBW +0.3	V
VDD18		-0.3	1.85	2	V
RSTN, VCC_MODE, GPIO versus VSS		-0.3	-	VDD5 + 0.3	V

Notes:

If operating conditions exceed the above "Absolute Maximum Ratings", it may cause permanent damage to the device. The above values are only maxima of operating conditions, and we do not recommend that devices run outside the scope of this specification. The stability of the device may be affected under the condition of absolute limit parameters.

3.2 Electrical characteristic

Table 3-2 Global electrical characteristics (for FU6812)

(unless otherwise stated, T_A = 25℃, VCC=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
VCC operating voltage	Single power supply high voltage mode, VCC_MODE=0	5	-	24	V
	Dual power supply high voltage mode, VCC_MODE=1, VCC greater than or equal to VDD5 (2)	5	-	36	V
	Single power supply low voltage mode, VCC_MODE=1, VCC connected with VDD5 (2)	3	-	5.5	V

Parameter	Conditions	Min	Type	Max	Unit
VDD5 working voltage	VCC_MODE=1, VCC is connected to VDD5 (2)	3	-	5.5	V
IOVCC working voltage		3	VDD5	VDD5+0.3	V
The system clock		-	24	-	MHz
I_{VCC} Working current	(1)	-	24	-	mA
I_{VCC} Standby current	(1)	-	6	-	mA
I_{VCC} Sleep current	VCC_MODE = 0	-	100	250	uA
	VCC_MODE = 1, VCC = VDD5 = 5V	-	45	100	uA

Notes:

1. Changes depending on how the program is running
2. VDD5 must be kept at 5~ 5.5V when Flash write or erase.
3. VCC_MODE=0, that is, VCC_MODE=GND; VCC_MODE=1, that is, VCC_MODE=VDD5, and the voltage of VCC_MODE is the same as this unless otherwise stated

Table 3-3 Global electrical characteristics (for FU6861)

(unless otherwise stated, T_A = 25℃, VCC=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
VCC working voltage	Single power supply high voltage mode, VCC_MODE=0	5	-	24	V
	Dual power supply high voltage mode, VCC_MODE=1, VCC greater than or equal to VDD5, (2)	5	-	36	V
VDD5 working voltage	VCC_MODE=1, VCC is connected to VDD5, (2)	3	-	5.5	V
IOVCC working voltage		3	VDD5	VDD5+0.3	V
VDRV operating voltage		7	-	18	V
VBU,VBV,VBW floating voltage		-	-	100	V
VBU versus VSU VBV versus VSV, VBW versus VSW		-	-	18	V
The system clock		-	24	-	MHz
i.$_{VCC}$ Working current	(1)	-	24	-	mA
i.$_{VCC}$ Standby current	(1)	-	6	-	mA
i.$_{VCC}$ Sleep current	VCC_MODE = 0	-	350	650	uA
	VCC_MODE=1, VCC=VDD5=5V	-	300	500	uA

Notes:

1. Changes depending on how the program is running
2. VDD5 must be kept at 5~ 5.5V for Flash write or erase operation

3.3 GPIO electrical characteristics

Table 3-4 GPIO electrical characteristics

(unless otherwise stated, T_A =25℃, VCC=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
Output rise time	50pF Load, up from 10% to 90% time, T_A= 25℃	-	15	-	ns
Output drop time	50pF Load, down from 90% to 10% time, T_A= 25℃	-	13	-	ns
Output high voltage V_{OH}	I_{OH}= 4mA, IOVCC=VDD5=5V	VDD5-0.7	-	-	V
Output low voltage V_{OL}	I_{OL}= 8mA, IOVCC=VDD5=5V	-	-	0.7	V
Input high voltage V_{IH}	(1)	0.7*VDD5	-	-	V
Input low voltage V_{IL}	IOVCC = VDD5 = 5 V	-	-	0.2*VDD5	V
Pull-up resistance of GPIO except P0[2:0], P1[6:3], P2[1], P3[7:6]	Vin = 0V	-	33	-	kΩ
Pull-up resistance of GPIO except P0[2:0], P1[6:3], P2[1], P3[7:6]	Vin = 0V	-	5	-	kΩ

(1) When VDD5=5V, the minimum value of V_{IH} can be 0.6*VDD5

3.4 PWM IO electrical characteristics (for FU6812)

Table 3-5 PWM IO electrical characteristics

(unless otherwise stated, T_A =25℃, VCC=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
Output pull current	HDIO = 1	-	50	-	mA
Output perfusion current	HDIO = 1	-	100	-	mA
Output rise time	50pF Load, from 10% to 90% time, T_A= 25℃	-	7	-	ns
Output drop time	50pF Load, down from 90% to 10% time, T_A= 25℃	-	5	-	ns

3.5 Predriver 6N IO electrical characteristics (for FU6861)

Table 3-6 Predriver 6N IO electrical characteristics (for FU6861)

(unless otherwise stated, T_A = 25℃, VCC=VDRV=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
High level output peak current		-	0.8	-	A
Low level output peak current		-	0.8	-	A
Output rise time	1nF Load, up from 10% to 90% of the time	-	15	30	ns
Output drop time	1nF Load, down from 90% to 10% time	-	15	30	ns

3.6 ADC electrical characteristics

Table 3-7 Electrical characteristics of ADC

(unless otherwise stated, T_A =25℃, VCC=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
INL		-	2	-	LSB
DNL		-	1.5	-	LSB
OFFSET		-	10	-	LSB
SNR	f_{IN} = 350 kHz, (1) (2)	-	70.8	-	dB
ENOB	f_{IN} = 350 kHz, (1) (2)	-	10.5	-	Bit
SFDR	f_{IN} = 350 kHz, (1) (2)	-	68.2	-	dB
THD	f_{IN} = 350 kHz, (1) (2)	-	67	-	dB
Input resistance R_{IN}		-	500	-	Ω
Input capacitance C_{IN}		-	30	-	pF
Conversion time		-	0.6	-	us
Sampling time	(1)	3	-	63	Number Of ADCLK

Notes:

(1) ADCLK = 12MHz

(2) Based on simulation results

3.7 Reference voltage electrical characteristics

Table 3-8 VREF & VHALF

(T_A =-40℃~85℃,VCC=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
VREF	VREFVSEL = 00B	-	4.5	-	V
	VREFVSEL = 01B	-	VDD5	-	V
	VREFVSEL = 11B	-	4	-	V
	VREFVSEL = 10B	-	3	-	V
VHALF		-	VREF/2,	-	V

3.8 Operational amplifier electrical characteristics

Table 3-9 Electrical characteristics of operational amplifiers

(unless otherwise stated, T_A =25℃, VCC=15V,VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
Common mode input range V_{ICMR}		0	-	VDD5-1.5	V
Mismatch voltage V_{OS} of operational amplifier		-	5	-	mV
The open loop gain A_{OL}	R_L= 100 k Ω	-	80	-	dB
unit gain bandwidth UGBW	C_L= 40 pF	6	10	-	MHz
Swing rate of SR operational amplifier	C_L= 40 pF	10	15	-	V/us

3.9 HALL/BEMF electrical features

Table 3-10 HALL/BEMF electrical characteristics

(unless otherwise stated, T_A =25℃,VCC=15V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
Internal BEMF resistance		5.4	6.8	8.2	kΩ
Relative precision between BEMF built-in resistors		-	1	-	%

3.10 Electrical characteristics of OSC

Table 3-11 Electrical characteristics of OSC

(T_A = - 40℃~85℃,VCC=5~24V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
Internal fast clock frequency		23.5	24	24.5	MHz
WDT clock frequency		29	32.8	37	kHz

3.11 Reset electrical characteristics

Table 3-12 Reset electrical characteristics

(T_A = - 40℃~85℃, VCC = 5~24V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
Reset the minimum width of the low level		-	25	50	us

3.12 LDO electrical characteristics

Table 3-13 LDO electrical characteristics

(T_A = -40℃~85℃, VCC=5v~24V, VCC_MODE=0)

Parameter	Conditions	Min	Type	Max	Unit
VDD5 voltage	VCC = 7~ 30V, VCC_MODE = 0	4.7	5	5.3	V
VDD18 voltage		-	1.85	-	V

3.13 Package thermal resistance

Table 3-14 LQFP48 package thermal resistance

Parameter	Conditions	Value	Unit
Chip junction temperature Θ_{JA} relative to ambient temperature	(1), (3)	52.4	℃/W
	(2), (3)	72.2	℃/W
Chip junction temperature Θ_{JC} relative to package surface temperature	(2), (3)	17	℃/W

(1) JEDEC standard, 2S2P PCB

(2) JEDEC standard, 1S0P PCB

(3) The actual application condition is different, it will differ with the test result

Table 3-15 QFN56 package thermal resistance

Parameter	Conditions	Value	Unit
Chip junction temperature Θ_{JA} relative to ambient temperature	(1), (3)	33	℃/W
	(2), (3)	55	℃/W
Chip junction temperature Θ_{JC} relative to package surface temperature	(1), (3)	9.2	℃/W

(1) JEDEC standard, 2S2P PCB

(2) JEDEC standard, 1S0P PCB

(3) The actual application condition is different, it will differ with the test result

Table 3-16 QFN40 package thermal resistance

Parameter	Conditions	Value	Unit
Chip junction temperature Θ_{JA} relative to ambient temperature	(1), (3)	40	℃/W
	(2), (3)	66	℃/W
Chip junction temperature Θ_{JC} relative to package surface temperature	(1), (3)	12	℃/W

(1) JEDEC standard, 2S2P PCB

(2) JEDEC standard, 1S0P PCB

(3) The actual application condition is different, it will differ with the test result

Table 3-17 QFN32 package thermal resistance

Parameter	Conditions	Value	Unit
Chip junction temperature Θ_{JA} relative to ambient temperature	(1), (3)	47	℃/W
	(2), (3)	74	℃/W
Chip junction temperature Θ_{JC} relative to package surface temperature	(1), (3)	20	℃/W

(1) JEDEC standard, 2S2P PCB

(2) JEDEC standard, 1S0P PCB

(3) The actual application condition is different, it will differ with the test result

Table 3-18 SSOP24 package thermal resistance

Parameter	Conditions	Value	Unit
Chip junction temperature Θ_{JA} relative to ambient temperature	(1), (2)	75	℃/W

(1) JEDEC standard, 2S2P PCB

(2) The actual application condition is different, it will differ with the test result

Table 3-19 LQFP32 Package thermal resistance

Parameter	Conditions	Value	Unit
Chip junction temperature Θ_{JA} relative to ambient temperature	(1), (3)	55	℃/W
	(2), (3)	75	℃/W
Chip junction temperature Θ_{JC} relative to package surface temperature	(1), (3)	20	℃/W

(1) JEDEC standard, 2S2P PCB

(2) JEDEC standard, 1S0P PCB

(3) The actual application condition is different, it will differ with the test result

4 Reset control

4.1 Reset source (RST_SRC)

The chip has 6 reset sources:

- Power-on reset (RSTPOW)
- External reset (RSTEXT)
- Low-voltage reset (RSTLVD)
- Watchdog reset (RSTWDT)
- Flash operation error reset (RSTFED)
- Debug reset (RSTDBG).

The reset flag can be queried and recorded in register RST_SR. The last reset will put the relevant position 1, the other signs clear 0. If the flag bit needs to be cleared, RST_SR[RSTCLR] can be cleared by setting 1.

4.2 Reset enable

Reset to enable reference to the associated configuration registers. LVDENB and WDTRSTEN control bits can respectively enable the reset source of LVD and WDT.

4.3 Power-on reset/external reset

When the chip RSTN pin is lower than 25us, the chip considers this to be a reset event. After reset, the MCU starts to execute the program from address 0. Setting the reset signal to be valid, the MCU will start the reset and BOOT procedures. Similarly, during the process of power on, the chip will also turn on the power on position of the internal circuit, start and reset. .

4.4 Low voltage reset

The internal circuit of the chip will monitor the VDD. If the voltage of the VDD drops to the reset threshold, the internal monitoring circuit will send out the corresponding reset signal, prompting the chip to reset.

Relative registers are configured to enable low voltage detection circuits, as well as low voltage thresholds.

4.5 Watchdog reset

After enabling watchdog timer, if the dog is not fed in time before its count overflow, counter overflow will cause the system reset. This reset source prevents the program from running away. After the watchdog overflows, the reset module will reset the MCU.

4.6 FEDR reset

The FLASH action module provides the ability for the software to "write", "erase" and read the encrypted sector of FLASH with MOVX (see the code protection section below). If the software tries to use this command to manipulate the sector where the encryption bit is located and the encrypted sector, it will issue FLASH illegal operation reset.The FEDR reset source has been enabled and cannot be disabled.

4.7 Reset register

Table 4-1 Reset control RST_SR (0xC9)

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

Bit	Name	Function
[7]	RSTPOW	Power on reset flag 0: Last reset did not come from the power on reset 1: Last reset came from the power on reset
[6]	RSTEXT	External reset flag 0: Last reset did not come from an external reset 1: Last reset came from external reset
[5]	RSTLVD	Low voltage reset flag 0: Last reset was not caused by a low voltage reset 1: Last reset came from the reset caused by low voltage
[4]	RSV	RSV
[3]	RSTWDT	Watchdog overflow reset flag 0: Last reset was not caused by a watchdog overflow 1: Last reset came from the overflow of the watchdog
[2]	RSTFED	FLASH code protection flag 0: Last reset was not caused by FLASH code protection 1: Last reset came from the reset caused by FLASH code protection
[1]	RSTDBG	Debug reset flag 0: Last reset was not caused by the debug interface 1: Last reset came from the reset caused by the debug interface
[0]	RSTCLR	Clear the analog reset flag register Clear Bit[7:3] reset flag when writing 1 and meaningless when reading.

5 Interrupt control

5.1 Interrupt register

5.1.1 IE (0xA8)

Table 5-1 IE (0xA8) Interrupt Enable

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

Bit	Name	Function
[7]	EA	Enable All Interrupts. 0: Disable 1: Enable
[6:5]	RSV	Reserved
[4]	ES0	UART interrupt enable 0: Disable 1: Enable
[3]	SPIIE	SPI interrupt enable 0: Disable 1: Enable
[2]	EX1	External interrupt 1 enable 0: Disable 1: Enable
[1]	TSDIE	TSD (Temperature sensor detect) Temperature sensing interrupt 0: Disable 1: Enable
[0]	EX0	External interrupt 0 enable 0: Disable 1: Enable

5.1.2 IP0 (0xB8)

Table 5-2 IP0 (0xB8) Interrupt Priority 0

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

Bit	Name	Function
[7:6]	PDRV	Driver interrupts priority control
[5:4]	PX1	INT1 (external interrupt 1) priority control
[3:2]	PX0	INT0 (external interrupt 0) priority control
[1:0]	PLVW	LVW (low voltage alarm) interrupt priority control

Note:

The interrupt priority control value from 0 to 3 successively indicates that the priority is from the lowest to the highest, and there are 4 levels of priority control.

5.1.3 IP1 (0xC0)

Table 5-3 IP1 (0xC0) Interrupt Priority 1

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

Bit	Name	Function
[7:6]	PCMP	The comparator interrupts priority control
[5:4]	PADC	ADC interrupts priority control
[3:2]	PTIM1	Timer 1 interrupts priority control
[1:0]	PTIM2	Timer 2 interrupts priority control

Note:

The interrupt priority control value from 0 to 3 successively indicates that the priority is from the lowest to the highest, and there are 4 levels of priority control.

5.1.4 IP2 (0xC8)

Table 5-4 IP2 (0xC8) Interrupt Priority-2

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

Bit	Name	Function
[7:6]	PTSD	TSD temperature sensor detect interrupt priority control
[5:4]	PTIM4	Timer 4 and systick interrupt priority control
[3:2]	PTIM3	Timer 3 interrupt priority control
[1:0]	RSV	Reserved

Note:

The interrupt priority control value from 0 to 3 successively indicates that the priority is from the lowest to the highest, and there are 4 levels of priority control.

5.1.5 IP3 (0xD8)

Table 5-5 IP3 (0xD8) Interrupt Priority 3

Bit	7	6	5	4	3	2	1	0
Name	PDMA0		PSPI		PI2C		PUART	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:6]	PDMA0	DMA0 interrupt priority control
[5:4]	PSPI	SPI interrupt priority control
[3:2]	PI2C	I2C interrupt priority control
[1:0]	PUART	UART interrupt priority control

Note:

The interrupt priority control value from 0 to 3 successively indicates that the priority is from the lowest to the highest, and there are 4 levels of priority control.

5.1.6 TCON (0x88)

Table 5-6 TCON (0x88)

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

Bit	Name	Function
[7:6]	RSV	Reserved
[5]	TSDIF	TSD temperature sensor detects interrupt flag.This flag bit is often used in conjunction with the temperature protection state bit (TSDF), which reflects a state that has exceeded the set temperature. 0: The chip has no interruption beyond the set temperature 1: There is an interruption of the chip exceeding the set temperature.The software writes 0 to clear this bit to zero
[4:3]	IT1[1:0]	INT1 external interrupt 1 level trigger control 00: Rising edge triggers interrupt 01: Falling edge triggers interrupt 1x: Level change (up or down) triggers the interrupt
[2]	IF0	INT0 external interrupt 0 flag 0: INT0 is not interrupted 1: INT0 broke.The software writes 0 to clear this bit to zero
[1:0]	IT0[1:0]	INT0 external interrupt 0 level trigger control 00: Rising edge triggers interrupt 01: Falling edge triggers interrupt 1x: Level change (up or down) triggers the interrupt

5.2 Interrupt description

Table 5-7 Interrupt Description

Interrupt source	Default priority	Vector address	Flag	Need software clear	Enable bit	Priority control
Reset	Highest	0x0000	N/A	N/A	Always enabled	Highest
LVW (Low Voltage Warning Interrupt)	0	0x0003	LVSR[0]	Y	CCFG1[6]	IP0[1:0]
External Interrupt (INT0)	1	0x000B	TCON[2]	Y	IE[0]	IP0[3:2]
External Interrupt (INT1)	2	0x0013	P1IF[7:0]/ P2IF[7:0]	Y	IE[2]	IP0[5:4]
DRV Interrupt	3	0x001B	DRV_SR[5:4]	Y	DRV_SR[2:0]	IP0[7:6]
TIM2 Interrupt	4	0x0023	TIM2_CR1[7:5]	Y	TIM2_CR1[4:3] TIM2_CR0[3]	IP1[1:0]
TIM1 Interrupt	5	0x002B	TIM1_SR[4:0]	Y	TIM_IER[4:0]	IP1[3:2]
ADC Interrupt	6	0x0033	ADC_CR[0]	Y	ADC_CR[1]	IP1[5:4]
CMP Interrupt	7	0x003B	CMP_SR[7]/ CMP_SR[6:4]	Y	CMP_CR0[7:0]	IP1[7:6]
RSV	8					
TIM3 Interrupt	9	0x004B	TIM3_CR1[7:5]	Y	TIM3_CR1[4:3] TIM3_CR0[3]	IP2[3:2]
TIM4 Interrupt Systick Interrupt	10	0x0053	TIM4_CR1[7:5] DRV_SR[7]	Y	TIM4_CR1[4:3] TIM4_CR0[3] DRV_SR[6]	IP2[5:4]
TSD Interrupt	11	0x005B	TCON[5]	Y	IE[1]	IP2[7:6]
UART Interrupt	12	0x0063	UT_CR[1:0]	Y	IE[4]	IP3[1:0]
I2C Interrupt	13	0x006B	I2C_SR[0]	Y	I2C_CR[0]	IP3[3:2]
SPI Interrupt	14	0x0073	SPI_CR1[7]	Y	SPI_CR1[0]	IP3[5:4]
DMA Interrupt	15	0x007B	DMA_CR0[7] DMA_CR1[7]	Y	DMA0_CR0[2]	IP3[7:6]

There are 15 interrupt sources inside the chip, as shown in Table 5-7.Each interrupt source has four levels of priority, configured through registers IP0~IP3.High-priority interrupt requests can be responded to in low-priority interrupt service routines.If two interrupts are in the same level, the order of priority is shown in the above table, the smaller the label, the higher the priority; new interrupts cannot interrupt interrupt processing of the same priority.

IE[EA] is global interrupts enabling, EA=0 disable all interrupt requests.

5.3 External interrupt

External interrupts have two interrupt sources.

When PORT0.0~PORT0.6 is set as digital IO input or comparator CMP4 is enabled, EX0=1 can be set as external interrupt 0 (INT0).When setting PORT1.0~1.7 and PORT2.0~2.7 as digital IO input, EX1=1 and corresponding P1IE/P2IE can be set to share external interrupt 1 (INT1).

There are external interrupt 0 enable bit EX0, interrupt flag bit IF0, interrupt level trigger control IT0.EXT0CFG in register LVSR specifies the source of external interrupt 0. These sources can be PORT0.0~PORT0.6 input or any of the comparator CMP4 output. All interrupt sources of external interrupt 0 share an interrupt entry and an interrupt flag bit.

External interrupt 1 enable bit EX1, 16 PIN interrupt enable depending on P1IE, P2IE.The corresponding interrupt flag bit is P1IF and P2IF, and the interrupt level trigger control is IT1.

Table 5-8 IO corresponding to external interrupt 1

SFR address	Field	Name	Describe	R/W	Reset value
0xD1	[7:0]	P1IE[7:0]	Interrupt enable bits of corresponding pin of Port1 when Port1 as EXTI1 source.	R/W	0x00
0xD2	[7:0]	P1IF[7:0]	Interrupt flag bits of corresponding pin of Port1 when Port1 as EXTI1 source. NOTE: When the MCU writes 0 to clear the corresponding flag bit, the flag bit that does not need to be cleared must be written with 1, otherwise the interrupt may be cleared by mistake. The following statements are recommended: mov D2h,#0FEh, for clearing P1IF[0]	R/W	0x00
0xD3	[7:0]	P2IE[7:0]	Interrupt enable bits of corresponding pin of Port2 when Port2 as EXTI1 source.	R/W	0x00
0xD4	[7:0]	P2IF[7:0]	Interrupt flag bits of corresponding pin of Port2 when Port2 as EXTI1 source. NOTE: When the MCU writes 0 to clear the corresponding flag bit, the flag bit that does not need to be cleared must be written with 1, otherwise the interrupt may be cleared by mistake. The following statements are recommended: mov D4h,#0FEh, for clearing P2IF[0]	R/W	0x00

6 I2C

The I2C (Inter-Integrated Circuit Bus) module provides a two-wire serial interface that complies with industry standards. It is a simple two-way synchronous serial bus that can be used for communication between MCU and external I2C devices. The bus consists of two serial lines: SDA (serial data line) and SCL (serial clock line). These lines are bidirectional I/O lines, so the bus interface is open-drain output. The bus can work normally by pulling up the resistor to VDD5.

Main features:

- Implemented the standard mode of I2C protocol (up to 100 kHz), fast mode (up to 400 kHz) and fast+ mode (up to 1MHz).
- Supports both master mode and slave mode
- Support 7-bit address mode and broadcast addressing.
- Supports DMA data transfer, which can effectively reduce the burden on the CPU.

SDA and SCL are high when the bus is idle. This is the only basis for the device to detect whether the bus is idle. During the transmission, there is only one master device and at least one slave device on the bus. In this case, if other devices want to initiate I2C communication, they must wait until the end of the current communication. The master device is used to start the bus to transmit data, and send clock signals to all devices through SCL, and send slave addresses and read and write modes through SDA. If there is a device on the bus that matches the address, the device will act as a slave. The relationship between master and slave devices and data transmission and reception on the bus is not constant. If the master wants to send data to the slave device, the master first addresses the slave device, then actively sends data to the slave device, and finally the master terminates the data transfer. The communication process is shown in Figure 6-1. If the master wants to receive data from the slave device, the master device first addresses the slave device, then the master receives the data sent from the device, and finally the master terminates the receiving process, the communication process is shown in Figure 6-2. Under these circumstances, the master is responsible for generating the timing clock and terminating the data transfer.

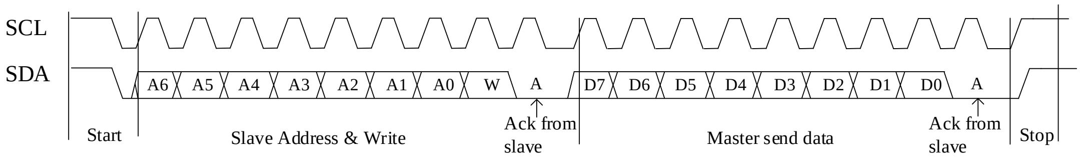


Figure 6-1 Master sends data to slave

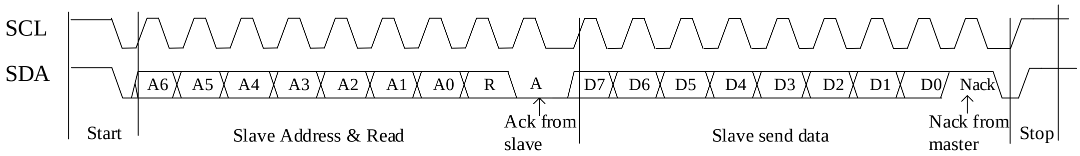


Figure 6-2 Master receives data from slave

The I2C of MCU series can be set to master mode or slave mode, the fastest speed up to 1MHz.When using I2C, it is only necessary to configure I2C and fill in the correct I2C address. Then, I2C communication is only controlled by the start signal (STA), read and write signal (DMOD), bus suspend signal (STR), reply signal (NACK) and stop signal (STP).

6.1 Operating instructions

6.1.1 Master mode

1. Configure I2C_CR[I2CMS] as master mode；
2. Configure I2C_CR[I2CSPD] to select the clock frequency SCL；
3. Configure I2C_ID to set target device address；
4. Configure I2C_SR[DMOD] to set the direction of the write/read；
5. Configure I2C_CR[I2CEN] to enable the I2C；
6. Configure I2C_SR[I2CSTA] to send START signal and address. After receiving ACK/NACK, I2C_SR[STR] is set by hardware and SCL is forced low by the master;
7. For sending data, after writing the I2C_DR register, reset I2C_SR[STR] to release SCL and the master starts sending data. When the data is sent and the ACK/NACK is received, I2C_SR[STR] is set to 1 by hardware and SCL is forced low by the master;
8. For receiving data, after resetting I2C_SR[STR], the master starts receiving data. When the data is received, I2C_SR[STR] is set by hardware, SCL is forced low by the master. At this time, you can set ACK/NACK through I2C_SR[NACK], then write 0 to I2C_SR[STR] to release SCL to send ACK/NACK signal. If new data is received, I2C_SR[STR] is set by hardware and SCL is forced low by the master.
9. If you want to stop sending, you can set I2C_SR[I2CSTP] when I2C_SR[STR] is 1, and master sends stop signal when I2C_SR[STR] is cleared.

6.1.2 Slave mode

1. Configure I2C_CR[I2CMS]=0 as slave mode；
2. Configure I2C_ID[I2CADD] to set the slave address; or configure I2C_ID[GC] = 1 to enable the broadcast mode.

3. Configure I2C_CR[I2CEN]=1 to enable I2C.

4. Waiting for the START signal and address. After receiving the START signal and the correct address, SCL is forced low by the slave. I2C_SR[I2CSTA] and I2C_SR[STR] are set by hardware. At this time, you can set ACK/NACK through I2C_SR[NACK], and select whether the communication is receiving or sending data through I2C_SR[DMOD].

5. For sending data, write the I2C_DAT register to send data. After resetting I2C_SR[STR] to release SCL, send data after sending ACK/NACK. When the data transmission is completed and the ACK/NACK from the master is received, the SCL is forced low by the slave, and I2C_SR[STR] is set to 1.

6. For receiving data, reset I2C_SR[STR] to release SCL when ready to receive data. When the slave finishes receiving data, the I2C_SR[STR] is set to 1 by hardware, SCL is forced low by the slave. After setting ACK/NACK through I2C_SR[NACK], reset I2C_SR[STR] to release SCL and send ACK/NACK. If new data is received, I2C_SR[STR] is set by hardware and SCL is forced low by the master.

7. RESTART: When the slave receives the START signal in the busy state, it will stop the current work and wait for the receiving address.

6.1.3 I2C interrupt resources

Interrupt sources of I2C are:

1．I2C_SR[STR] = 1, this interrupt source is valid in both master and slave modes.

2．I2C_SR[I2CSTP] = 1, This interrupt source is only valid in slave mode.

If the I2C interrupt I2C_CR [I2CIE] is set, the I2C will generate an interrupt request.

6.2 I2C register

6.2.1 I2C_CR (0x4028)

Table 6-1 I2C_CR (0x4028)

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

Bit	Name	Function
[7]	I2CEN	I2C Enable 0: Disable 1: Enable, switch the corresponding GPIO to I2C mode, OPEN DRAIN output.Whether the I2C pull-up is open or not is determined by the pull-up setting of its IO
[6]	I2CMS	I2C mode selection 0: Slave mode 1: Master mode
[5:3]	RSV	Reserved
[2:1]	I2CSPD	I2C rate setting, only valid for master mode 00:100kHz transmission rate 01:400kHz transmission rate 10:1MHz transmission rate 11: Reserved
[0]	I2CIE	Interrupt enable bit 0: I2C access interruption is prohibited 1: I2C is allowed to enter the interrupt. The interrupt request is generated by I2C_SR[I2CIF]

6.2.2 I2C_ID (0x4029)

Table 6-2 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	Function
[7:1]	I2CADD	I2C address
[0]	GC	Broadcast call support, valid only in slave mode. 0: Broadcast calls are not supported 1: Supports broadcast calls, namely 0x00 address will also respond

6.2.3 I2C_DR (0x402A)

Table 6-3 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	Function
[7:0]	I2C_DR	I2C data register

6.2.4 I2C_SR (0x402B)

Table 6-4 I2C_SR (0x402B)

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

Bit	Name	Function
[7]	I2CBSY	I2C busy status flag When I2CEN is 0, I2CBSY becomes 0 automatically. **Master mode:** After sending START successfully, the hardware sets '1'; after sending STOP successfully, the hardware clears '0'. **Slave mode:** After receiving START and address matching successfully, the hardware is set to '1'; after receiving STOP, the hardware is clear to '0'.
[6]	DMOD	I2C read or write flags **Master mode:** 0: Write mode (the Master side sends data and the slave side receives data) 1: Read mode (the Master side receives data and the slave side sends data) In Master mode, DMOD can only be effectively modified if: 1. I2CSTA bit to 1 2. When write 1 to I2CSTA , change DMOD **Slave mode:** 0: Write mode (the master side sends data and the slave side receives data) 1: Read mode (the master side receives data and the slave side sends data)
[5]	RSV	Reserved
[4]	I2CSTA	**Master mode:** The software is set to 1. When SCL and SDA are high, START and address bytes are sent, and 0 is automatically cleared by the hardware after sending. I2CSTA is prohibited from writing when sending or receiving data. To send RESTART, set I2CSTA to '1' after sending or receiving data. 0: Non-start and address bytes

<table>
<tr><td></td><td></td><td>1: Send START or RESTART and address bytes
Slave mode:
The hardware receives START and the address byte matches the postposition 1; the software writes 0 to clear; if the address does not match after receiving START, I2CSTA will not set. All subsequent events are ignored until the next START event is received.
In slave mode, I2CSTA and I2CSTP determine the current situation of I2C data:
Table 6-5 I2C status flags<table><tr><th>START</th><th>STOP</th><th>Desctiption</th></tr><tr><td>0</td><td>0</td><td>Data bytes are currently being sent/received</td></tr><tr><td>0</td><td>1</td><td>You currently receive STOP</td></tr><tr><td>1</td><td>0</td><td>The START + address byte is currently received</td></tr><tr><td>1</td><td>1</td><td>You currently receive STOP and then START + address bytes</td></tr></table>**Note: when I2CEN is' 0 ', I2CSTA will be automatically cleared by the hardware '0'.**</td></tr>
<tr><td>[3]</td><td>I2CSTP</td><td>**Master mode:**
When I2CBSY is 1, the software can effectively write 1, and the hardware will sends STOP when STR is 0, and the hardware will automatically clear after sending.If I2CSTA and I2CSTP write 1 at the same time, and I2CBSY will be '1', I2C first sends STOP, then sends START and address bytes after STOP, and STR hardware sets 1 after START and address bytes are sent. In the process of sending or receiving data, writing I2CSTP is prohibited until the data transmission is completed.
0: Do not send STOP
1: Send STOP
Slave mode:
I2CSTP is set by hardware after receiving STOP, software writes 0 to clear
Status flags refer to Table 6-5
Note: when I2CEN is 0, I2CSTP will be automatically cleared by hardware.</td></tr>
<tr><td>[2]</td><td>STR</td><td>I2C event completion indicates that when the bit is 1, the SCL will be forcibly pulled down to keep the bus busy.This bit is set by hardware 1, software write 0 to clear.
Master mode:
1. When hardware sends START+ address byte and receives ACK/NACK signal;
2. When the hardware has sent STOP, START + address bytes in order and received ACK/NACK signal;
3. When the hardware sends the data and receives an ACK/NACK signal;
4. When the hardware receives the data;
Slave mode:
1. When the hardware receives the START + matching address;</td></tr>
</table>

		2. When the hardware receives the data; 3. When the hardware sends an ACK/NACK signal and data; **Note:** when I2CEN is 0, I2CSTP will be automatically cleared by hardware.
[1]	NACK	The reply signal after I2C sends or receives the address byte or data byte, namely the 9th bit of data. In send mode, this bit is read only and can only read the response signal of the communication device. In the receiving mode, the bit can be read and written to send a reply signal to the communication device. The bit can only read the written value 0: ACK 1: NACK **Note:** when I2CEN is 0, I2CSTP will be automatically cleared by hardware.
[0]	I2CIF	I2C interrupt request flag bit, clearing I2CIF will allow I2C to continue transmitting data.This bit is controlled by the hardware. 0: No I2C interrupt request 1: There are I2C interrupt requests **Master mode:** When STR is 1, I2CIF set 1, otherwise set 0 **Slave mode:** When I2CSTP is 1 or STR is 1, I2CIF set 1, otherwise set 0 **Note:** when I2CEN is 0, I2CSTP will be automatically cleared by hardware.

7 SPI

SPI is the abbreviation of Serial Peripheral Interface, and it is a kind of high-speed full-duplex synchronous serial bus. SPI of FU6812 can be either a master or a slave, and can use a 3-wire or 4-wire transmission mode to allow multiple master devices and slave devices to exist on the bus.The complete SPI consists of four signal lines, namely MOSI, MISO, SCLK and NSS.

MOSI signal is the data signal of SPI. When SPI is the master, it outputs the data signal; SPI is the slave, it receives the data signal.

MISO signal is the data signal of SPI. When SPI is the master, it receives the data signal; when SPI is the slave, it outputs the data signal.The MISO pin is placed in a high impedance state when SPI is disabled or works in a 4-wire slave mode and is not selected.

SCLK signal is SPI clock signal, is the data signal transmission benchmark signal, sent by the master.

NSS signal is the selected communication number of SPI device. When SPI works in 3-wire mode, NSS signal will be disabled and NSS port will be just a normal IO port.When SPI works in slave mode, NSS ports can be configured as input ports to detect NSS signals from the master.When SPI works in single-master single-slave mode, NSS signal of the master can be configured as output to start SPI of the slave machine.When SPI works in multi-master mode, NSS signal is configured as input to detect whether other masters are communicating with the bus at present, so as to avoid conflicts when data transmission between more than two masters occurs.When SPI works in single-master multi-slave mode, the master can communicate by configuring multiple IO ports as NSS signals to select different slave machines.

7.1 Operation declaration

7.1.1 SPI master mode

When SPI mode select bit SPI_CR0[SPIMS] as 1, SPI will work in master mode.In this mode, SPI starts the transfer based on whether the shift register is empty.When writing data to SPI_DR, the data is actually written to the sending buffer. At this time, the null flag of sending buffer SPI_CR1[TXBMT] will be set to 0. If the shift register is empty at this time, the data in the sending buffer will be transferred to the shift register and the transmission starts. SCK outputs clock signal, MOSI and MISO will send or receive data from higher bit according SCK. After the transmission, SPI_CR1[SPIIF] and SPI_CR1[TXBMT] will set 1. The data of the shift register will be the data received by MISO, and the data will be sent to the receiving buffer. When the data is read from SPI_DR, the data of the receiving buffer will be obtained. In the case of SPI_CR1[TXBMT]=0, write data to SPI DR, then write clashes flag bit SPI_CR1[WCOL] will be set, and keep the data in the sending buffer.

7.1.1.1 Master mode configuration

1. Configure SPI_CR1[NSSMOD] and Set the working mode of SPI
2. Configure SPI_CR1[CPOL] and set the clock polarity.
3. Configure SPI_CR1[CPHA] and set the clock phase;
4. Configure SPI_CR1[SPIMS]=1, set the master mode;
5. Configure SPI_CLK and set SCK frequency;
6. Configure SPI_CR1[SPIEN]=1 to enable SPI;
7. Configure SPI_DR to write the data to be sent, and SPI will transfer it every time it writes.

7.1.2 SPI slave mode

When SPI_CR0[SPIMS] is 0, SPI will work in slave mode.In this mode, SPI's SCK signal will be provided by the SPI of the master.When SCK signal is not input, sending buffer flag bit will be the initial state. When SCK signal is input, MOSI and MISO of slave machine will start to receive and send data. When transmission is completed, SPI_CR1[SPIIF] and SPI_CR1[TXBMT] will be set to 1, and SPI_CR0[RXBMT], empty flag bit of receiving buffer, will be set to 0 to indicate that there is currently unread data. If SPI_CR0[RXBMT]=0 and new data is ready to be sent to the receiving buffer, SPI_CR1[RXOVR] will set 1 and the data in the receiving buffer will remain unchanged. SPI_CR1[TXBMT] is set to 0 when writing data to SPI_DR. If the data is then written to SPI_DR, SPI_CR1[WCOL] will set to 1 and keep the data in the sending buffer. If the 4-wire mode is used, the NSS signal will be the input signal and the descending edge of the NSS will reset the bit counter.

7.1.2.1 Slave configuration

1. Configure SPI_CR1[NSSMOD] to use 3-line slave mode and 4-line slave mode
2. Configure SPI_CR1[CPOL] to set the clock polarity.
3. Configure SPI_CR1[CPHA] to set the clock phase;
4. Configure SPI_CR1[SPIMS]=0, set as slave mode;
5. Configure SPI_CR1[SPIEN]=1 to enable SPI;
6. Configure SPI_DR, write operation data, and wait for the master to send clock signal.

7.1.3 SPI interrupt source

If SPI interrupt is allowed (SPIIE=1 of IE register), the interrupt will occur when one of the following 4 flag bits is set to 1.

Note: all four of these flag bits must be cleared by software.

1. At the end of each byte transfer, SPI interrupt flag SPIIF is set to 1.This flag applies to all SPI modes.
2. The write conflict flag WCOL is set to 1 if SPI_DR is written when the data in the sending buffer has not been transferred to the shift register.When this happens, the write to SPI_DR is ignored and does

not write to the send buffer.This flag applies to all SPI modes.

3. When SPI is configured to work as a master device in a multi-master mode and NSS is pulled down to low level, the mode error flag MODF is set to 1.When a mode error occurs, the SPIMS and SPIEN bits are cleared to disable SPI and allow another master device to access the bus.

4. The receive overflow flag RXOVR is set to 1 when SPI is configured to be slave device and the transfer ends at one time, while the receive buffer holds the data from the last transmission unread.The newly received bytes will not be transferred to the receive buffer, allowing the previously received bytes to be read. The data bytes that caused the overflow are lost.

7.1.4 SPI working modes

SPI works in several ways: 3-line SPI, 4-line slave/multi-master, and 4-line single-master. The work mode of SPI is configured by SPI_CR1[NSSMOD].

When SPI_CR1[NSSMOD]=00, SPI will work in 3-wire mode and NSS port will not be connected to IO port, which can be used in master-slave mode. Since there is no NSS signal as the device selection signal, it is required that there cannot be more than one slave machine on the bus, that is, point-to-point communication can only be carried out. The connection mode of the master slave machine is shown in Figure 7-1.

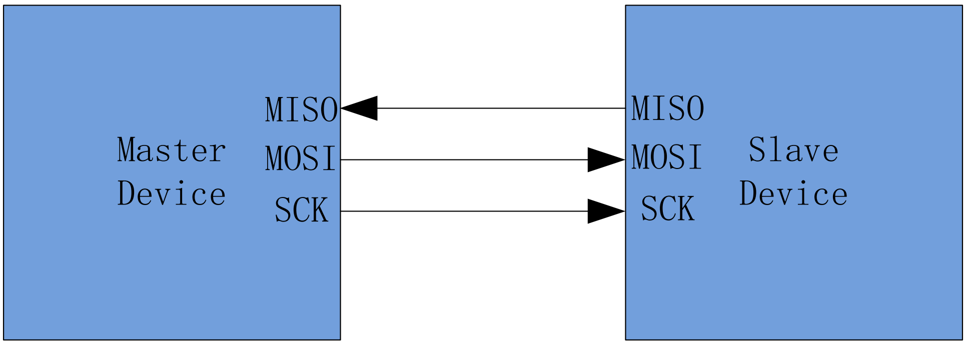


Figure 7-1 Diagram of 3-line single-master mode and 3-line single-slave mode connection

When SPI_CR1[NSSMOD]=01, SPI will work in 4-wire mode, NSS port will be used as the input port to detect and select the communication number. When SPI_CR0[SPIMS]=1, this mode is multi-master mode.When SPI_CR0[SPIMS]=0, this mode of operation is slave mode. For multi-master mode, when the NSS of a master on the bus is pulled down, the main opportunity sets SPI_CR1[MODF] to 1 and changes it to slave mode, and SPI is disabled. For slave mode, SPI will start the transfer when slave NSS is pulled down two system cycles.The connection mode of multiple masters is shown in Figure 7-2.

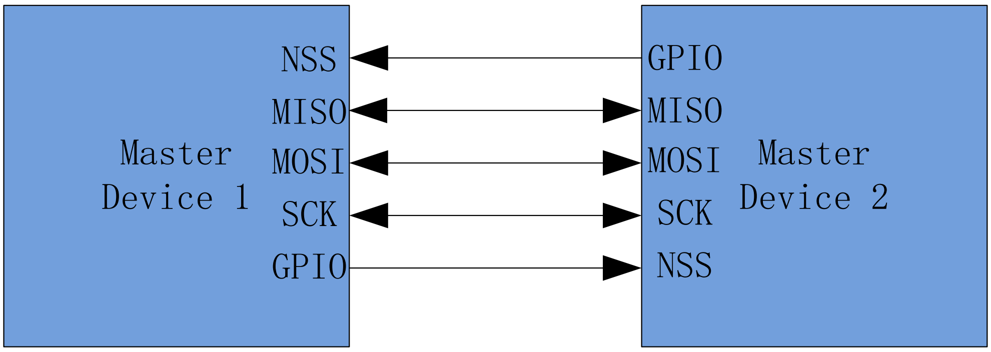


Figure 7-2 Multi-master mode connection diagram

When SPI_CR1[NSSMOD]=1x, SPI will work in 4-wire mode, which is only applicable to master mode. In this way, NSS signal is the output signal, which can be controlled by writing the value of SPI_CR1[NSSMOD0]: when SPI_CR1[NSSMOD0]=1, NSS port will output high level; when SPI_CR1[NSSMOD0]=0, NSS port will output low level. The connection between a single master and a single slave is shown in Figure 7-3.

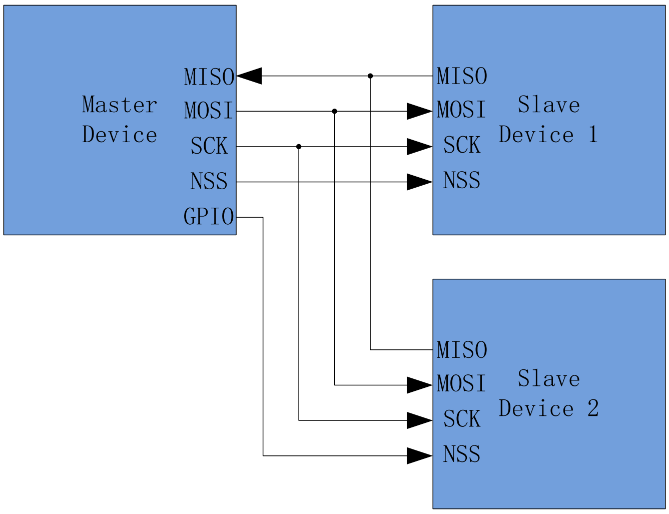


Figure 7-3 4-line single-master mode and 4-line slave mode connection diagram

7.1.5 Serial clock timing sequence

Configure the clock control selector bit in register SPI_CFG to select one of four combinations of serial clock phase and polarity. The CPHA bit of the SPI_CFG register selects one of two clock phases (the edge used to latch data).The CPOL bit of the SPI_CFG register selects one of the high level effective clock and the low level effective clock. The master and slave devices must be configured to use the same clock phase and polarity. Note: SPI should be prohibited during clock phase and polarity changes (by clearing SPIEN bits). The timing sequence relationship between clock and data line in the mater mode is shown in Figure 7-4. Timing sequence relationship between clock and data line in the slave mode such as (CPHA=0) in Figure 7-5 and (CPHA=1) in Figure 7-6.

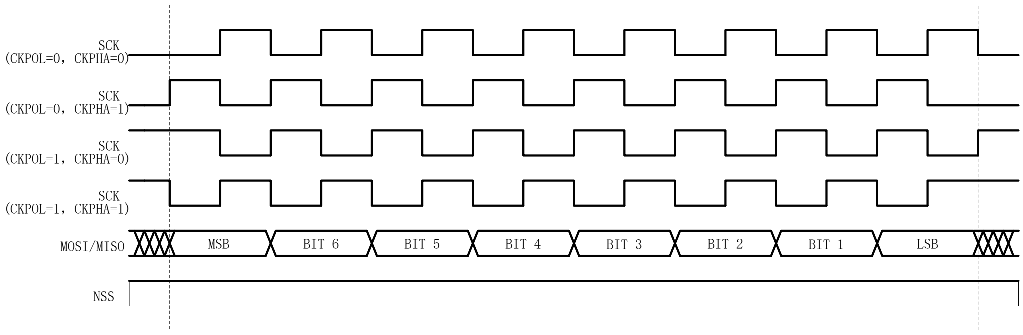


Figure 7-4 Master mode data/clock timing diagram

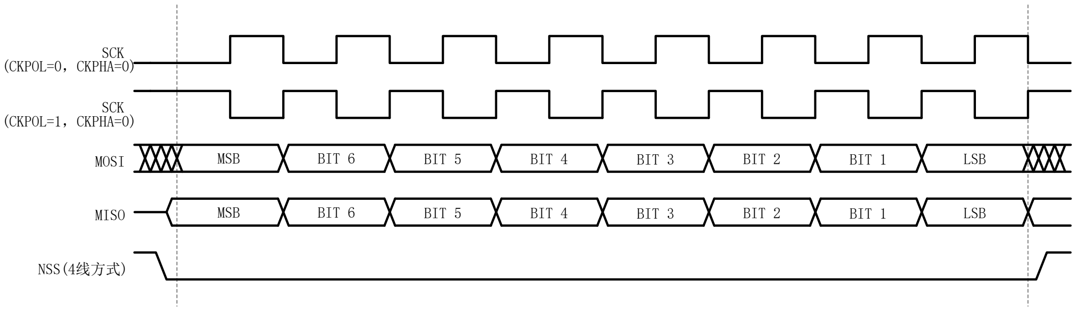


Figure 7-5 Slave mode data/clock timing sequence diagram (CPHA=0)

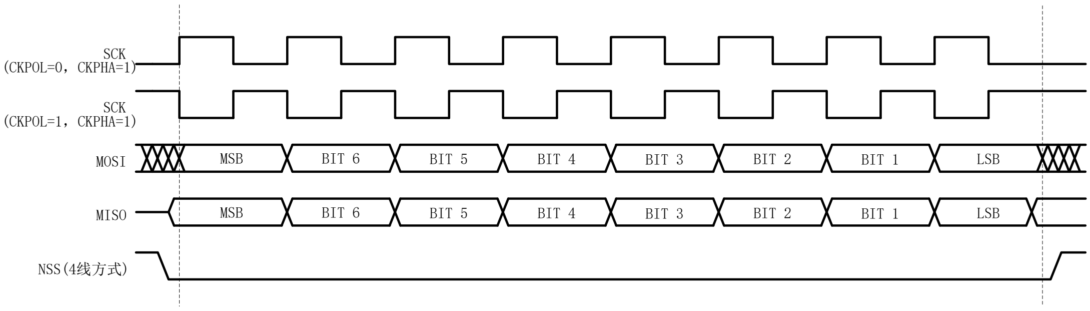


Figure 7-6 Slave mode data/clock timing sequence diagram (CPHA=1)

7.2 SPI register

7.2.1 SPI_CR0 (0x4030)

Table 7-1 SPI_CR0 (0x4030)

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

Bit	Name	Function
[7]	SPIBSY	When a SPI transport is in progress (master or slave mode), the bit is set to logical 1.
[6]	SPIMS	Master/slave mode setting 0: Slave 1: Master
[5]	CPHA	SPI clock phase 0: Sample data at the first edge of SCK period 1: Sample data at the second edge of SCK period
[4]	CPOL	SPI clock polarity 0: The idle level is low 1: The idle level is high
[3]	SLVSEL	When NSS pin is low level, the bit is set to 1, indicating SPI is the selected slave device.When NSS pin is high level (not selected as slave device), the bit is cleared.This bit does not indicate the instant value of the NSS pin, but the de-noising signal input by the pin.
[2]	NSSIN	This bit indicates the current value of the NSS pin when reading the register. The signal is not de-noised.
[1]	SRMT	Empty flag of shift register (valid only in slave mode) This bit is set to 1 when all data is moved into/out of the shift register and no new data can be read from the sending buffer or written to the receive buffer. When the data byte is transferred from the sending buffer to the shift register or SCK changes, the bit is cleared. Note: in the main mode, SRMT = 1
[0]	RXBMT	Receive the null flag of the buffer (valid only in slave mode) This bit is set to 1 when the receiving buffer is read and there is no new data. If new data is not read in the receive buffer, the bit is cleared. Note: in the master mode, RXBMT = 1
Phase mode/clock polarity: 00: Rising edge receives, descending edge sends, idle level is low 01: Rising edge sends, descending edge receives, idle level is high 10: Rising edge sends, descending edge receives, idle level is low 11: Rising edge receives, descending edge sends, idle level is high		

7.2.2 SPI_CR1 (0x4031)

Table 7-2 SPI_CR1 (0x4031)

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

Bit	Name	Function
[7]	SPIIF	SPI interrupt flag bit After one data(8 bits) is transferred at a time, this is pulled up by the hardware.This bit must be cleared by software.
[6]	WCOL	Write conflict flag bit When TXBMT is 0, write SPI_DR and pull this bit up, indicating that SPI data registers were written during data transmission. This bit must be cleared by software
[5]	MODF	Pattern error flag bit This bit is set when master mode conflict is detected (NSS is low, SPIMS = 1 and NSSMD[1:0]=01). This bit must be cleared by software.
[4]	RXOVR	Receiving overrun flag (valid only in the slave mode) The last bit of the current transfer has been moved into the SPI shift register and the receiving buffer still holds data that was not read in the previous transfer, this bit is set to logical 1 by hardware (and generates a SPI interrupt).This bit will not be automatically cleared by hardware, must be cleared by software.
[3:2]	NSSMOD	Select NSS working mode: 00: 3 line slave mode or 3 line master mode. NSS signals are not connected to the port pin. 01: 4 line slave mode or multi-master mode (default). NSS is always the input of the device. 1x: 4 line single master mode. NSS is assigned an output pin and outputs the value of NSSMOD0.
[1]	TXBMT	Sending buffer empty flag The bit is cleared when new data is written to the sending buffer. When data from the sending buffer is transferred to the SPI shift register, the bit is set to 1, indicating that new data can be written to the sending buffer.
[0]	SPIEN	SPI Enable Bit 0: Disable 1: Enable

7.2.3 SPI_CLK (0x4032)

Table 7-3 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	Function

[7:0]	SPI_CLK	SPI clock frequency setting, it is valid in the master mode, and can be written only when SPIEN=0.
		fsck = sysclk/(2x(SPI_CLK[7:0] + 1))
		For example: if sysclk = 24MHz,SPI_CLK=0x04,
		fsck= 24000000/(2x(4+1))=2400kHz

7.2.4 SPI_DR (0x4033)

Table 7-4 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	Function
[7:0]	SPI_DR	The SPI_DR register is used to send and receive SPI data.In the master mode, when writing to the SPI_DR, the data is placed in the sending buffer and start to send. Reading SPI_DR returns the contents of the receive buffer.

8 UART

8.1 UART Operation Descriptions

8.1.1 Mode 0

Mode 0 works in shift mode and can used to expand IO port. TXD is clock bus, RXD is data bus and clock frequency is fcpu_clk/12. The data is transmitted from lowest bit to highest bit. When UT_CR[REN] = 0, UART wrok in send mode and vice verse.

To send data, write the data to UT_DR and reset UT_CR[TI]. TXD will output shift pulse and RXD will output data in UT_DR. UT_CR[TI] is set to 1 at the end of transmission.

To receive data, reset UT_CR[RI] and set UT_CR[REN] to 1. TXD will output shift pulse and RXD will receive the data. UT_CR[RI] is set to 1 at the end of receiving data and UT_DR can be read to get the received data.

8.1.2 Mode 1

Mode 1 supports full-duplex and half-duplex modes. TXD is send data bus, RXD is receive data bus. 10bit protocol: 1bit start + 8 data bits (UT_DR) + 1bit stop. The baud rate is configured with frequency division according to UT_BAUD.

To send data, write the data to UT_DR and reset UT_cR[TI]. TXD will output 10 bit data, and UT_CR[TI] is set to 1 at the end of transmission.

To receive data, set UT_CR[REN] to 1 and then reset UT_CR[RI]. RXD receive the data by UART. UTCR[RB8] and UT_CR[RI] are set to 1 at the end of receiving data, and UT_DR can be read to get the received data.

8.1.3 Mode 2

Mode 2 supports full-duplex and half-duplex modes. TXD is send data bus, RXD is receive data bus. 11bit protocol: 1bit start + 9 data bits(UT_DR + UT_CR[RB8]/UT_CR[TB8]) + 1bit stop. The baud rate is configured with frequency division according to UT_BAUD.

To send data, write the data to UT_DR, configure UT_CR[TB8] and reset UT_cR[TI]. TXD will output 11 bit data, and UT_CR[TI] is set to 1 at the end of transmission.

To receive data, set UT_CR[REN] to 1 and then reset UT_CR[RI]. RXD receive the data by UART. UT_CR[RI] are set to 1 at the end of receiving data, and the data is 8 data bits(UT_dR) + 9th bit(UT_CR[RB8]).

8.1.4 Mode 3

The basic operation is the sane as Mode 2, however, the baud rate configuration is the same as Mode 1.

8.1.5 UART interrupt source

When UART interrupts are enabled (ES0 = 1), the following two flags will generate an interrupt when one of they is set.

Note: All of the following bits must be cleared by software

1. Sending interrupt flag TI will be set to 1 by hardware after UART has completed sending a group of data (8bit for Mode 0 & Mode 1, 9bit for Mode 2 & Mode 3)
2. Receiving interrupt flag RI will be set to 1 by hardware after UART has completed receiving a group of data and stop bit.

8.2 UART Register

8.2.1 UT_CR (0x98)

Table 8-1 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	Function
[7:6]	MOD	Mode Control 00: Mode 0: Shift register The baud rate is fixed at 2MHz in this mode Baud rate = f_{cpu_clk} / 12= 24MHz / 12 = 2MHz 01: Mode 1:8-bit UART The baud rate is calculated by the follow formula: f_{cpu_clk} / (16 × (1+UT_BAUD[BAUD_SEL]) × (UT_BAUD + 1)) 10: Mode 2: 9-bit UART The baud rate is fixed at 750KHz in this mode Baud rate = f_{cpu_clk} / 32 = 24MHz / 32 = 750KHz 11: Mode3: 9-bit UART The baud rate is calculated by the follow formula: f_{cpu_clk} / (16 × (1 + UT_BAUD[BAUD_SEL]) × (UT_BAUD + 1))
[5]	SM2	0: Does not allow multi-thread cpu operation. 1: Allow multi-thread cpu operation.
[4]	REN	0: Does not allow serial input operation. 1: Allow serial input operation and and cleared by software.
[3]	TB8	To set the 9th bit of data transmittion in the Mode 2 and Mode 3. It is cleared by hardware.
[2]	RB8	To set the 9th bit of data receiving in the Mode 2 and Mode 3. It will work as stop bit when SM2 is 0. This bit cannot be used in the mode 0
[1]	TI	Sending finished interrupt flag. It will be set to 1 by hardware after completing the data transfer. It is cleared by software.
[0]	RI	Receiving finished interrupt flag, it will be set to 1 by hardware after data has been finished to receive. It is be cleared by software.

8.2.2 UT_DR (0x99)

Table 8-2 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	Function
[7:0]	UT_DR	Send/Receive data.

8.2.3 UT_BAUD (0x9A, 0x9B)

Table 8-3 UT_BAUDH (0x9B)

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

Table 8-4 UT_BAUDL (0x9A)

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

Bit	Name	Function
[11:0]	UT_BAUD	Setting of baud rate in the Mode 1 & Mode 3.

9 MDU

Multiply and Division Unit (MDU) is a built-in Multiply and DIV processor, which supports 16bit×16bit multiplication and 32bit/16bit division. It supports both unsigned and signed multiplication. The DIV supports unsigned division. MDU mode is set by register MD_CR.

Multiply mode:

Register (MD_MC3~MD_MC0) = register {MD_MAH、MD_MAL} * Register {MD_MBH、MD_MBL}

Note:

Registers MD_MC3~2 reuse registers MD_MAH and MD_MAL, while registers MD_MC1~0 reuse registers MD_MBH and MD_MBL. In a multiplication operation, software should write then multiplier to MD_MAH/MD_MAL and then multiplicand to MD_MBH/MD_MBL. The result of the multiplication is read using software, by reading the higher 16 bits from register MD_MAH/MD_MAL, and the lower 16 bits from register MD_MBH/MD_MBL.

One multiplication process consumes single clock cycle time. The result can be shifted to the right according to ALIGN (MDU_CR[3:2]).

Division mode:

MD_DC3~0= MD_DA3~0 / MD_DB1~0.

MD_DD1~0= MD_DA3~0 % MD_DB1~0.

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

Note:

Registers MD_DC3~0 reuse registers MD_DA3~0, while register MD_DD1~0 reuse registers MD_DB1~0. In a division operation, software should write dividend to MD_DA3~0 and divisor to MD_DB1~0. The result of the division is read using software with the quotient from register MD_DC3~0, and remainder from register MD_DD1~0.

One division process consumes about 16 clock cycles. Software can read register DIVDNOE to check the division status. When DIVDNOE changes to 0 from 1, it means the division process has end.

9.1 Multiplication Using Step:

Step 1: Software sets the register MDSN (MD_CR[1]) according to the multiplication type. It is set to 0 for unsigned multiplication, and 1 for signed multiplication. Meantime, software sets the register ALIGN(MD_CR[3:2]) according to the result shift.

Step 2: Software write multiplier to MD_MAH/MD_MAL, and write multiplicand to MD_MBH/MD_MBL.

Step 3: Software reads the higher 16 bits of the product from register MD_MAH/MD_MAL, and the lower 16 bits from register MD_MBH/MD_MBL.

9.2 DIV Using Step:

Step 1: Software writes dividend to MD_DA3~0, and divisor to MD_DB1~0

Step 2: Software sets register DIVSTA (MD_CR[0]) to 1 for starting the division

Step 3: Software reads register DIVDONE(MD_CR[7]). 1 means division process has been completed, 0 means waiting for new result.

Step 4: Software reads quotient from MD_DA3~0, and remainder from MD_DB1~0.

9.3 Note

Note 1: If divisor is set to 0, the DIV will generate an error flag, which is register DIV_ERR(MD_CR[6]). The flag will be keep to 1, until the MD_DB is not zero in next computing.

Note 2: When DIV is busy, the quotion and the remainder is unknown, and it will recover to normal until division process is complished, that is register DIVDONE=1.

Note 3: When DIV is busy, changing divisor or division value will not affect the final result, unless set DIVSTA again to restart a new division process.

Note 4: The input data registers of multiplier and divider only have one level, so interrupts can change the result. For example, after loading the multiplicand to MD_MA and ready to load the multiplier to MD_MB, interrupt, which multiplier is also used in the service routine, occurs. However, MD_MA has already changed after interrupt finished. Therefore, software developers should take reasonable metheds to avoid the happening of such situation.

9.4 MDU Register

9.4.1 MDU_CR (0xC1)

Table 9-1 MDU mode control and status

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

Bit	Name	Function
[7]	DIVDONE	DIV complishment flag 0: DIV is busy. 1: DIV process has complished.
[6]	DIVERR	DIV divisor error flag 0: last divison process is wrong (divisor is zero). 1: last divison process is correct (divisor is not zero).
[5:4]	RSV	Reserved
[3:2]	ALIGN	The right shift mode setting for multiply process product. 00: not right shift 01: right shift 8bit 10: right shift 12bit 11: right shift 15bit
[1]	MDSN	Multiply sign setting 0: unsigned computing. 1: signed computing.
[0]	DIVSTA	DIV start, software sets this bit to 1 to start DIV computing process. When process is complished, the bit will be cleared by hardware automatically. 0: DIV is not started, last DIV process has been complished. 1: DIV is started and is busy.

9.4.2 MD_MBL (0xCA)

Table 9-2 Multiplier Bit [7:0] (write only), or multiply product Bit [7:0] (read only)

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

Bit	Name	Function
[7:0]	MD_MBL	Multiplier Bit [7:0](write only) or multiply product [7:0](read only)

9.4.3 MD_MBH (0xCB)

Table 9-3 Multiplier Bit [15:8](write only), or multiply product Bit [15:8](read only)

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

Bit	Name	Function
[7:0]	MD_MBH	Multiplier Bit [15:8](write only) , or multiply product Bit [15:8] (read only)

9.4.4 MD_MAL (0xC2)

Table 9-4 Multiplicand Bit [7:0] (write only) or multiply product Bit [23:16] (read only)

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

Bit	Name	Function
[7:0]	MD_MAL	Multiplicand Bit[7:0](write only) or multiply product [23:16](read only)

9.4.5 MD_MAH (0xC3)

Table 9-5 Multiplicand Bit [15:8] (write only) or multiply product [31:24] (read only)

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

Bit	Name	Function
[7:0]	MD_MAH	Multiplicand Bit [15:8] (write only) or multiply product [31:24] (read only)

9.4.6 MD_DA0 (0xC4)

Table 9-6 Dividend [7:0] (Write only) or quotient [7:0] (Read only)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MD_DA0							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Function**
[7:0]	MD_DA0	Dividend [7:0] (Write only) or quotient [7:0] (Read only)

9.4.7 MD_DA1 (0xC5)

Table 9-7 Dividend [15:8] (Write only), or quotient [15:8] (Read only)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MD_DA1							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Function**
[7:0]	MD_DA1	Dividend A[15:8] (Write only), or quotient [15:8] (Read only)

9.4.8 MD_DA2 (0xC6)

Table 9-8 Dividend [23:16] (Write only)or quotient [23:16](Read only)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MD_DA2							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Function**
[7:0]	MD_DA2	Dividend A[23:16] (Write only),or quotient[23:16] (Read only)

9.4.9 MD_DA3 (0xC7)

Table 9-9 Dividend [31:24] (Write only), or quotient[31:24] (Read only)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MD_DA3							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Function**
[7:0]	MD_DA3	Dividend A[31:24] (Write only), or quotient[31:24](Read only)

9.4.10 MD_DB0 (0xCC)

Table 9-10 Divisor [7:0] (Write only), or remainder [7:0] (Read only)

Bit	7	6	5	4	3	2	1	0
Name	MD_DB0							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:0]	MD_DB0	Divisor [7:0] (Write only), or remainder[7:0] (Read only)

9.4.11 MD_DB1 (0xCD)

Table 9-11 Divisor [15:8] (Write only), or remainder [15:8] (Read only)

Bit	7	6	5	4	3	2	1	0
Name	MD_DB1							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:0]	MD_DB1	Divisor [15:8] (Write only), or remainder [15:8] (Read only)

10 PI

10.1 PI Operation

1、 PI format:

$$U(K) = U(k-1) + Kp*(E(k) – E(k-1)) + Ki*E(k) \text{ ----- } (Uk\_min < U(k) < Uk\_max)$$

2、 PI controller is started when PISTA bit is set in the PL_LPF_CR register. The operation is completed after 4 MCU clocks, and the result is updated in PI_UK register.

3、 The data format of control parameters can be selected as Q12 or Q15. By default, the data format of PI_KP and PI_KI are Q12 and others are Q15.

4、 The contents U(k-1) and E(k-1) are the previous updates of U(k) and E(k). To update E(k-1), write value to PI_EK and start PI. To update U(k-1),write value to U(k-1).

MCU has only one PI controller. If the PI controller is used for serval applications, initialization should be done before the operation of PI controller.

```
PI_EK = X;                      initialize E(k-1)
SetBit(PL_CR,PISTA,1);          start PI controller
_nop_();
_nop_();
_nop_();
_nop_();                        wait for the completing of PI operation
PI_UK = X;                      initialize U(k-1)
```

10.2 PI register

10.2.1 PI_LPF_CR (0xF9)

Table 10-1 PI_LPF_CR (0xF9)

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

Bit	Name	Function
[7]	T2SS	Input mode selection of TIM2 stepper motor mode 0: P10 is the direction, P07 is the pulse count 1: P10 is the pulse count of negative direction, P07 is the pulse count of positive direction
[6:3]	RSV	Reserved
[2]	PIRANGE	PI parameter format PI parameter data format 0: Q12. The range of KP, KI (-32768, 32767) corresponds to the actual

		value (-8, 8) 1: Q15. The range of KP, KI (-32768, 32767) corresponds to the actual value (-1, 1)
[1]	PISTA	Start PI controller. To be set by software to logic 1. It will be cleared by hardware at next clock. 0：No start 1：Start
[0]	LPFSTA	Start LPF controller. To be set by software to logic 1. It will be cleared by hardware at next clock. 0：No start 1：Start

10.2.2 PI_EK (0xEA,0xEB)

Table 10-2 PI_EKH (0xEB)

Bit	7	6	5	4	3	2	1	0
Name	PI_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 10-3 PI_EKL (0xEA)

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

Bit	Name	Function
[15:0]	PI_EK	The input error of PI controller. Range: (-32768, 32767).

10.2.3 PI_UK (0xEC,0xED)

Table 10-4 PI_UKH (0xED)

Bit	7	6	5	4	3	2	1	0
Name	PI_UK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 10-5 PI_UKL (0xEC)

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

Bit	Name	Function
[15:0]	PI_UK	The output of PI controller. Range: (-32768,32767).

10.2.4 PI_KP (0xEE,0xEF)

Table 10-6 PI_KPH (0xEF)

Bit	7	6	5	4	3	2	1	0
Name	PI_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 10-7 PI_KPL (0xEE)

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

Bit	Name	Function
[15:0]	PI_KP	Proportion coefficient KP. Range: (-32768,32767).

10.2.5 PI_KI (0xF2,0xF3)

Table 10-8 PI_KIH (0xF3)

Bit	7	6	5	4	3	2	1	0
Name	PI_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 10-9 PI_KIL (0xF2)

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

Bit	Name	Function
[15:0]	PI_KI	Integral coefficient KI. Range: (-32768,32767).

10.2.6 PI_UKMAX (0xF4,0xF5)

Table 10-10 PI_UKMAXH (0xF5)

Bit	7	6	5	4	3	2	1	0
Name	PI_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 10-11 PI_UKMAXL (0xF4)

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

Bit	Name	Function
[15:0]	PI_UKMAX	Maximum limit of UK. Range: (-32768,32767).

10.2.7 PI_UKMIN (0xF6,0xF7)

Table 10-12 PI_UKMINH (0xF7)

Bit	7	6	5	4	3	2	1	0
Name	PI_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 10-13 PI_UKMINL (0xF6)

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

Bit	Name	Function
[15:0]	PI_UKMIN	Minimum limit of UK. Range: (-32768,32767).

11 LPF

11.1 LPF operation

1、 LPF format:

Y(k) = Y(k-1) + LPF_K*(X(k) – Y(k-1))

2、 LPF controller is started when LPFSTA is set in the PI_LPF_CR. The operation is completed after 4 MCU clocks, and the result is updated in LPF_Y register.

3、 The content of Y(k-1) is the previous update of Y(k). If you want to update Y(k-1), please write the value to the LPF_Y.

11.2 LPF register

11.2.1 PI_LPF_CR (0xF9)

Table 11-1 PI_LPF_CR (0xF9)

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

Bit	Name	Function
[7:1]	Please refer to Table 10-1	
[0]	LPFSTA	LPF starts, it’s set by software and cleared by hardware at the next moment 0: not start 1: start

11.2.2 LPF_K (0xDD)

Table 11-2 LPF_K (0xDD)

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

Bit	Name	Function
[7:0]	LPF_K	Low pass filtering coefficient The value range is (-128,127) corresponds to the actual value (-1,1).

11.2.3 LPF_X (0xDE,0xDF)

Table 11-3 LPF_XH (0xDF)

Bit	7	6	5	4	3	2	1	0
Name	LPF_X[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 11-4 LPF_XL (0xDE)

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

Bit	Name	Function
[15:0]	LPF_X	The input values,the value range is (-32768,32767)

11.2.4 LPF_Y (0xE6,0xE7)

Table 11-5 LPF_YH (0xE7)

Bit	7	6	5	4	3	2	1	0
Name	LPF_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 11-6 LPF_YL (0xE6)

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

Bit	Name	Function
[15:0]	LPF_Y	The output value,the value range is (-32768,32767)

12 FOC/SVPWM

12.1 FOC/SVPWM operation

12.1.1 Introduction

The FOC/SVPWM module is used in sensorless FOC, sensiable FOC and sensiable SVPWM applications. Since SVPWM is a subset of FOC module, the following FOC/SVPWM module is referred to as FOC module for short. FOC acts as a stand-alone module that stops the clock when it is not working.The FOC_EN of register DRV_CR is used to enable the FOC module. Before operating the FOC module, the FOC_EN of DRV_CR must be changed to one, otherwise the FOC module cannot work and the relevant FOC registers are in the reset state and cannot be written.

FOC module contains a angle module, a PI controller, a coordinate transform module and a output module. FOC control without HALL sensors can be realized by using the internal angle estimation module. MCU may also process the HALL signals to realize the FOC control with Hall sensors. The internal in FOC module contains a closed current loop, user can output six channel PWM signals to drive the motor according the reference value ID and IQ provided by the user. At the same time, the ADC samples the current signal to implement closed loop current control.

A) FOC without HALL sensors: the angle estimation module is used for coordinate transforms. Meanwhile, the speed is provided to realize the closed speed loop and back-EMF is used to starting.

B) FOC with HALL sensors (single HALL/dual HALL/triple HALL): FOC module provides the angle input interface. The angle value is obtained by utilizing the HALL signals, with the angle value sent to the FOC module.

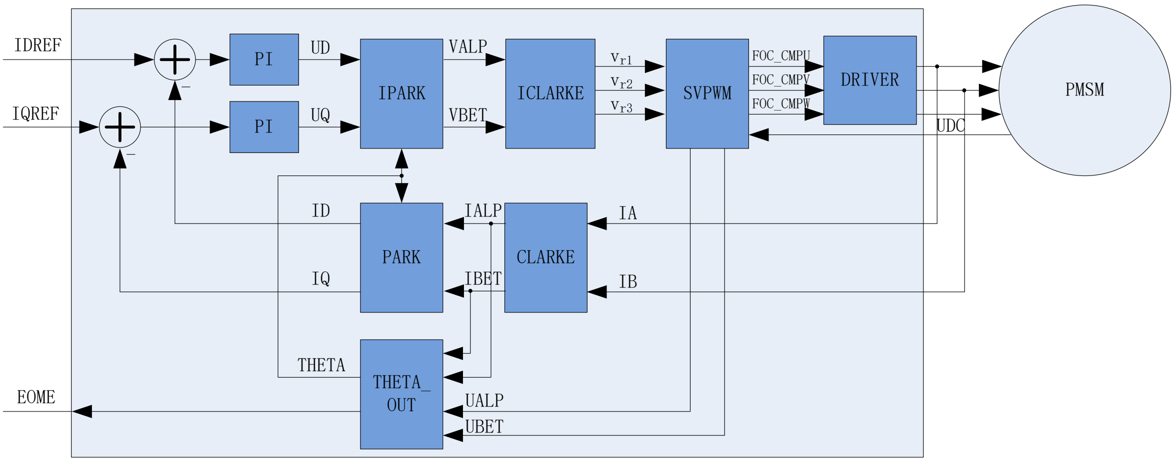


Figure 12-1 FOC principle block diagram

12.1.2 Reference input

The FOC module is closed-loop control based on current feedback, so the d-axis reference current (IDREF) and the Q-axis reference current (IQREF) are used as reference inputs. If the speed-current double loop control is needed, the speed signal is processed by the MCU or PI module according to the output speed EOME of the FOC module to achieve the speed outer loop control.

12.1.3 PI controller

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

1、 Rotor flux control: PI controller of axis D, reference current IDREF minus feedback current ID as deviation input, proportional coefficient DKP and integral coefficient DKI to adjust the performance of PI controller, DMAX and DMIN to limit the output, and finally output D axis voltage UD.

2、 Rotor torque control: PI controller of q-axis, reference current IQREF minus feedback current IQ as deviation input, proportional coefficient QKP and integral coefficient QKI to adjust PI controller performance, QMAX and QMIN to limit the output, and finally output q-axis voltage UQ.

3、 Angle estimation: PI controller of the estimator, proportional coefficient EKP and integral coefficient EKI adjust the performance of PI controller, and finally output estimates Angle ETHETA.

4、 PLL estimation: PI controller of PLL estimator, proportional coefficient PLLKP and integral coefficient PLLKI adjust the performance of PI controller, and finally output estimation of reverse electromotive force EALPHA and EBETA.

12.1.4 Coordinate transforms

12.1.4.1 Inverse Park transform

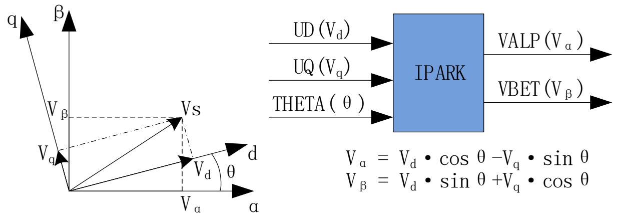


Figure 12-2 PARK inverse transformation

After passing through the PI controller of d-axis and q-axis, UD and UQ components of the voltage vector rotating the d-q coordinate system can be obtained.At this point, it needs to be converted back to 3-phase motor voltage by inverse transformation.First, the PARK inversion is used to transform the voltage vector from the 2-axis rotation d-q coordinate system to the 2-axis stationary α-β coordinate system.

12.1.4.2 Inverse Clarke transform

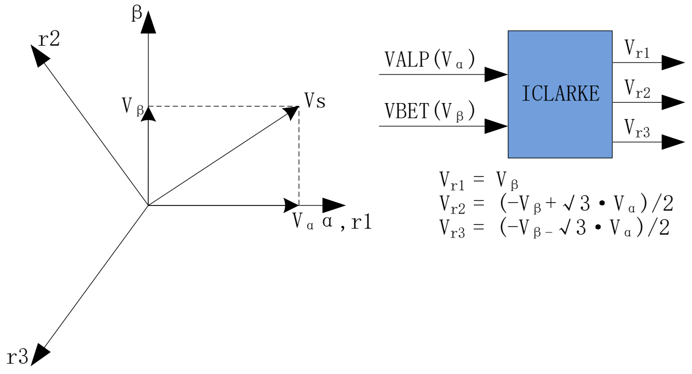


Figure 12-3 CLARKE inverse transformation

Inverse Clarke is to transform the voltage vector from the stationary two-axis α-β Frame to the stationary three-axis, 3-phase reference Frame of the stator.

12.1.4.3 Clarke transform

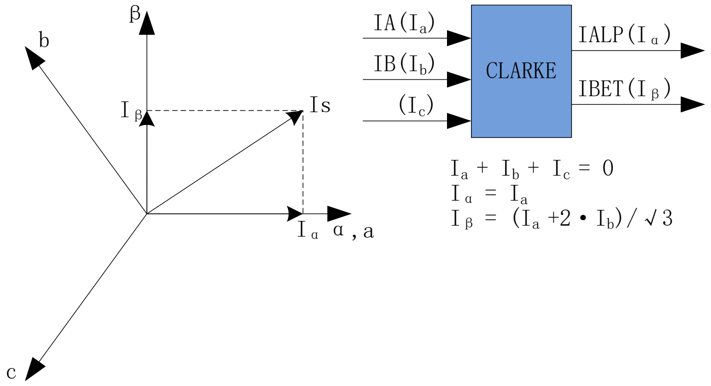


Figure 12-4 CLARKE transformation

Clarke transform transforms the current vector from a 3-axis 2-dimensional stator Frame to a 2-axis α-β stator Frame.

12.1.4.4 Park transform

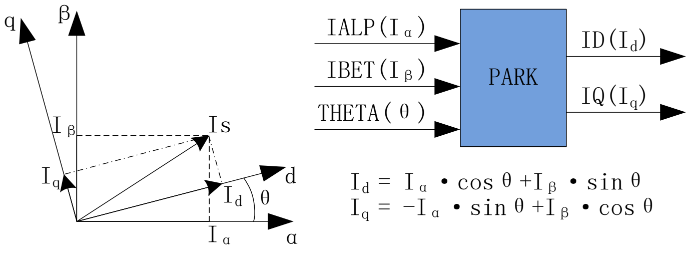


Figure 12-5 PARK transformation

The PARK transform transforms the current vector from the 2-axis α-β stator Frame to the 2-axis d-q Frame rotating with the rotor flux.

12.1.5 SVPWM

Space vector pulse width modulation (SVPWM) algorithm is an important part of FOC control. The main idea is to use the switching of inverter space voltage vector to obtain the quasi-circular rotating magnetic field. This method can decrease the harmonic components of the inverter output current and the harmonic losses of the motor, reducing the torque ripple and have a higher utilization space.

SVPWM generate pulse-width modulation signals for the 3-phase motor voltage control. The process of generating the pulse width for each of the three phases is reduced to a few simple equations. Each of the three inverter outputs can be in one of two states. The inverter output can be connected to eitherthe plus (+) BUS rail or the minus (-) BUS rail, which allows for 2^3 = 8 possible states of the output. Two states in which all three outputs are connected to either the plus (+) BUS or the minus (-) BUS are considered null states because there is no line-to-line voltage across any of the phases. These are plotted at the origin of the SVPWM star. The remaining six states are represented as vectors with 60 degree rotation between each state.

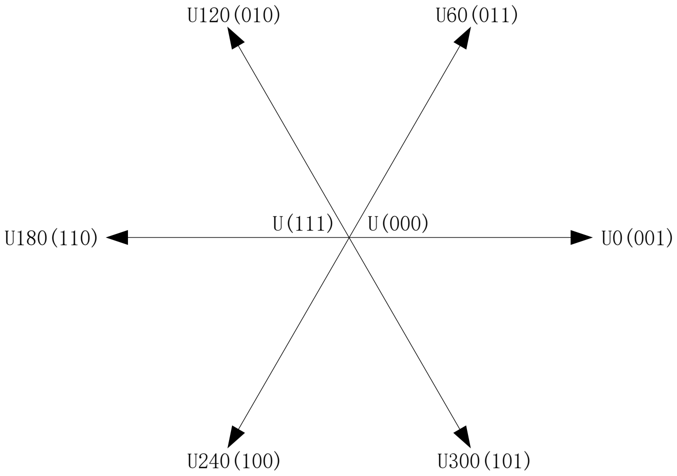


Figure 12-6 SVPWM vector control

The process of SVPWM allows the representation of any resultant vector with the sum of the components of two adjacent vectors. Suppose that UOUT is the desired vector and it lies in the sector between U60 and U0. UOUT is then represented as a time average where during a given PWM period T , U0 is output for 2*T1/T and U60 is output for 2*T2/T . T0 represents a time where no effective voltage is applied into the windings, that is, where a null vector is applied.

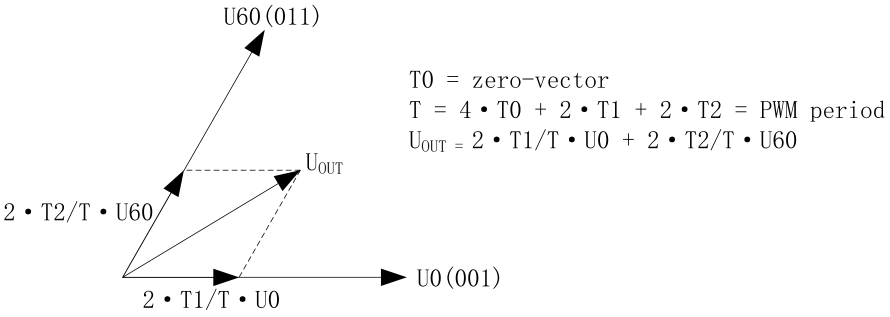

U60(011)
T0 = zero-vector
T = 4 • T0 + 2 • T1 + 2 • T2 = PWM period
U_OUT = 2 • T1/T • U0 + 2 • T2/T • U60
U_OUT
2 • T2/T • U60
U0(001)
2 • T1/T • U0

Figure 12-7 SVPWM voltage synthesis

Table 12-1 State of space vector modulation inverter

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

12.1.5.1 Seven-segment SVPWM

In the single-resistor current sampling mode, FOC algorithm is fixed to seven-segment SVPWM. In the dual-resistors current sampling mode, user can set F5SEG=0 in FOC_CR2 register to choose the seven-segment SVPWM as the output mode.

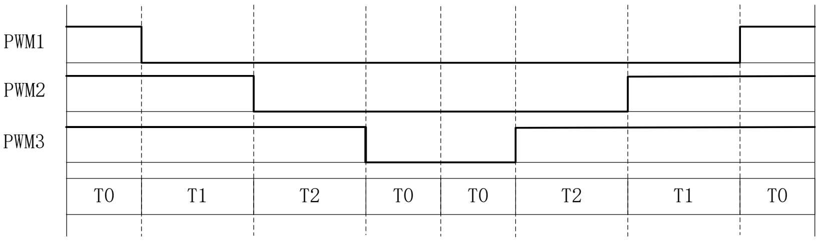


Figure 12-8 Seven-segment SVPWM output level

12.1.5.2 Five-segment SVPWM

Five-segment SVPWM is only used in the dual-resistors current sampling mode. User is required to set F5SEG=1 in FOC_CR2.

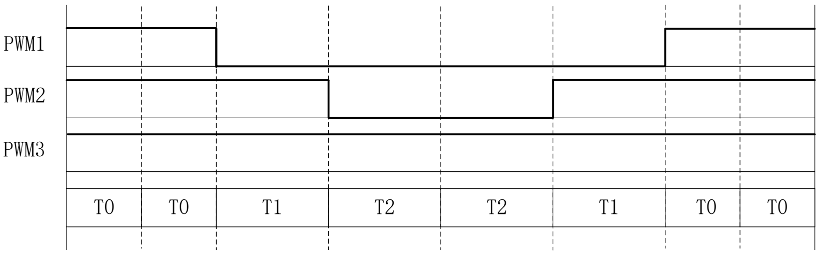


Figure 12-9 Five-segment SVPWM output level

12.1.6 Overmodulation

Overmodulation can be used in single/double/triple resistor mode. Configure the OVMDL of the FOC_CR1 register to enable overmodulation. After overmodulation is enabled, the output will be amplified by 1.15 times, that is, FOC__UD and FOC__UQ and the related limit values MAX/MIN are both amplified by 1.15 times.

12.1.7 Dead zone compensation

Deadzone compensation is only applied in dual/triple resistor mode. Configure the FOC_TSMIN register to set the dead band compensation value. The output after compensation will be a little larger than before compensation, and the sine of the current will be better.

12.1.8 Current and voltage sampling

The FOC module needs to collect the bus voltage and three-phase current of the motor. Before the FOC module works, you need to enable the ADC (ADCEN = 1 in the ADC_STA register) and the op amp, and configure the relevant setting registers. But this does not need to configure ADC channel and scan mode. According to the CSM of the FOC_CR1 register, you can choose single resistance current sampling, double resistance current sampling, or three resistance current sampling. In the single resistance current sampling mode, the default channel 4 is the bus current itrip sampling channel; in the dual resistance current sampling mode, the default channel 0 is the i_a sampling channel and the channel 1 is the i_b sampling channel; in the three resistance current sampling mode, the default Channel 0 is the sampling channel for $i_{a,}$, channel 1 is the sampling channel for i_b, channel 4 is the sampling channel for $i_{c,}$ and channel 2 or channel 14 for bus voltage sampling. The channel 2 is the default sampling channel of the bus voltage.

12.1.8.1 Single resistance sampling mode

Configure CSM = 0 in the FOC_CR1 register to select the single-resistance current sampling mode. In the single-resistance current sampling mode, the FOC module samples the bus current itrip (channel 4) twice in the interval where the counter counts up. In the interval where the counter counts down, the bus voltage is sampled after the FOC module operation is completed.

The dead time affects the time for current sampling. The FOC module will automatically adjust the sampling time according to the dead time set by the user to ensure that the sampling is in the middle of the actual power-on time T1', T2'. Meanwhile, the user can set the FOC_TRGDLY register to advance or delay the sampling time. For example, if the MCU clock is 24MHz (41.67ns) and FOC_TRGDLY = 5, the delay is 41.67 * 5 = 208ns. If FOC_TRGDLY is 0xFB (-5), 208ns is advanced.

For single resistance sampling, the sampling window is not wide enough, so the minimum sampling window needs to be artificially made. Users can set TS (TS = minimum window time + dead time) to let the FOC module adjust the PWM waveform according to TS.

12.1.8.2 Dual three resistance sampling mode

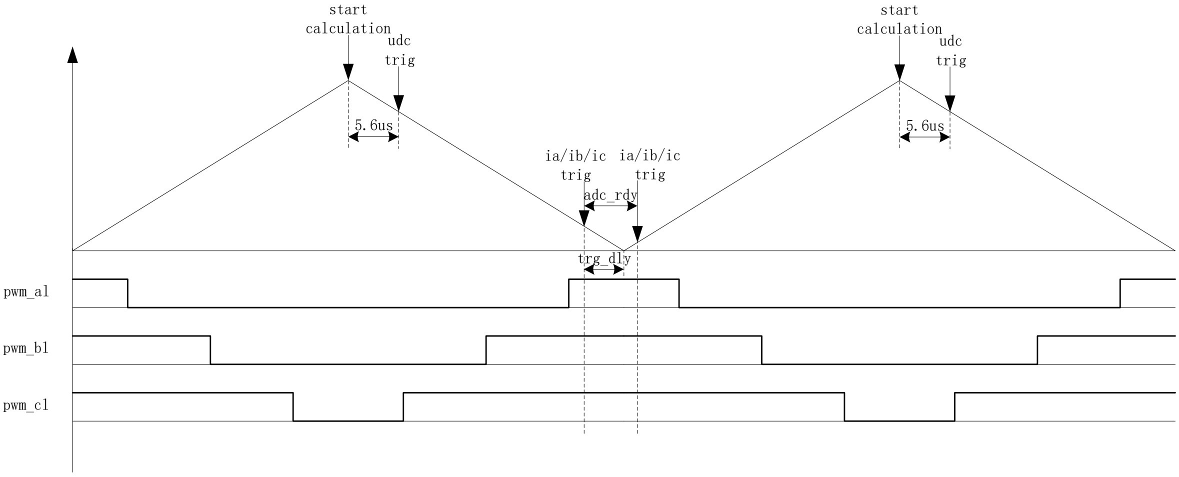


Figure 12-10 Sequential sampling mode of dual and triple resistance current

Configure CSM=10/11 of register FOC_CR1 and DSS=0 of register FOC_CR2, that is, select dual/triple resistance current sequential sampling mode.In double three resistance current sequential sampling mode, through FOC_TRGDLY register set the sampling time of one of the three phase current phase (determined according to the sector's ia/ib/ic). As soon as one phase is sampled the other phase is sampled. In the interval where the counter counts down, the bus voltage is sampled after the FOC module operation is completed. It should be noted that the current sampling timing should be set so that the three-phase current sampling points are in the zero vector interval (i.e. pwm_al, pwm_bl, pwm_cl = 111). For example, if the MCU clock is 24MHz (41.67ns) and FOC_TRGDLY = 0xB2, then when the FOC counter counts down, samples ia/ib/ic 41.67 * 50 = 2.08us before the underflow event. After the sampling is completed, another phase $i_a/i_b/i_c$ is sampled.

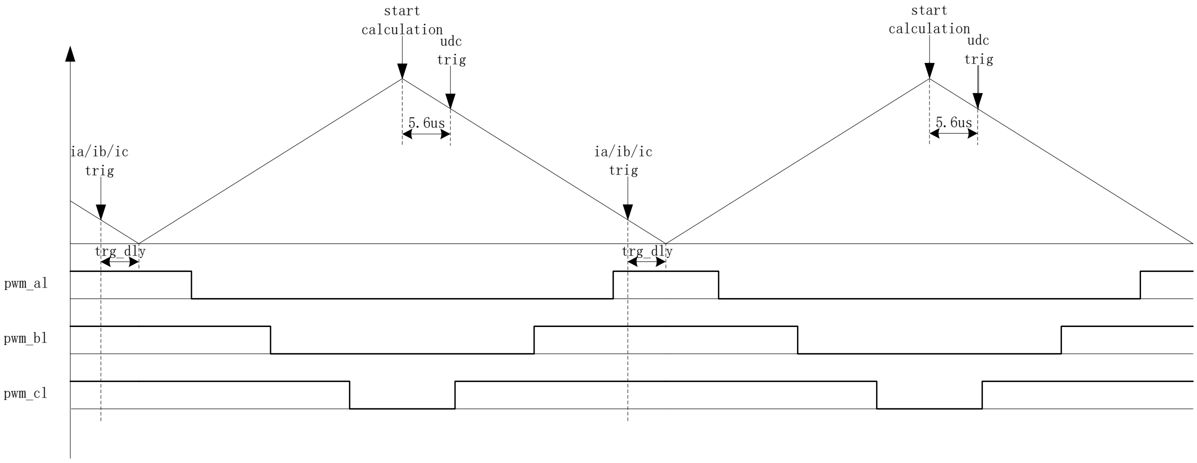


Figure 12-11 Alternating sampling mode of double and triple resistance current

Configure CSM = 10/11 in FOC_CR1 register and DSS = 1 in FOC_CR2 register, that is, select the dual/three resistor current alternate sampling mode. In the double-three resistor current alternating sampling mode, one carrier cycle is calculated once, but each carrier cycle only samples one phase current (which phase of ia/ib/ic is determined according to the sector). The first carrier cycle collects $i_a/i_b/i_c$, and the second carrier cycle collects another phase current $i_a/i_b/i_c$, so that the current acquisition is performed alternately on one of the three-phase currents. In the interval where the counter counts down, the bus voltage is sampled after the FOC module operation is completed. Set the sampling timing for current i_a (channel 0), i_b (channel 1), i_c (channel 4) through the FOC_TRGDLY register. It should be noted that the current sampling timing should be set so that the $i_a/i_b/i_c$ sampling points are all in the zero vector interval (ie, pwm_al, pwm_bl, pwm_cl = 111). For example, if the MCU clock is 24MHz (41.67ns) and FOC_TRGDLY = 0xB2, the FOC counter counts down. 41.67*50 = 2.08μs before the underflow event, the current is sampled.

12.1.8.3 Current sampling reference

Since there are positive and negative phase currents, it is necessary to increase the input current by half of the range before ADC sampling. So it is required to minus the reference value during an operation, which the default value is 0x4000. However, there is a deviation between the default value and the real value, owing to ADC and hardware offsets, so usually the user will required to do a a calibration. The calibration procedure is as the following: When FOC module dosen’t work, and there is no current in the three phases, MCU starts to sample the corresponding channel and do a write to the FOC_CSO register after averaging all the sampled value. If the voltage range is 0~5V, and the reference voltage is 2.5V, FOC_CSO = 2.5/5V*32768 = 16384(0x4000).

1、 When FOC_CR2[CSOC] = 00/11, writing FOC_CSO is the benchmark for modifying ITRIP and IC;

2、 When FOC_ CR2 [CSOC] = 01, writing FOC_CSO is the benchmark for modifying IA;

3、 When FOC_ CR2 [CSOC] = 10, writing FOC_CSO is the benchmark for modifying IB.

12.1.9 Angle pattern

The Angle module includes three parts: angle estimation module, ramp module and estimating angle smooth switching module. The sources of the angle are as follows:

1、 Ramp force angle

2、 Forced angle of pull

3、 Estimated angle of estimator

4、 Forced angle of estimator

Table 12-2 Angle sources

RFAE	ANGM	EFAE	Point source
1	x	x	Ramp force angle
0	0	x	Forced angle of pull
0	1	0	Estimation angle of estimator
0	1	1	Estimated Speed>EFREQMIN: Estimator estimation angle Estimated Speed<EFREQMIN: Estimator force angle

12.1.9.1 Ramp force angle

Ramp force angle consists of angle THETA, speed RTHESTEP, acceleration RTHEACC and ramp counter RTHECNT. The ramp formula is

RTHESTEP (32bit) = RTHESTEP (32bit) + RTHEACC(32bit, the upper 16bit are fixed to zero and the lower 16bit are configurable)

THETA(16bit) = THETA(16bit) + RTHESTEP(high 16bit)

Ramp force angle has the highest priority. Set the RFAE = 1 in FOC_CR1 register to enable the ramp function. In every cycle, ramp module makes a ramp operation and ramp counter adds 1. When the value of the counter reaches RTHECNT, RFAE is cleared by hardware, and then the ramp is over. Then if ANGM=1 in the FOC_CR1 register, the angle comes from estimator, while ANGM=0, the angle comes from force pull angle.

12.1.9.2 Force pull angle

Force pull angle consists of angle THETA and the speed RTHESTEP. The formula is

THETA (16bit) = THETA (16bit) + RTHESTEP (high 16bit)

There are two states in the force pull angle:

1. Setting RFAE = 1 and ANGM = 0 in FOC_CR1 register, and the force pull mode is started after ramp function. The speed RTHESTEP will then be the cumulative result at the end of the ramp. This model can realize the function of uniform strong pull without angle feedback.
2. Setting RFAE = 0 and ANGM = 0 in FOC_CR1, angle comes from the force pull angle without going through the ramp module. The speed RTHESTEP will then be the initial speed writen in the register by software. When RTHESTEP = 0, alignment function is realized. When RTHESTEP != 0,

FOC control with the Hall sensor is enabled. (the principle of FOC control with the Hall sensor is that MCU calculate the existing angle and the speed when the HALL signal is received, and writes them to the THETA and RTHESTEP to make modifications).

12.1.9.3 Estimator angle

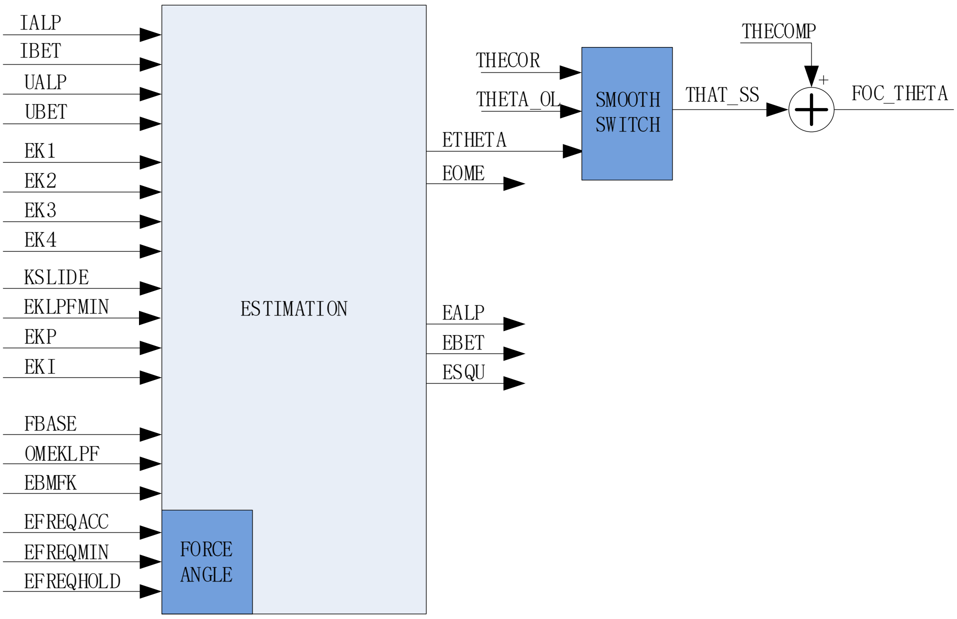


Figure 12-12 Principle block diagram of estimator

The estimator samples the current and voltage of the motor and outputs the angle, speed and back electromotive force based on the motor parameters and control parameters.

1、 Estimated angle of the estimator

The estimator builds the motor model based on the motor parameters and control parameters, and samples the the current and voltage of the motor to update the estimated value and the estimator outputs the angles which are consistent with the real situation. The estimator can be selected with the PLL mode and sliding mode by setting the ESEL in FOC_CR1 register.

2、 Forced angle of the estimator

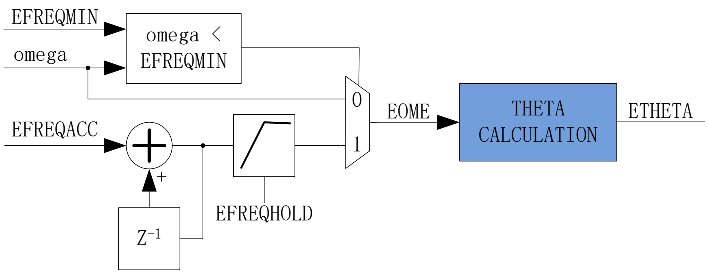


Figure 12-13 Schematic diagram of forced angle of estimator

This function is similar to the ramp function. Due to the small ouput at motor starting, the sampling current is small and there can be a huge difference in angle estimation and the speed. This may result in the launch failure. In this case, the estimator outputs the force angle to make the motor start successfully.

Configuration EFAE = 1 in the register FOC_CR1, when the estimated speed of estimator is less than the minimum value EFREQMIN set by user, function start and mandatory speed start from 0, plus the change EFREQACC in speed each operation cycle. At the same time, the maximum amplitude is limited according to EFREQHOLD and the force speed as the final speed EOME for angle calculation module to calculate the angle estimated value ETHETA. When omega is greater than or equal to EFREQMIN, the output estimated speed omega is the final speed EOME.

3、 Smooth angle switching

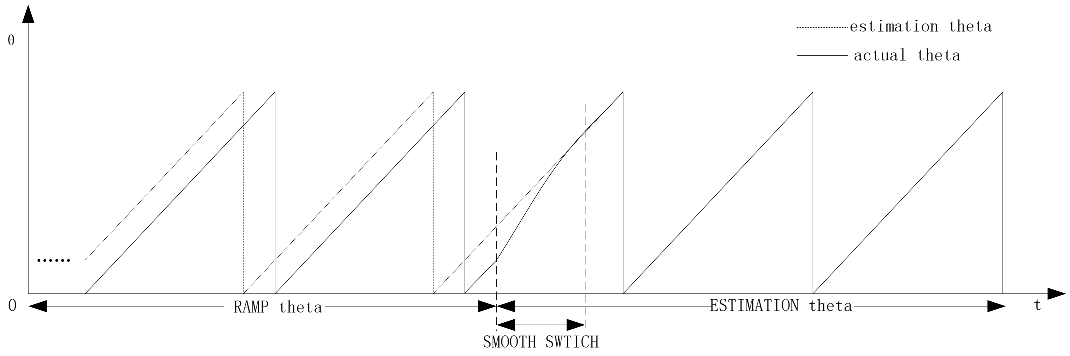


Figure 12-14 Smooth angle switching curve

Setting the RFAE = 1 and ANGM = 1 in FOC_CR1 register, the motor starts with the ramp function. At the end of the ramp, it switches to the estimator mode. The estimator also estimates the angle in the ramp, but usually there is a deviation between the estimated angle and the ramp force angle. If the angle is switched from ramp forced angle to estimated angle after the ramp, there could be motor jitter due to the mutation of angle, so the switching module smoothly is required.

At the end of the ramp, if the devation between estimated angle ETHETA and ramp force angle theta_ol is less than or equal to the THECOR, the devation is considered as very small and ETHETA is used directly as the output angle. If the devation is larger than THECOR, theta_ol will be modified smoothly to close the value ETHETA with the step of the THECOR at every operation cycle. When the devation is less than THECOR, ETHETA is used directly as the output angle.

4、 Angle compensation

The compensation value THECOMP compensates the estimated angle. If the highest bit of THECOMP is 1, it is negative compensation. If the highest bit of THECOMP is 0, it is positive compensation.

12.1.10 Real-time motor parameters

During the operation of FOC module, users can view THETA, the real-time angle value, estimated speed EOME, and other real-time parameters of the motor. Users can judge the real-time state of the motor according to the parameters.The FOC module provides the following real-time parameters:

1. Angle value, THETA
2. Estimated angle, ETHETA, Theta, and the estimated velocity, EOME
3. D axis voltage UD, Q axis voltage UQ
4. D axis current ID, Q axis current IQ
5. IPARK module coordinates conversion of ALPHA axis voltage VALP, BETA axis voltage VBET
6. Bus voltage UDCFLT
7. Three phase current IA,IB,IC
8. CLARKE module coordinate conversion of ALPHA current IALP (IA), BETA axis current IBET
9. ALPHA axis reverse electromotive force EALP, BETA axis reverse electromotive force EBET
10. Inverse electromotive force squared ESQU
11. Power POW

12.1.10.1 Downwind and headwind detection

FOC provides a dedicated downwind and headwind detection module. Set the reference input current IDREF and IQREF to 0, and with the FOC module enabled, the motor state can be determined by reading the estimated angle ETHETA and estimated speed EOME. If ETHETA decreases downward or EOME is a negative value, it is in a reverse rotation. It is then required to brake first and start motor with the force angle mode. If ETHETA increases forward or EOME is a positive value, it is in a forward rotation. At this moment, it could swich to estimator angle mode to start motor directly.

12.1.10.2 Counter electromotive force detection

Estimator estimates the α-axis Back-emf EALF and the β-axis Back-emf EBET with the input motor's parameters and outputs $e\alpha^2 + e\beta^2$, namely ESQU. The user can determine the launch state by using the value of ESQU to implement functions such as locked-rotor protection or phase-break protection.

12.1.10.3 Power

Acording to the current and power-on time obtained in the SVPWM module inside the FOC, as well as the filtered bus voltage, can calculate the power, which is save in FOC__POW.

12.2 FOC register

12.2.1 FOC_CR1 (0x40A0)

Table 12-3 FOC_CR1 (0x40A0)

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

Bit	Name	Function
[7]	OVMDL	Overmodulation enable 0: Disable 1: Enable
[6]	EFAE	ESTIMATION Force angle enable After enabling, the angle is forced to be given by the estimator and automatically switched to the closed loop of the estimator angle 0: Disable 1: Enable
[5]	RFAE	Ramp Force Angle enable After enabling, the angle is forced to be given by the climbing module. At the end of climbing, the angle is automatically switched to estimator mode or force pull mode according to ANGM, and RFAE is cleared by hardware 0: Disable 1: Enable
[4]	ANGM	Angle mode When RFAE=0, the Angle comes from estimator or force pull When RFAE=1, switch the estimator or force pull after climbing 0: force pull mode 1. Estimation model
[3:2]	CSM	Current sample mode X0: single resistance sampling 01: dual resistance sampling 11: three-resistance sampling
[1]	SPWMSEL	Polarity selection in SPWM mode 0: Unipolarity 1: Bipolar
[0]	SVPWMEN	SVPWM/SPWM mode selection 0: SPWM 1: SVPWM

12.2.2 FOC_CR2 (0x40A1)

Table 12-4 FOC_CR2 (0x40A1)

Bit	7	6	5	4	3	2	1	0

Name	ESEL	RSV	F5SEG	DSS	CSOC		UQD	UDD
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7]	ESEL	ESTIMATION select 0: SMO (Sliding mode) 1: PLL (Phase-locked loop), where the FOC_KSILDE register is KP of PLL's PI controller and the FOC_KLPFMIN register is KI of PLL's PI controller
[6]	RSV	Reserved
[5]	F5SEG	Selection of SVPWM modulation mode under FOC double-triple resistance current sampling (single-resistance current sampling is prohibited to be 1) 0:7 Segment selection 1:5 Segment selection
[4]	DSS	Dual three resistance current sampling mode 0: sequential sampling mode: a carrier period sampling two-phase current 1: alternating sampling mode: alternating sampling of two-phase current for each carrier period, and calculation for each carrier period
[3:2]	CSOC	Current sample offset calibrate Configure the bits to write the calibration value to FOC_CSO. For single resistance sampling, configure 00 or 11 to calibrate itrip; for dual-resistance sampling, configure 01 to calibrate ia, configure 10 to calibrate ib; for three resistance sampling, configure 01 to calibrate ia, configure 10 to calibrate ib, configure 00 or 11 to calibrate i_c. 00, 11: itrip and ic 01: ia 10: ib
[1]	UQD	When the q-axis PI controller is disabled, the FOC_UQ value is no longer updated by the PI controller 0: Enable 1: Disable
[0]	UDD	When the d-axis PI controller is disabled, the FOC_UD value is no longer updated by the PI controller 0: Enable 1: Disable

12.2.3 FOC_TSMIN (0x40A2)

Table 12-5 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	Function
[7:0]	FOC_TSMIN	In single-resistance sampling mode, the minimum window reserved for ADC sampling Dead zone compensation in doubl-three resistance sampling mode Range is (0,255)
TS = Sample window △T + dead zone time DT Suppose △T =1us, DT=1us, TS=2us, carrier cycle 62.5us TS=2/62.5*4096=131		

12.2.4 FOC_TGLI (0x40A3)

Table 12-6 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	Function
[7:0]	FOC_TGLI	Narrow pulse elimination when the upper brige is conducting. This function is used in high voltage applications. The upper bridge of the high-voltage driver has a requirement that the minimum conducting pulse that must be greater than a certain value. If this register is set and the conducting pulse is less than the set value, there is no conduction. The value varys from 0 to 255.
It is assumed that narrow pulses less than 1us are eliminated, DT=1us, carrier period is 62.5us, FOC_TGLI=2/62.5*2048=65		

12.2.5 FOC_TBLO (0x40A4)

Table 12-7 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	Function
[7:0]	FOC_TBLO	When the three resistors are the current sampling shielding time, the current bridge conduction time is less than FOC_TBLO, then the current in the phase is not sampled, and the current is obtained by special processing. The range is (0,255)
Assuming that the lower bridge conduction time is less than 1us, FOC_ TBLO =1000/41.67=24		

12.2.6 FOC_TRGDLY (0x40A5)

Table 12-8 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	Function
[7:0]	FOC_TRGDLY	Single resistance sampling mode: ADC current acquisition trigger delay The range is (-128,127) Dual three resistance sampling mode: current acquisition timing TRGDLY[7] set the acquisition time to be on the descending or ascending interval of the counter: TRGDLY[7]=0: ascending interval; TRGDLY [7]=1: descending interval The range is (0,DRV_ARR[6:0])
Single resistance sampling mode: assuming that the MCU clock is 24MHz(41.67ns) and TRGDLY = 5, the delay 41.67*5=208ns; TRGDLY is -5, 208ns advance Double triple resistance sampling mode: assuming that the MCU clock is 24MHz(41.67ns), TRGDLY = 0x85, then when the DRV counter counts down, start to sample 41.67*5=0.208us before the underflow event; if TRGDLY = 0x05, the DRV counter is counted up and samples 41.67*5=0.208us after the underflow;		

12.2.7 FOC_CSO (0x40A6,0x40A7)

Table 12-9 FOC_CSOH (0x40A6)

Bit	7	6	5	4	3	2	1	0
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

Table 12-10 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	Function
[15:0]	FOC_CSO	Current sampling reference Configure the CSOC in the FOC_CR1 register and write the calibration values on the FOC_CSO to calibrate the itrip benchmark for single-resistance sampling mode, ia, ib benchmark for double-resistance sampling mode, and ia, ib, ic benchmark for three-resistance sampling mode. The value range is (0,32767), MSB is constantly 0
Assuming the voltage range of ADC is 0~5V, the reference is 2.5V Then FOC_CSO = 2.5/5V *32768 = 16384(0x4000)		

12.2.8 FOC_RTHESTEP (0x40A8,0x40A9)

Table 12-11 FOC_RTHESTEPH (0x40A8)

Bit	7	6	5	4	3	2	1	0
Name	FOC_RTHESTEP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-12 FOC_RTHESTEPL (0x40A9)

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

Bit	Name	Function
[15:0]	FOC_RTHESTEP	Rate of climb; The format is consistent with FOC_THETA Software write: initial speed Software read: current speed The value range is (-32768,32767) Note: the internal FOC_RTHESTEP is 32bit, the highest bit is sign bit, and the higher 16 bits are configurable.
RTHESTEP(32bit)=RTHESTEP(32bit) + RTHEACC(32bit, higher 16bit is always 0, lower 16bit can be configurable); THETA(16bit) = THETA(16bit) + RTHESTEP(higher 16bit)		

12.2.9 FOC_RTHEACC (0x40AA,0x40AB)

Table 12-13 FOC_RTHEACCH (0x40AA)

Bit	7	6	5	4	3	2	1	0
Name	FOC_RTHEACC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-14 FOC_RTHEACCL (0x40AB)

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

Bit	Name	Function
[15:0]	FOC_RTHEACC	Acceleration of climbing module. The format is consistent with FOC_THETA The value range is (-32768,32767) Note: the internal FOC_RTHEACC is 32bit, the highest bit is sign bit, and the lower 16 bits are configurable.
RTHESTEP(32bit)=RTHESTEP(32bit) + RTHEACC(32bit, higher 16bit is always 0, lower 16bit can be		

configurable);
THETA(16bit) = THETA(16bit) + RTHESTEP(higher 16bit)

12.2.10 FOC_RTHECNT (0x40AC)

Table 12-15 FOC_RTHECNT (0x40AC)

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

Bit	Name	Function
[7:0]	FOC_RTHECNT	Times of ramps = RTHECNT * 256 After the ramp function is enabled (RFAE = 1 in the FOC_CR1 register), the climbing operation is performed once per calculation cycle. When the times of ramps reaches RTHECNT * 256, the ramp ends

12.2.11 FOC_THECOR (0x40AD) shared with BLDC

Table 12-16 FOC_THECOR (0x40AD)

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

Bit	Name	Function
[7:0]	FOC_THECOR	Angle switching correction: At the end of ramp, it is necessary to switch to the estimation mode. Due to the deviation between the estimation angle and the ramp angle, smooth switching is required. The format is consistent with FOC_THETA The value range is (0,255)

12.2.12 FOC_THECOMP (0x40AE,0x40AF)

Table 12-17 FOC_THECOMPH (0x40AE)

Bit	7	6	5	4	3	2	1	0
Name	FOC_THECOMP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-18 FOC_THECOMPL (0x40AF)

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

Bit	Name	Function
[15:0]	FOC_THECOMP	Angle compensation value: after the estimator estimates the angle, add the compensation value to estimated andgle as the final output angle of the estimator; the format is consistent with FOC_THETA The value range is (-32768,32767)

12.2.13 FOC_DMAX (0x40B0,0x40B1)

Table 12-19 FOC_DMAXH (0x40B0)

Bit	7	6	5	4	3	2	1	0
Name	FOC_DMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-20 FOC_DMAXL (0x40B1)

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

Bit	Name	Function
[15:0]	FOC_DMAX	The upper limit of output UD of the PI controller on axis D The value range is (-32768,32767)

12.2.14 FOC_DMIN (0x40B2,0x40B3)

Table 12-21 FOC_DMINH (0x40B2)

Bit	7	6	5	4	3	2	1	0
Name	FOC_DMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-22 FOC_DMINL (0x40B3)

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

Bit	Name	Function
[15:0]	FOC_DMIN	The lower limit of output UD of the PI controller on axis D The value range is (-32768,32767)

12.2.15 FOC_QMAX (0x40B4,0x40B5)

Table 12-23 FOC_QMAXH (0x40B4)

Bit	7	6	5	4	3	2	1	0

Name	FOC_QMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-24 FOC_QMAXL (0x40B5)

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

Bit	**Name**	**Function**
[15:0]	FOC_QMAX	The upper limit of output UQ of the PI controller on axis Q The value range is (-32768,32767)

12.2.16 FOC_QMIN (0x40B6,0x40B7)

Table 12-25 FOC_QMINH (0x40B6)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FOC_QMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-26 FOC_QMINL (0x40B7)

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

Bit	**Name**	**Function**
[15:0]	FOC_QMIN	The lower limit of output UQ of the PI controller on axis Q The value range is (-32768,32767)

12.2.17 FOC_UD(0x40B8,0x40B9)

Table 12-27 FOC_UDH (0x40B8)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
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

Table 12-28 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	Function
[15:0]	FOC_UD	D axis voltage The voltage calculated by the PI controller of axis D The value range is (-32768,32767)

12.2.18 FOC_UQ (0x40BA,0x40BB)

Table 12-29 FOC_UQH (0x40BA)

Bit	7	6	5	4	3	2	1	0
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

Table 12-30 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	Function
[15:0]	FOC_UQ	Q axis voltage The voltage calculated by the PI controller of axis Q The value range is (-32768,32767)

12.2.19 FOC_ID (0x40BC,0x40BD)

Table 12-31 FOC_IDH (0x40BC)

Bit	7	6	5	4	3	2	1	0
Name	FOC_ID[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-32 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	Function
[15:0]	FOC_ID	The ID calculated by the PARK transform. The value range is (-32768,32767)

12.2.20 FOC_IQ (0x40BE,0x40BF)

Table 12-33 FOC_IQH (0x40BE)

Bit	7	6	5	4	3	2	1	0
Name	FOC_IQ[15:8]							
Type	R	R	R	R	R	R	R	R

Reset	0	0	0	0	0	0	0	0

Table 12-34 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	Function
[15:0]	FOC_IQ	The IQ calculated by the PARK transform, The value range is (-32768,32767)

12.2.21 FOC_IBET (0x40C0,0x40C1)

Table 12-35 FOC_IBETH (0x40C0)

Bit	7	6	5	4	3	2	1	0
Name	FOC_IBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-36 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	Function
[15:0]	FOC_IBET	Current IBETA. The value range is (-32768,32767)

12.2.22 FOC_VBET (0x40C2,0x40C3)

Table 12-37 FOC_VBETH (0x40C2)

Bit	7	6	5	4	3	2	1	0
Name	FOC_VBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-38 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	Function
[15:0]	FOC_VBET	VBETA calculated by the IPARK transform The value range is (-32768,32767)

12.2.23 FOC_VALP (0x40C4,0x40C5)

Table 12-39 FOC_VALPH (0x40C4)

Bit	7	6	5	4	3	2	1	0
Name	FOC_VALP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-40 FOC_VALPL (0x40C5)

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

Bit	Name	Function
[15:0]	FOC_VALP	The VALPHA calculated by the IPARK transformation. The value range is (-32768,32767)

12.2.24 FOC_IC (0x40C6,0x40C7)

Table 12-41 FOC_ICH (0x40C6)

Bit	7	6	5	4	3	2	1	0
Name	FOC_IC[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-42 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	Function
[15:0]	FOC_IC	Phase current IC. The value range is (-32768,32767)

12.2.25 FOC_IB (0x40C8,0x40C9)

Table 12-43 FOC_IBH (0x40C8)

Bit	7	6	5	4	3	2	1	0
Name	FOC_IB[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-44 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	Function
[15:0]	FOC_IB	Phase current IB The value range is (-32768,32767)

12.2.26 FOC_IA(0x40CA,0x40CB)

Table 12-45 FOC_IAH (0x40CA)

Bit	7	6	5	4	3	2	1	0
Name	FOC_IA[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-46 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	Function
[15:0]	FOC_IA	Phase current IA The value range is (-32768,32767)

12.2.27 FOC_THETA (0x40CC,0x40CD)

Table 12-47 FOC_THETAH (0x40CC)

Bit	7	6	5	4	3	2	1	0
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

Table 12-48 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	Function

[15:0]	FOC_THETA	Angle.
		Software writing: force pull angle;
		Software reading: current FOC work angle;
		The value range is (-32768,32767)

The value range of THETA is (-32768,32767), corresponding to (-180 °, 180 °)		

Assuming that THETA = 8192, corresponding to 8192/32768 * 180 ° = 45 °

12.2.28 FOC_ETHETA (0x40CE,0x40CF)

Table 12-49 FOC_ETHETAH (0x40CE)

Bit	7	6	5	4	3	2	1	0
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

Table 12-50 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	Function
[15:0]	FOC_ETHETA	Read: the angle estimated by the estimator (the angle before FOC_THECOMP compensates) is in the same format as FOC_THETA Write: initial angle estimated by estimator The value range is (-32768,32767)

12.2.29 FOC_EALP (0x40D0,0x40D1)

Table 12-51 FOC_EALPH (0x40D0)

Bit	7	6	5	4	3	2	1	0
Name	FOC_EALP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-52 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	Function
[15:0]	FOC_EALP	EALPHA estimated by the estimator The value range is (-32768,32767)

12.2.30 FOC_EBET (0x40D2,0x40D3)

Table 12-53 FOC_EBETH (0x40D2)

Bit	7	6	5	4	3	2	1	0
Name	FOC_EBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-54 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	Function
[15:0]	FOC_EBET	EBETA estimated by the estimator The value range is (-32768,32767)

12.2.31 FOC_EOME (0x40D4,0x40D5)

Table 12-55 FOC_EOMEH (0x40D4)

Bit	7	6	5	4	3	2	1	0
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

Table 12-56 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	Function
[15:0]	FOC_EOME	The speed OMEGA estimated by the estimator The value range is (-32768,32767)

12.2.32 FOC_ESQU (0x40D6,0x40D7)

Table 12-57 FOC_ESQUH (0x40D6)

Bit	7	6	5	4	3	2	1	0
Name	FOC_ESQU[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-58 FOC_ESQUL (0x40D7)

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

Bit	Name	Function
[15:0]	FOC_ESQU	EBETA’s square + EALPHA’s square.The result is 16 bits high, and the highest bit is fixed as 0 The value range is (0,32767)

12.2.33 FOC_POW (0x40D8,0x40D9)

Table 12-59 FOC_POWH (0x40D8)

Bit	7	6	5	4	3	2	1	0
Name	FOC_POW[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-60 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	Function
[15:0]	FOC_POW	Power The value range is (-32768,32767)

12.2.34 FOC_EKP (0x4074,0x4075) shared with BLDC

Table 12-61 FOC_EKPH (0x4074)

Bit	7	6	5	4	3	2	1	0
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

Table 12-62 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	Function
[15:0]	FOC_EKP	KP, the coefficient of PI controller in the estimator The value range is (0,32767), MSB is constantly 0 and Q12 format

12.2.35 FOC_EKI (0x4076,0x4077) shared with BLDC

Table 12-63 FOC_EKIH (0x4076)

Bit	7	6	5	4	3	2	1	0
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

Table 12-64 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	Function
[15:0]	FOC_EKI	The coefficient KI of PI controller in the estimator The value range is (0,32767), MSB is constantly 0 and Q15 format

12.2.36 FOC_EBMFK (0x407C,0x407D) shared with BLDC

Table 12-65 FOC_EBMFKH (0x407C)

Bit	7	6	5	4	3	2	1	0
Name	FOC_EBMFK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-66 FOC_EBMFKL (0x407D)

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

Bit	Name	Function
[15:0]	FOC_EBMFK	The coefficients EKLPF of lowpass filter , which is calculated in the estimator The value range is (-32768,32767) and Q15 format

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

12.2.37 The FOC_KSLIDE (0x4078,0x4079) shared with BLDC

Table 12-67 FOC_KSLIDEH (0x4078)

Bit	7	6	5	4	3	2	1	0
Name	FOC_KSLIDE/FOC_PLLKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-68 FOC_KSLIDEL (0x4079)

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

Bit	Name	Function
[15:0]	FOC_KSLIDE/ FOC_PLLKP	When ESEL=0 (sliding mode) of FOC_CR1, it is the coefficient KSLIDE in the estimator and it’s Q15 format When ESEL=1 (PLL mode) of FOC_CR1, it is the coefficient KP of the PI controller of PLL and it’s Q12 format The value range is (0,32767), MSB is constantly 0

12.2.38 FOC_EKLPFMIN (0x407A,0x407B) shared with BLDC

Table 12-69 FOC_EKLPFMINH (0x407A)

Bit	7	6	5	4	3	2	1	0
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

Table 12-70 FOC_EKLPFMINL (0x407B)

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

Bit	Name	Function
[15:0]	FOC_EKLPFMIN / FOC_PLLKI	When ESEL=0 (sliding mode) of FOC_CR1, is the minimum value of the lowpass filter coefficient of the back electromotive force in the estimator; when the lowpass filter coefficient calculated by the estimator is less than the minimum value, the coefficient is equal to the minimum value and it’s Q15 format When ESEL=1 (PLL mode) of FOC_CR1, it is the coefficient KI of the PI controller of PLL and it’s Q15 format The value range is (0,32767), MSB constant is 0

12.2.39 FOC_OMEKLPF (0x407E,0x407F)

Table 12-71 FOC_OMEKLPFH (0x407E)

Bit	7	6	5	4	3	2	1	0
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

Table 12-72 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	Function
[15:0]	FOC_OMEKLPF	Lowpass filter’s coefficients calculated throught the speed in the estimator. The value range is (0,32767), MSB is constantly 0 and Q15 format

12.2.40 FOC_FBASE (0x4080,0x4081)

Table 12-73 FOC_FBASEH (0x4080)

Bit	7	6	5	4	3	2	1	0
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

Table 12-74 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	Function
[15:0]	FOC_FBASE	The coefficient of the DELTA THETA calculated through the speed OMEGA in the estimator
FBASE = fbase *△T * 32768 Assuming fbase = 200Hz, △T = 62.5us, fbase = 409		

12.2.41 FOC_EFREQACC (0x4082,0x4083) shared with BLDC

Table 12-75 FOC_EFREQACCH (0x4082)

Bit	7	6	5	4	3	2	1	0
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

Table 12-76 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	Function
[15:0]	FOC_EFREQACC	The increment OMEGA of the estimator in force angle mode. The value range is (0,65535) Note: the internal FOC_EFREQACC is 24bit, the highest bit is sign bit, and the lower 16 bits are configurable
Assuming that fbase = 200Hz and polar logarithm pp=4, speed_base=60*fbase/pp=3000rpm, set OMEGA increment to 3rpm Then FOC_EFREQACC = 3/speed_base*32768*256 = 8388(0x20c4)		

12.2.42 FOC_EFREQMIN (0x4084,0x4085) shared with BLDC

Table 12-77 FOC_EFREQMINH (0x4084)

Bit	7	6	5	4	3	2	1	0
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

Table 12-78 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	Function
[15:0]	FOC_EFREQMIN	OMEGA minimum value: When the estimator forced angle mode is enabled and the estimated OMEGA is less than this value, the force angle mode takes effect. The value range is (-32768,32767). Note: FOC_EFREQMIN is internally 24bit, the highest bit is the sign bit, and the upper 16 bits are configurable.
Assuming that fbase = 200Hz and polar logarithm pp=4, speed_base=60*fbase/pp=3000rpm, set the minimum OMEGA value to 30rpm Then foc_efreqmin = 30/speed_base*32768 = 327(0x147)		

12.2.43 FOC_EFREQHOLD (0x4086,0x4087) Shared with BLDC

Table 12-79 FOC_EFREQHOLDH (0x4086)

Bit	7	6	5	4	3	2	1	0
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

Table 12-80 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	Function
[15:0]	FOC_EFREQHOLD	OMEGA retention value: when OMEGA increases to this value, it remains at this value The value range is (-32768,32768) Note: the internal FOC_ EFREQHOLD is 24-bit, the highest bit is sign bit, and the higher bit is 16 bits
Assuming that fbase=200Hz and polar logarithm pp=4, speed_base=60*fbase/pp=3000rpm, set the		

OMEGA retention value to 60rpm
Then FOC_EFREQHOLD = 60/speed_base*32768 = 655(0x28f)

12.2.44 FOC_EK3 (0x4088,0x4089)

Table 12-81 FOC_EK3H (0x4088)

Bit	7	6	5	4	3	2	1	0
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

Table 12-82 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	Function
[15:0]	FOC_EK3	The third coefficient for the estimator to estimate the current The value range is (0,32767), MSB is constantly 0. It’s Q15 format

12.2.45 FOC_EK4 (0x408A,0x408B)

Table 12-83 FOC_EK4H (0x408A)

Bit	7	6	5	4	3	2	1	0
Name	FOC_EK4[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-84 FOC_EK4L (0x408B)

Bit	7	6	5	4	3	2	1	0
Name	FOC_EK4[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	FOC_EK4	The fourth coefficient for the estimator to estimate the current. The value range is (-32768,32767) and it’s Q15 format

12.2.46 FOC_EK1 (0x408C,0x408D)

Table 12-85 FOC_EK1H (0x408C)

Bit	7	6	5	4	3	2	1	0
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

Table 12-86 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	Function
[15:0]	FOC_EK1	The first coefficient for the estimator to estimate the current. The value range is (0,32767), MSB is constantly 0. It’s Q15 format

12.2.47 FOC_EK2 (0x408E,0x408F)

Table 12-87 FOC_EK2H (0x408E)

Bit	7	6	5	4	3	2	1	0
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

Table 12-88 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	Function
[15:0]	FOC_EK2	The second coefficient for the estimator to estimate the current. The value range is (0,32767), MSB is constantly 0. It’s Q15 format

12.2.48 FOC_IDREF (0x4090,0x4091) shared with BLDC

Table 12-89 FOC_IDREFH (0x4090)

Bit	7	6	5	4	3	2	1	0
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

Table 12-90 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	Function
[15:0]	FOC_IDREF	The current ID reference value given by the user The value range is (-32768,32767)

12.2.49 FOC_IQREF (0x4092,0x4093) shared with BLDC

Table 12-91 FOC_IQREFH (0x4092)

Bit	7	6	5	4	3	2	1	0
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

Table 12-92 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	Function
[15:0]	FOC_IQREF	The reference value of current IQ given by the user The value range is (-32768,32767)

12.2.50 FOC_DQKP (0x4094,0x4095) shared with BLDC

Table 12-93 FOC_DQKPH (0x4094)

Bit	7	6	5	4	3	2	1	0
Name	FOC_DQKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-94 FOC_DQKPL (0x4095)

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

Bit	Name	Function
[15:0]	FOC_DQKP	KP, the coefficient of PI controller on DQ axis The value range is (0,32767), MSB is constantly 0. It’s Q12 format

12.2.51 FOC_DQKI (0x4096,0x4097) shared with BLDC

Table 12-95 FOC_DQKIH (0x4096)

Bit	7	6	5	4	3	2	1	0
Name	FOC_DQKI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 12-96 FOC_DQKIL (0x4097)

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

Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	FOC_DQKI	KI, the coefficient of PI controller of DQ axis The value range is (0,32767), MSB is constantly 0. It’s Q15 format

12.2.52 FOC_UDCFLT (0x4098,0x4099)

Table 12-97 FOC_UDCFLTH (0x4098)

Bit	7	6	5	4	3	2	1	0
Name	FOC_UDCFLT[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 12-98 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	Function
[15:0]	FOC_UDCFLT	Filtered bus voltage FOC module samples bus voltage and filter for the use of software, Fixed ADC channel 2 The value range is (0,32767)
Assuming that the bus voltage 1/6 partial voltage is fed into ADC, the voltage range of ADC is 0~5V, that is, the effective range of the bus voltage is 0~30V, if FOC_UDCFLT is 19661 (0x4CCD), the bus voltage = 19661/32768*5*6 = 18V		

13 TIM1

13.1 Timer1 operation

Timer1 contains a 16-bit up-counting basic timer and a 16-bit up-counting reload timer, both of which use internal clock counting sources.Timer1 is mainly used for BLDC square wave automatic control or HALL signal processing.Timer1 has the following features:

1. 16-bit up-counting basic timer, which is used to record the interval between the basic timer's two position detection or writing timings, that is, for counting the 60-degree commutation time;
2. 16-bit reload timer for up counting, used for timing: the interval between the position detection and the reload timer overflow, which is used to count the freewheeling mask time and zcp to commutation time;
3. 3-bit programmable frequency divider, which can divide the count clock of two timers
4. Input filtering and sampling;
5. Position detection module, which generates a position detection signal according to the input signal;
6. Write the timing module to update the output status register;
7. 7 groups of status registers control comparators and outputs;
8. Interrupt event generation
   a) Basic timer overflow interrupt
   b) Reload timer overflow interrupt
   c) Writing timing interrupt
   d) Position detection interrupt
   e) Shield freewheeling end interrupt

The internal structure of Timer1 is shown in Figure 13-1.

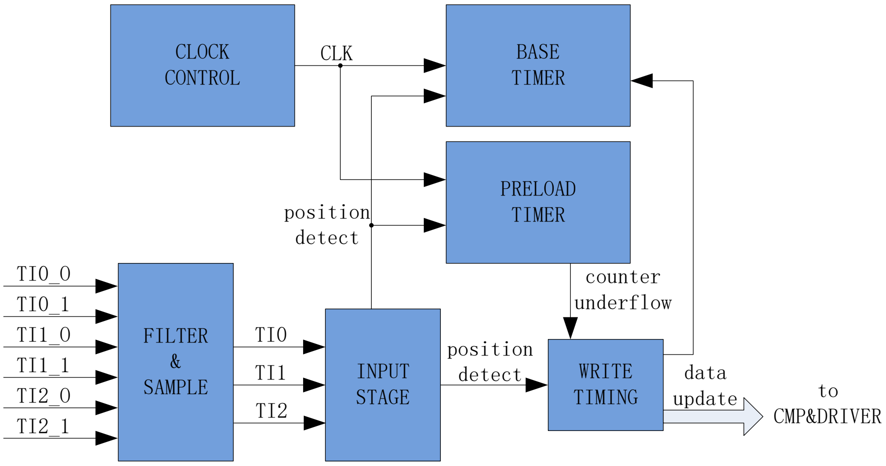


Figure 13-1 Internal structure of Timer1

13.1.1 Timer counter unit

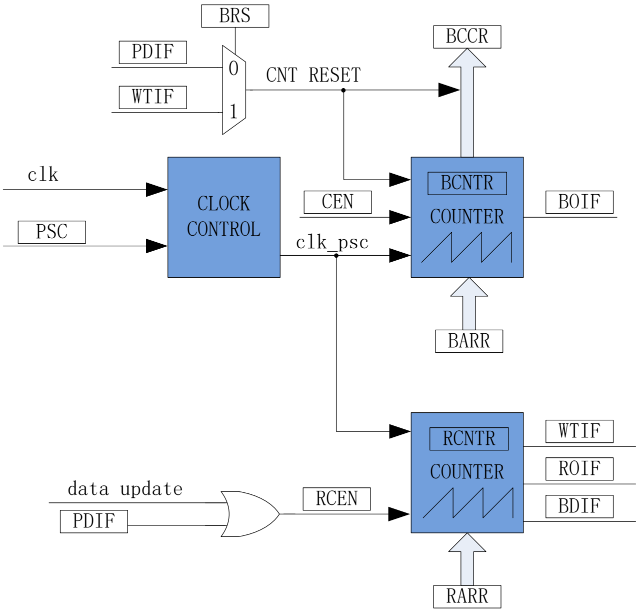


Figure 13-2 The time base unit

Timer1 contains a frequency divider, a 16-bit up counting basic timer, and a 16-bit up counting reload timer.

13.1.1.1 Timer clock controller

The Timer clock controller is used to generate count clock source for basic timer and the reload timer. The counting clock is divided by the predivider.The predivider is based on a 12-bit counter controlled by a 3-bit register PSC. 8 divider coefficients can be selected. The clock source is the internal clock.Since there is no buffer in this control register, the divider coefficient changes immediately, so the divider coefficient should be updated when both the basic timer and the reload timer are not working.

The frequency of the counter can be calculated by the following formula:

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

Suppose the MCU clock is 24MHz(41.67ns)

Table 13-1 The PSC values of the register corresponding to different clock frequency

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

13.1.1.2 Basic timer

The basic timer contains a 16-bit up counter. When the value of TIM1__BCNTR equals to the TIM1__BARR, overflow event generates.The Overflow interrupt flag BOIF of the basic counter is set. At the same time, TIM1__BCNTR is counting rather than reset and restart counting. BRS in TIM1_CR2 chooses the counter reseting source from position detection events, or writting the timing events. When the event reset signal is generated, the current count value TIM1__BCNTR is sent to register TIM1__BCCR, and TIM1__BCNTR value is set to 0, and then the counting is restarted.

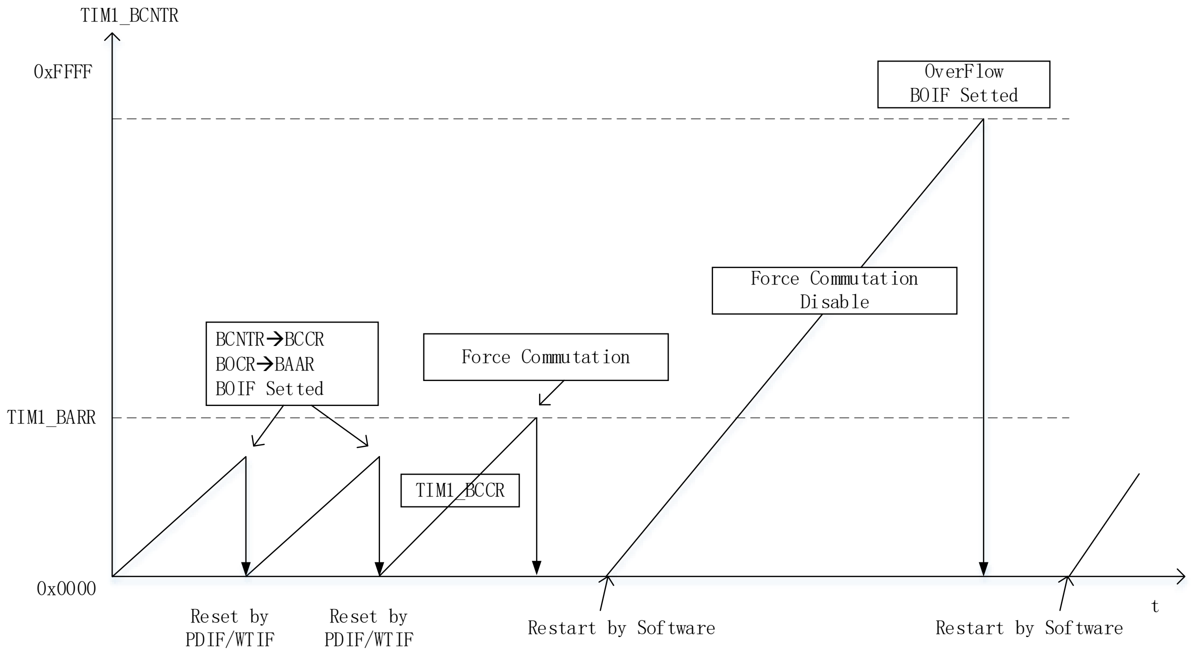


Figure 13-3 Basic timer count waveform

The value of the TIM1_BARR register is immediately applied to the counter, so the register should be updated when the basic timer stops working.Only when the count value TIM1_BCNTR is equal to TIM1_BARR, the overflow event can occur. If TIM1_BCNTR is greater than TIM1_BARR, TIM1_BCNTR accounting number reaches 0xFFFF and then counts from 0. Therefore, it should be noted that TIM1_BCNTR cannot be greater than TIM1_BARR when the register initials.

13.1.1.3 Reload timer

The reload timer contains a 16-bit counting counter. When the count value TIM1_RCNTR is counted to TIM1_RARR, an overflow event occurs. The overflow interrupt flag ROIF on the reload timer is set to 1 and RCEN is set to 0. And the TIM1__RCNTR is cleared, and restart counting after waiting for RCEN to be set.

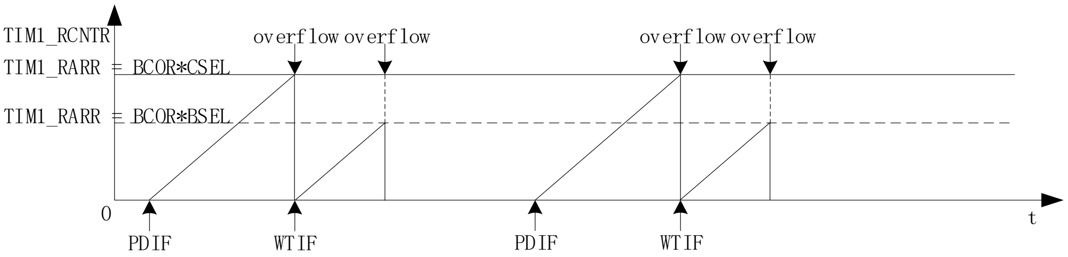


Figure 13-4 Reload timer count waveform

Location detection events and writing timing events can automatically enable T1RCEN by hardware, when reload timer generates overflow event, T1RCEN is cleared by hardware, reload timer stop counting.The reload timer is mainly used to realize BLDC square wave freewheeling mask and delay commutation after detecting the zero point. The rest of time, the reload timer doesn’t work.

13.1.2 Input filtering and sampling

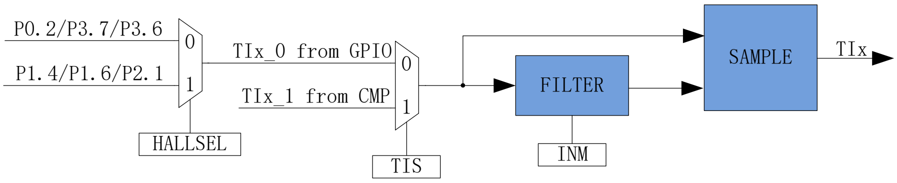


Figure 13-5 Schematic diagram of input signal filtering and sampling

T1TIS of TIM1_CR3 registers selects whether the input source is from comparator or GPIO, in which HALLSEL of CMP_CR1 registers selects whether GPIO comes from (P1.4/P1.6/P2.1) or (P0.2/P3.7/P3.6), INM of TIM1_CR3 registers is input setting to select whether noise filtering is performed, and SAMSEL of CMP_CR3 registers is input setting to select whether sampling is performed.

13.1.2.1 Filtering

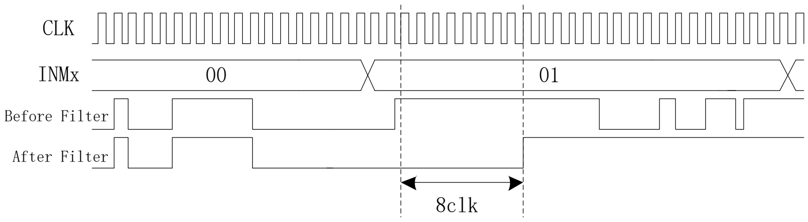


Figure 13-6 Timing diagram of filtering module

According to the TIx of register TIM1_CR2, the filter circuit can choose to filter the input noise with the pulse width of 8/32/64 clock cycle.Enable filtering function, the filtered signal will be about 8~9/32~33/64~65 clock cycles later than the filtered signal.

13.1.2.2 Sampling

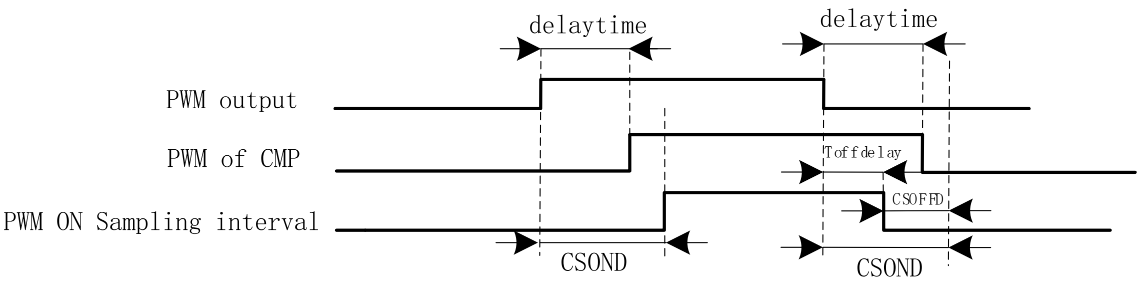


Figure 13-7 PWM ON sampling mode

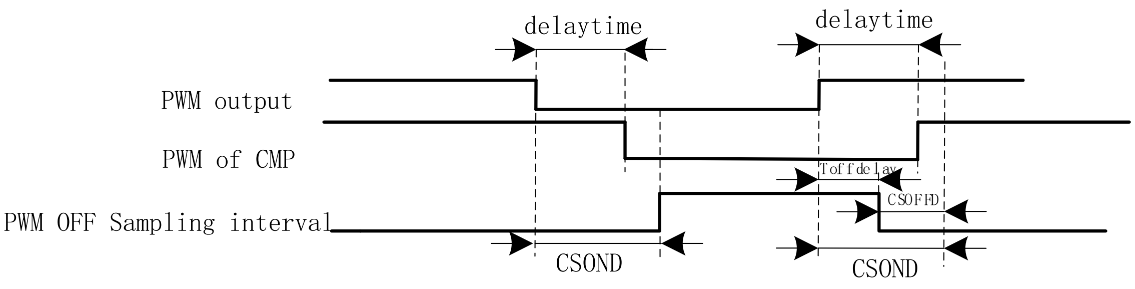


Figure 13-8 PWM OFF sampling mode

In the square wave control mode of BLDC, the input of TI2/TI1/TI0 comes from the comparator. Since the output of the comparator may be interfered by MOS switch of peripheral drive circuit, there is the interference noise of PWM signal.Set the SAMSEL of register CMP_CR3 to select sampling mode, set CSOFFD and CSOND of register CMP_SAMR to adjust sampling interval.

The interference reflected in the output of PWM to the comparator is delayed relative to the jump edge of PWM, which is mainly affected by the following factors: the size of drive resistance, MOS switching speed, input delay and hysteresis setting of comparator.Set CSOFFD mask comparator interference stage, offdelay=CSOND-CSOFFD is to delay closing sampling time of comparator CMP0,CMP1 and CMP2.

Example: if the delay of PWM output to comparator is 2us and the interference width is 1us, it can be set

CSOFFD >1us = 1000ns/41.67ns/8 = 3

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

The method of measuring the delay between PWM output and comparator: set the SAMSEL=00 of register CMP_CR3 to forbid the comparator sampling function, set the CMPSEL of register CMP_CR3 to output corresponding comparator comparison value, enable PWM output and comparator, manually rotate the motor to flip the comparator value, measure the delay between PWM output and comparator output.

Method for measuring the interference width: the above method is used to measure the width of interference level which is outputted by the comparator.

13.1.3 Position detection event

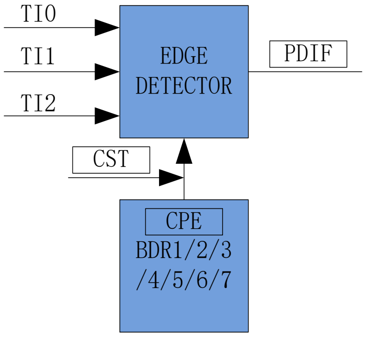


Figure 13-9 Principle block diagram of position detection

The position detection event determines the valid edge of the input based on the CPE of the TIM1_DBR1/2/3/4/5/6/7 register. When the valid edge of the input (TI2/TI1/TI0) arrives, a position detection event occurs. According to the state of the CST of the TIM1_CR4 register, the corresponding CPE of the TIM1_DBR1/2/3/4/5/6/7 register is determined to take effect.

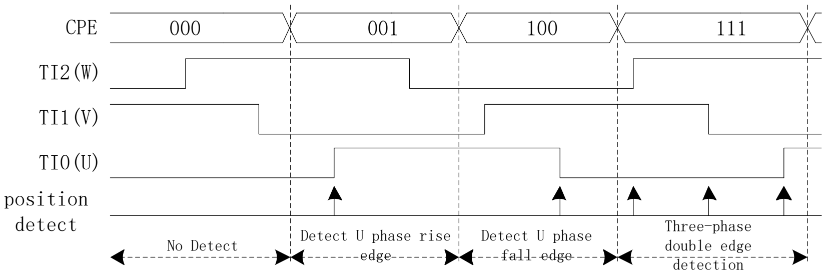


Figure 13-10 Timing diagram of position detection

CPE _DBR1 of TIM1/ 2/3/4/5/6/7 register decided to enter the effective along as follows.

CPE	Description	CPE	Description
000	0	100	Detect the descending edge of U phase and enable the comparator of U phase
001	Detect the rising edge of U phase and enable the comparator of U phase	101	Detect the rising edge of W phase and enable the comparator of W phase
010	Detect the descending edge of W phase and enable the comparator of W phase	110	Detect the descending edge of V phase and enable the comparator of V phase
011	Detect the rising edge of V phase and enable the comparator of V phase	111	Detect three-phase double-edge and enable three-phase corresponding comparator

13.1.4 Writes sequence events

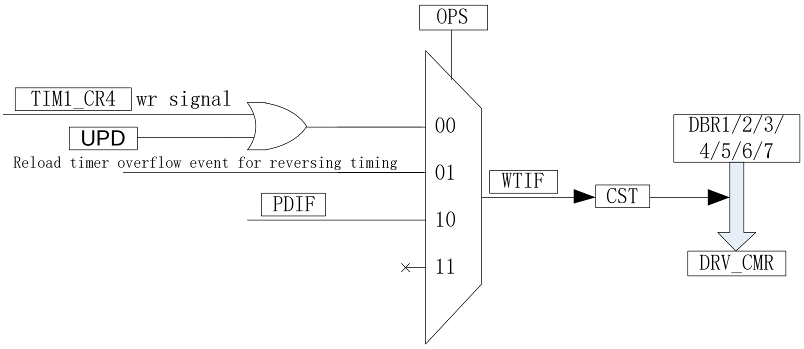


Figure 13-8 Writes a sequence block diagram

The writing timing event is based on the OPS of the TIM1_CR0 register to determine the source of the event. After the writing timing event occurs, the writing timing interrupt flag WTIF is set. If the CST of the TIM1_CR4 register is 001 ~ 110 at the same time, the CST will automatically increase by one, and the corresponding the value of TIM1_DBR1/2/3/4/5/6 register will also be updated to DRV_CMR.

13.1.5 Timer1 interrupt

Timer has 5 interrupt request sources:

1. Overflow interrupt of basic timer
2. Overflow interrupt of reload timer
3. Writing timing interrupt
4. Position detection interrupt
5. Interrupt at end of freewheeling shield

Configuring the interrupt enablement bit of TIM1_IER can enable the corresponding interrupt request

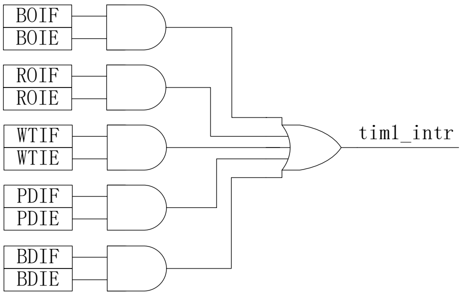


Figure 13-9 TIMER1 interrupt source

13.2 BLDC square wave application

For BLDC square wave application, Timer1 with the comparator and DRIVER module has the following functions:

1. The benchmark time of commutation of 60 degrees can be automatically recorded, and the benchmark time can be filtered.

2. When the position signal cannot be detected, the commutation can be automatically forced
3. Automatically mask freewheeling, that is, during the freewheeling time, the comparator does not work
4. Automatically record the time from the detection position signal to the commutation, and automatically changes commutate.
5. Take over CMP0_SEL of CMP_CR2 register, control comparator 0 automaticly
6. Comparator signal sampling can be set in PWM ON/OFF, and signal filtering can be selected
7. Take over the DRV_CMR register and automatically control the 3-phase 6-channel PWM output

BLDC square wave applications are mostly sensorless applications, and the implementation scheme is back electromotive forcewith a delay of 30 degrees after crossing zero, on which the following discussion is based (T1OPS=01).

13.2.1 Six steps of BLDC comutation

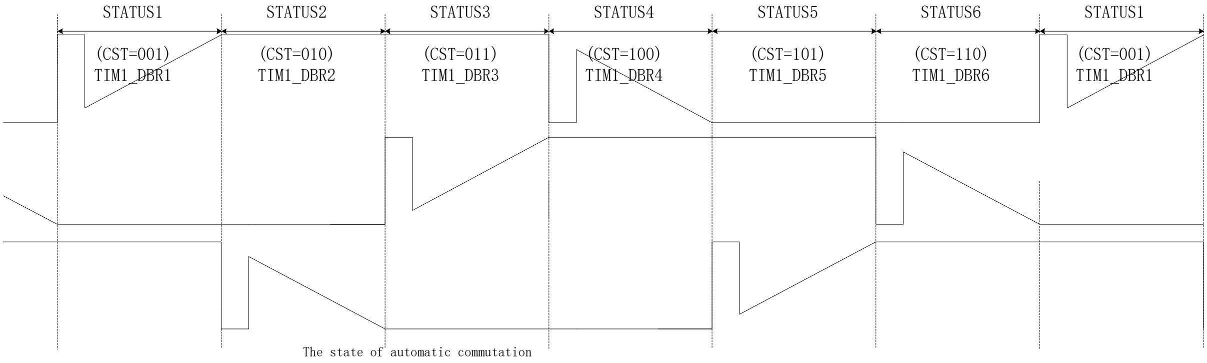


Figure 13-10 Six-step phase transition diagram of BLDC

The CST of register TIM1_CR4 is the commutation state machine, where state 0 is used for output shutdown; state 7 is used for customization, such as brake, pre-charging, pre-positioning, starting and other functions; state 1~6 is used for six steps of automatic commutation, after commutation, the state will automatically add one.

State 1~7 corresponds to register TIM1_DBR1~7. When the writing timing event occurs, TIM1_DBRx corresponding to the current state will be automatically updated to register DRV_CMR and the CMP0_SEL of register CMP_CR2.

13.2.2 BLDC working principle

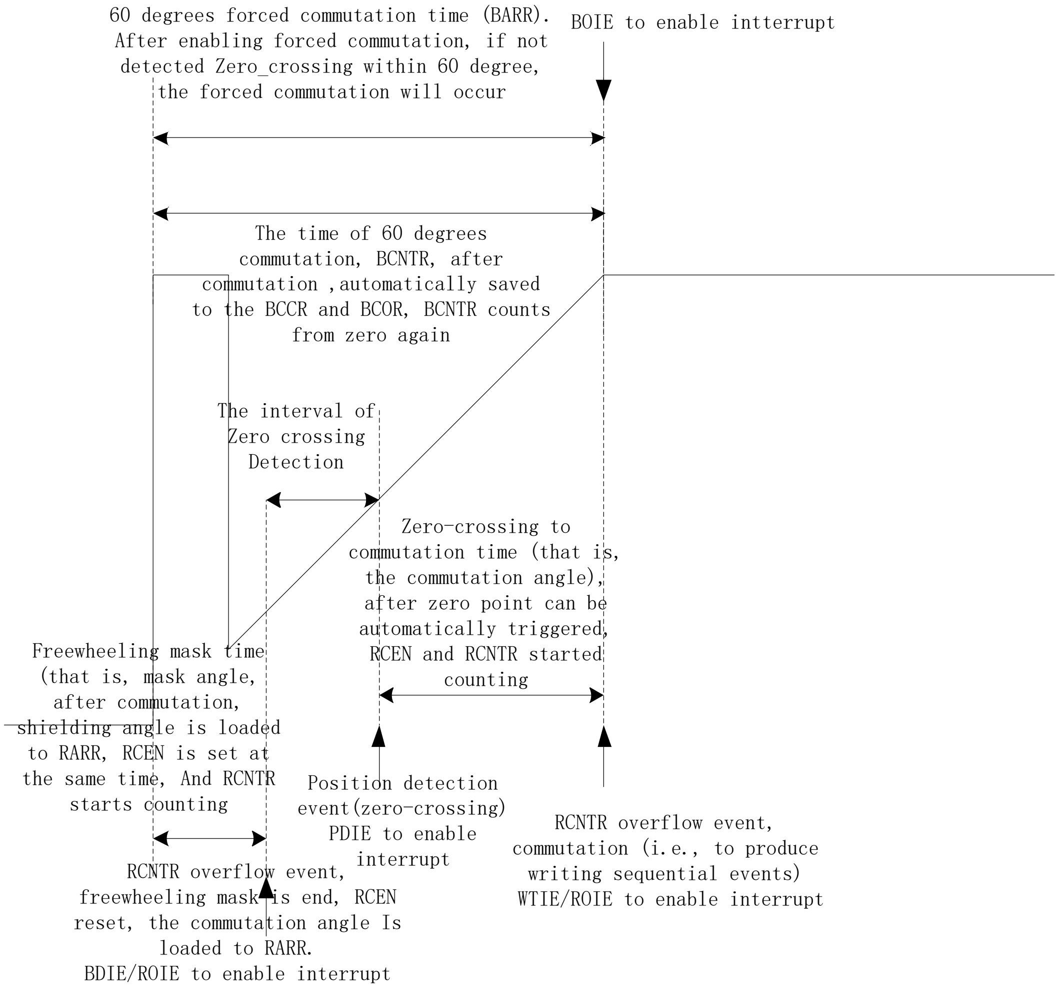

60 degrees forced commutation time (BARR). After enabling forced commutation, if not detected Zero_crossing within 60 degree, the forced commutation will occur
BOIE to enable intterrupt
The time of 60 degrees commutation, BCNTR, after commutation ,automatically saved to the BCCR and BCOR, BCNTR counts from zero again
The interval of Zero crossing Detection
Zero-crossing to commutation time (that is, the commutation angle), after zero point can be automatically triggered, RCEN and RCNTR started counting
Freewheeling mask time (that is, mask angle, after commutation, shielding angle is loaded to RARR, RCEN is set at the same time, And RCNTR starts counting
Position detection event(zero-crossing) PDIE to enable interrupt
RCNTR overflow event, commutation (i.e., to produce writing sequential events) WTIE/ROIE to enable interrupt
RCNTR overflow event, freewheeling mask is end, RCEN reset, the commutation angle Is loaded to RARR. BDIE/ROIE to enable interrupt

Figure 13-11 Working principle of BLDC

13.2.2.1 60 degrees benchmark

The TIM1_BCCR register is 60 degrees of the last time, configure BRS=0 of TIM1_CR2 register to set the time between two commutations as 60 degrees, and configure BRS=1 to set the time between two zero points detected as 60 degrees.

The TIM1_BCOR register is the filtered 60-degree, namely 60-degree benchmark. Setting the CFLT of the TIM1_CR0 register can select the first 1/2/4/8 TIM1_BCCR to average them to get TIM1_BCOR.

BLDC calculates the freewheeling shielding angle, the angle betweetn zero-crossing (zcp) and the commutation and the forced commutation angle through the 60-degree benchmark TIM1_BCOR.

13.2.2.2 Commutation

Commutation, that is writing timing events. T1OPS of TIM1_CR1 register is used to choose phase change mode, in which T1OPS=00 is mostly used for sensorless startup, T1OPS=01 is mostly used for automatic commutation without sensors, and T1OPS=10 is mostly used for sensortive commutation.

After commutation, TIM1 will automatically perform the following operations:

- Save the current TIM1_BCNTR to TIM1_BCCR, and TIM1_BCCR is filtered and saved to TIM1_BCOR as the 60-degree reference value
- TIM1_BCNTR counts from 0 again
- Start the freewheeling mask, load the mask angle into TIM1_RARR and set the RCEN as 1. TIM1_RCNTR begins to count
- If the CST in register TIM1_CR4 is in state 1-6, the CST will automatically switch to the next state
- Can produce write sequence interrupt T1WTIF and reload timer overflow interrupt T1ROIF

13.2.2.3 Forced commutation at 60 degrees

When the motor is rotating smoothly, the zero-crossing point is generally detected at about 30 degrees after commutation. If no zero crossing is detected within 60 degrees after commutation, a forced commutation is usually required. Set FORC = 1 in the TIM1_CR0 register to enable the 60-degree forced commutation function. When no zero crossing is detected within 60 degrees after commutation, the hardware commutates forcibly and sets the basic timer overflow interrupt flag BOIF at the same time (Note: FORC = 1, detected within 60 degrees after commutation and TIM1__BCNTR>TIM1__BARR, BOIF will not be set). Set FORC = 0 in the TIM1_CR0 register, when TIM1__BCNTR>TIM1__BARR and the interrupt flag BOIF is set, basic timer continues to count rather than reset. The software can determine the basic timer overflow interrupt flag BOIF and the position detection interrupt flag PDIF to perform manual commutation.

13.2.2.4 Freewheel mask

After commutation, the original conduction phase becomes a floating phase. At this time, the electrical energy in the inductance of this phase needs to be released to the power supply or ground through a freewheeling diode. During the freewheeling process, the comparator will be affected. Therefore, it is necessary to mask the trigger edges that are generated by the comparators during the freewheeling time, so as to prevent the wrong signal generated by the freewheeling from causing wrong commutation. Freewheeling mask ends, generating masking Freewheeling end interrupt flag BDIF.

In freewheeling mask time, TIM1 keeps last latched comparator level value, when the end of the freewheeling mask, TIM1 acquisition level value of the comparator, it's important to note that when the freewheeling mask time less than the freewheeling time, the comparator level at the end of the freewheeling mask is equal to the comparator level after zero-crossing, false passing zero trigger edge will occur, so you need to adjust freewheeling mask time according to the characteristics of the motor, make freewheeling mask time greater than the freewheeling time.

Freewheel mask time is set by BSEL in the TIM1_CR1 register. The conversion formula is:

Mask angle = BSEL/128*60.

13.2.2.5 The angle between zcp and commutation (delayed commutation)

The interval from the end of freewheeling mask to the detection of zero crossing is the zero crossing detection interval. If the zero crossing is not detected after the freewheeling mask, then the interval from the end of freewheeling mask to the commutation is the zero crossing detection interval. The position detection event is an edge trigger, and only the first valid trigger edge is detected. When the effective triggering edge is detected in the zero-crossing detection interval, the zero-crossing detection interval ends and the subsequent triggering edge are all invalid, so the user needs to adjust the parameters of filtering and sampling to ensure that the first triggering edge is the real zero-crossing.

zcp(that is, position detection event) is detected in the zero-crossing detection interval. The hardware starts TIM1_RCNTR for timing according to the time set by the software from ZCP to phase change. When the timing is finished, the hardware will automatically commutate, and WTIF will be generated as the writing sequence interrupt flag.

The time from ZCP to commutation is set by CSEL in register TIM1_CR2, the formula:

Commutation angle =CSEL/128*60

13.2.2.6 Cycle-by-cycle current limit

Refer to wave-by-wave current limiting 26.1.1.2

13.2.3 Debugging of BLDC is related

The chip provides the following debugging methods

1. GP07 displays the comparator signal in real time
2. GP01 displays the status of TIM1 in real time
3. SPI debugger is used to display the TIM1 related registers on the oscilloscope

13.2.3.1 Comparator debugging

Setting CMP_CR3 register CMPSEL can output the results of BLDC related CMP0/1/2_OUT through GP07, where CMP0/1/2_OUT is the result of comparator after filtered sampling.

Configure DBGSEL=11B of register CMP_CR3 to output the comparator sampling interval through GP01. Please refer to the charpter about sampling for waveform. SAMSEL of register CMP_CR3 is set to select the sampling interval of comparator, which corresponds to GP01 output as shown in the table below

SAMSEL	The sampling interval	GP01 display
00	Sample on and off without delay	The constant high level
01	Only sampling on off, configure the delay of sampling according to CMP_SAMR	PWM off sampling interval
10	Only sampling on on, configure the delay of sampling according to CMP_SAMR	PWM on sampling interval
11	Sampling on on and off, configure the delay of sampling according to CMP_SAMR	PWM off+on sampling interval

The comparator sampling interval, one of the signals of CMP0/1/2_OUT and one signal of the corresponding UVW output, is displayed by oscilloscope, the CMP_SAMR register is adjusted, and the comparator sampling interval is placed in the PWM interval corresponding to UVW (the filter delay of the comparator needs to be taken into account) to observe whether CMP0/1/2_OUT meets the requirements.

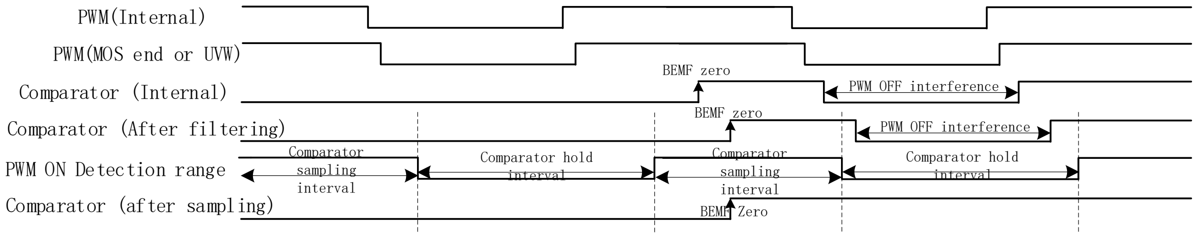


Figure 13-12 comparator debugging

13.2.3.2 Debug of freewheeling mask and commutation

Both the freewheeling mask interval and the delayed commutation interval use the reload timer RCNTR to count, so the waveform of RCNTR can be displayed on the oscilloscope through SPI debugger.

GP01 was used to display the real-time status of TIM1, the commutation point and the freewheeling mask interval, the interval of waiting for zero-crossing and delayed commutation interval can be obtained by cooperating with UVW phase to obtain

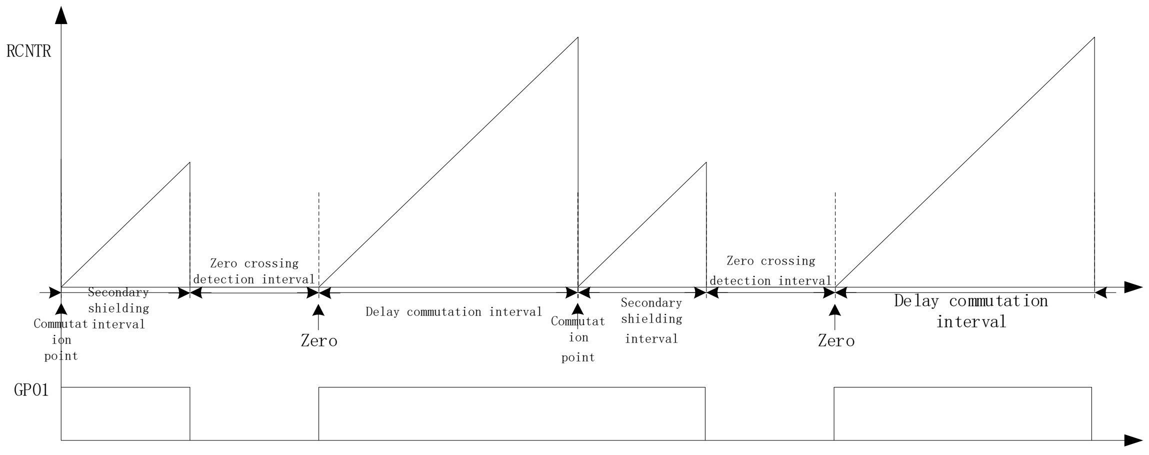

RCNTR
Secondary shielding interval
Zero crossing detection interval
Delay commutation interval
Secondary shielding interval
Zero crossing detection interval
Delay commutation interval
Commutat ion point
Zero
Commutat ion point
Zero
GP01

Figure 13-13 Waveform of freewheeling mask and 30 ° phase delay RCNTR/GP01

13.3 Timer1 register

13.3.1 TIM1_CR0 (0x4068)

Table 13-2 TIM1_CR0 (0x4068)

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

Bit	Name	Function
[7]	T1RWEN	When operating TIM1_CR0, RWEN must write one before T1RCEN can write. This bit only can read 0.
[6:5]	T1CFLT	Commutation filter selection The average time of X commutations was used as the benchmark of 60 degrees 00: The time of 1 commutation 01: The time of 2 commutations 10: The time of 4 commutations 11: The time of 8 commutations
[4]	T1FORC	Enable Forced automatic commutation at 60 degrees When no zero is detected within 60 degrees after commutation, hardware commutates forcibly.When a zero-crossing is detected within 60 degrees, the hardware does not commutate forcibly even if TIM1_BCNTR exceeds TIM1_BARR. Note: if T1FORC=0, even if TIM1_BCNTR exceeds TIM1_BARR, TIM1_BCNTR will continue to count rather than starting from 0, and the hardware will not commutates forcibly 0: Disable 1: Enable

[3:2]	T1OPS	Data transmission mode selection These bits are used to select the transfer mode in which the TIM1_DBRx register writes to the DRV_CMR register, that is, writing timing events/ commutations 00: the software writes 1 to UPD or writes TIM1_CR4 to trigger data transmission (mainly used for sensorless square wave startup) 01:16-bit reload timer is used for triggering data transmission at the overflow event of commutation time timing (mainly used for sensorless square wave startup) 10: position detection input triggers data transmission (mainly used for inductive square wave) 11: Reserved
[1]	T1BCEN	Enable basic timer counter. 0: Disable 1: Enable
[0]	T1RCEN	Enable reload timer counter The software must write 1 to RWEN to operate T1RCEN. The location detection time and writing sequence events of the hardware can automatically enable T1RCEN. When the reload timer produces an overflow event, T1RCEN is cleared by hardware. 0: Disable 1: Enable

13.3.2 TIM1_CR1 (0x4069)

Table 13-3 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	Function
[7]	T1BAPE	TIM1_BARR automatically loads enable After enabled, when the base timer is reset because of a position detection event or when a writing timing event, the 60-degree reference value is stored in the TIM1_BARR register.(used for forced 60-degree commutatioin when zcp cannot be detected) 0: Disable 1: Enable
[6:0]	BSEL	Freewheeling mask angle selection The angle (time) of the freewheeling mask after the commutation, during which the input edge is not detected The formula: freewheeling mask angle =BSEL/128*60 Note: if the mask angle is 0, BSEL shall be set to 1

13.3.3 TIM1_CR2 (0x406A)

Table 13-4 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	Function
[7]	T1BRS	Basic timer reset source selection 0: writing timing reset and 60 degree forced automatic phase reversal reset (common setting) 1: position detection and reset
[6:0]	CSEL	Phase change Angle selection After the input of position detection is triggered, the phase change corresponding to CSEL is delayed Formula: phase change Angle =CSEL/128*60 Note: if the phase change Angle is 0, CSEL shall be set to 1

13.3.4 TIM1_CR3 (0x406B)

Table 13-5 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	R/W
Reset	0	0	0	0	0	1	0	0

Bit	Name	Function
[7]	RSV	Reserved
[6:4]	T1PSC	Timer clock frequency division selection These bits are used to perform N frequency division on the MCU clock as the counting clock of the basic timer and reload timer. It is assumed that the MCU clock is 24MHz (41.67ns). 000:0x1 (24MHz) 001:0x2 (12MHz) 010:0x4 (6MHz) 011:0x8 (3MHz) 100:0x10 (1.5MHz) 101:0x20 (750kHz) 110:0x40 (375kHz) 111:0x80 (187.5kHz)
[3:2]	T1TIS	Select the input source (TI0/TI1/TI2) TIM1 filters, samples and generates position detection time for the selected input source.These bits affect CMP0/1/2_OUT and CMP0/1/2_IF of the comparator module CMP_SR 00: GPIO is taken as input, where the results of CMP_SR are generated through GPIO according to the selection of CMP_CR1[7] (P1.4/P1.6/P2.1) or (P0.2/P3.7/P3.6)

		01: the output of the comparator (CMP0/CMP1/CMP2) is taken as input, and the result of CMP_SR is generated through CMP 1x: reserved bits
[1:0]	T1INM	Input TI0/TI1/TI2 noise pulse width selection, when the noise pulse width is less than the set value, the noise will be filtered. Suppose the MCU clock is 24MHz(41.67ns) 00: no filtering 01:8 clock cycles, 8 x 41.67ns 10:32 clock cycles, 32 x 41.67ns 11:6 4 clock cycles, 64 x 41.67ns

13.3.5 TIM1_CR4 (0x406C)

Table 13-6 TIM1_CR4 (0x406C)

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

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>[7:3]</td><td>RSV</td><td>Reserved</td></tr>
<tr><td>[2:0]</td><td>T1CST</td><td>Commutation state machine
State machine in different states will correspond to different CPE and CMR (TIM1_DBRx);
When CST is in 001~111 state, timer1 will automatically take over the enablement CMP0/1/2 and decide the switch of comparator according to the CPE of corresponding state
When CST is in 001~110 state, it will automatically increase by1 circularly when the writing sequence event is triggered
<table>
<tr><td>CST</td><td>TIM1_DBRx</td><td>CST</td><td>TIM1_DBRx</td></tr>
<tr><td>000</td><td>0</td><td>100</td><td>TIM1_DBR4</td></tr>
<tr><td>001</td><td>TIM1_DBR1</td><td>101</td><td>TIM1_DBR5</td></tr>
<tr><td>010</td><td>TIM1_DBR2</td><td>110</td><td>TIM1_DBR6</td></tr>
<tr><td>011</td><td>TIM1_DBR3</td><td>111</td><td>TIM1_DBR7</td></tr>
</table>
</td></tr>
</table>

13.3.6 TIM1_IER (0x406D)

Table 13-7 TIM1_IER (0x406D)

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

Bit	Name	Function
[7]	T1UPD	When OPS=00, the software writes 1 to UPD to trigger data transfer. This bit can write only and is unreadable, and the hardware

		automatically clears it after writing 1 to it
[6]	RSV	Reserved
[5]	RSV	Reserved
[4]	T1BOIE	Enable overflow interrupt of basic timer 0: Disable overflow interrupt 1: Enable overflow interrupt
[3]	T1ROIE	Enable overflow interrupt of reload timer 0: Disable overflow interrupt of reload timer 1: Enable overflow interrupt of reload timer
[2]	T1WTIE	Enable write sequence interrupt 0: Disable writing timing interrupt 1: Enable writing timing interrupt
[1]	T1PDIE	Enable position detection interrupt 0: Disable position detection interrupt 1: Enable position detection interrupt
[0]	T1BDIE	Freewheeling mask end interrupt enable 0: Disable comparison interrupts 1: Enable to comparison interrupts

13.3.7 TIM1_SR (0x406E)

Table 13-8 TIM1_SR (0x406E)

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

Bit	Name	Function
[7:6]	RSV	Reserved
[5]	RSV	Reserved
[4]	T1BOIF	Overflow interrupt flag of basic timer When the basic timer counts up, an overflow event occurs when the value of the TIM1_BCNTR register matches the value of the TIM1_BARR register.If TIM1_CR1[T1FORC]=1, TIM1_BCNTR is cleared, otherwise, TIM1_BCNTR continues to count.This bit is set by hardware, it is cleared by software. Note: to clear TIM1_BCNTR in the interrupt, write UPD or TIM1_CR4 when TIM1_CR2[T1BRS]=0 0: no event occurred; 1: overflow event occured.
[3]	T1ROIF	Overflow interrupt flag of reload timer When the reload timer counts up and the value of the TIM1_RCNTR register matches the value of the TIM1_RARR register, an overflow event occurs and the value of TIM1_RCNTR is cleared. The bit is set by the hardware and it is cleared by the software. 0: no event occurred;

		1: overflow event occured.
[2]	T1WTIF	Writing timing interrupt flag When the TIM1_DBRH/TIM1_DBRL register is transferred to the TIM1_DRH/TIM1_DRL register, the bit is set by hardware and is cleared by software. Note: when OPS=00, the software writes 1 to WTIF, which will generate a write event. 0: no event occurred; 1: write timing occurred.
[1]	T1PDIF	Position detection interrupt flag Position detection event interrupt occurrs when the input (TI2,TI1,TI0) matches the TIM1_DBRx[CPE] corresponding to the current state TIM1_CR4[CST].The bit is set by hardware.It clears by software. 0: no event occurred; 1: location detection event occurred.
[0]	T1BDIF	Freewheeling mask end interrupt flag After commutation, freewheeling mask starts. When the mask ends, the bit is set by hardware and is cleared by software. 0: no event occurred; 1: the event occurred

13.3.8 TIM1_BCOR (0x4070,0x4071)

Table 13-9 TIM1_BCORH (0x4070)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCORH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 13-10 TIM1_BCORL (0x4071)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCORL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function

[15:0]	TIM1_BCOR	Capture the basic timer count after filter
		TIM1_BCCR filtered value, that is, the 60-degree benchmark value
		Note: when the user initializes the 60-degree benchmark value, TIM1_BCCR and TIM1_BCOR should be initialized simultaneously.When configuring TIM1_BCCR, the 60 degree benchmark value can be directly written into TIM1_BCCR.When configuring TIM1_BCOR, please follow these steps:
		T1CFLT=00,60 degree reference value;
		T1CFLT=01,60 degree reference value /2;
		T1CFLT=10,60 degree reference value /4;

		T1CFLT=11,60 degree reference value /8.

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

TIM1_DBRx(x=1~7) corresponds to the data when CST=1/2/3/4/5/6, respectively. Take TIM1_DBR1 as an example of TIM1_DBRx register.

Table 13-11 TIM1_DBR1H (0x4074)

Bit	7	6	5	4	3	2	1	0
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
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

Table 13-12TIM1_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

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>[15]</td><td>RSV</td><td>Reserved</td></tr>
<tr><td>[14:12]</td><td>T1CPE</td><td>Select TI0/TI1/TI2 input edge polarity and comparator enablement
These bits are used to select the polarity of the input edge used for position detection and the enablement of the corresponding comparator. Position detection is triggered according to the polarity of the input edge set by these bits.
<table>
<tr><th>CPE</th><th>Description</th><th>CPE</th><th>Description</th></tr>
<tr><td>000</td><td>0</td><td>100</td><td>Detect U phase descent edge,
enable U phase corresponds to the comparator</td></tr>
<tr><td>001</td><td>Detect the U phase rise edge,
enable U phase corresponds to the comparator</td><td>101</td><td>Detect W phase rising edge,
enable W phase corresponds to the comparator</td></tr>
<tr><td>010</td><td>Detect the W phase descent edge,
enable W phase corresponds to the comparator</td><td>110</td><td>Detect the V phase descent edge,
enable V phase corresponds to the comparator</td></tr>
<tr><td>011</td><td>Detect V phase rise edge,
enable V phase corresponds to the comparator</td><td>111</td><td>Detect three-phase double-edge,
enable three-phase comparators</td></tr>
</table></td></tr>
<tr><td>[11]</td><td>T1WHP</td><td>Enable W phase upper bridge output
0: high level effective
1: low level effective</td></tr>
<tr><td>[10]</td><td>T1WLP</td><td>W phase lower bridge output polarity
0: high level effective
1: low level effective</td></tr>
<tr><td>[9]</td><td>T1VHP</td><td>Enable V phase upper bridge output</td></tr>
</table>

		0: high level effective 1: low level effective
[8]	T1VLP	V phase lower bridge output polarity 0: high level effective 1: low level effective
[7]	T1UHP	Enable U phase upper bridge output 0: high level effective 1: low level effective
[6]	T1ULP	U phase lower bridge output polarity 0: high level effective 1: low level effective
[5]	T1WHE	Enable W phase upper bridge output 0: off -- disable output 1: on -- enable output Note: when WLE and WHE are both 1, U phase upper and lower bridges complement the output, and the output is automatically inserted into the dead zone.
[4]	T1WLE	Enable W phase lower bridge output 0: off -- disable output 1: on -- enable output Note: when both WLE and WHE are 1, the upper and lower bridges of the W phase complement the output, and the output is automatically inserted into the dead zone.
[3]	T1VHE	Enable V phase upper bridge output 0: off -- disable output 1: on -- enable output Note: when both VLE and VHE are 1, the upper and lower bridges of V phase complement each other and the output is automatically inserted into the dead zone.
[2]	T1VLE	Enable V phase lower bridge output 0: close -- disable output 1: enable -- enable output Note: when both VLE and VHE are 1, the upper and lower bridges of V phase complement each other and the output is automatically inserted into the dead zone.
[1]	T1UHE	Enable U phase upper bridge output 0: close -- disable output 1: enable -- enable output Note: when ULE and UHE are both 1, the upper and lower bridges of the U phase complement each other and the output is automatically inserted into the dead zone.
[0]	T1ULE	Enable U phase lower bridge output 0: close -- disable output 1: enable -- enable output Note: when ULE and UHE are both 1, the upper and lower bridges of the U phase complement each other and the output is automatically inserted into the dead zone.

13.3.10 TIM1_BCNTR (0x4082,0x4083)

Table 13-13 TIM1_BCNTRH (0x4082)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCNTRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 13-14 TIM1_BCNTRL (0x4083)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCNTRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	TIM1_BCNTR	The counter value of the basic timer used for counting the commutation time of 60 degrees. Note: TIM1_BCNTR only selects the reset source according to TIM1_CR2[T1BRS]. The overflow of TIM1_BCNTR does not cause a recount of TIM1_BCNTR.

13.3.11 TIM1_BCCR(0x4084,0x4085)

Table 13-15 TIM1_BCCRH(0x4084)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCCRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 13-16 TIM1_BCCRL (0x4085)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCCRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	TIM1_BCCR	Captures the base timer count value When the basic timer is reset because of the position detection event or writing timing event the pre-reset count value is stored in the BCCR register.

13.3.12 TIM1_BARR (0x4086,0x4087)

Table 13-17 TIM1_BARRH (0x4086)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BARRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 13-18 TIM1_BARRL (0x4087)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_BARRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	TIM1_BARR	Automatic reload value of the basic timer When the counter value of the basic timer is equal to the value of the BARR register, the overflow interrupt occurs and the counter value is set to 0

13.3.13 TIM1_RARR (0x4088,0x4089)

Table 13-19 TIM1_RARRH (0x4088)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_RARRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 13-20 TIM1_RARRL (0x4089)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_RARRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	TIM1_RARR	Automatic reload value of the reload timer Overflow interrupts occur when the value of the reload timer is equal to the value of the RARR register and the value of the counter is set to 0

13.3.14 TIM1_RCNTR (0x408A,0x408B)

Table 13-21 TIM1_RCNTRH (0x408A)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_RCNTRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 13-22 TIM1_RCNTRL (0x408B)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_RCNTRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	TIM1_RCNTR	The counter value of reload timer used for counting the freewheeling

		mask time and interval between ZCP and commutation

13.3.15 TIM1_ITRIP (0x4098,0x4099)

Table 13-23 TIM1_ITRIPH (0x4098)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_ITRIP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Table 14-24 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	Function
[15:0]	TIM1_ITRIP	Filtered bus current The core module sampled the bus current and filtered it for use by the software. The default ADC channel 4 The value range is (0,32767)

14 TIM2

14.1 TIM2 operation

TIM2 has three modes of output, input timer and input counter:

1. Output mode: generate PWM output waveform
2. Input timer mode: detect the high and low level duration of input PWM, which can be used to calculate the duty cycle of PWM
3. Input counter mode: the time it takes to detect the specified number of PWM
4. QEP&RSD mode: orthogonal encoder & direct wind detection

TIM2 mainly includes:

1. The 3-bit programmable frequency divider divides the counting clock of the basic counter
2. A 16-bit upward counting base counter whose counting clock source is the output of the clock controller
3. A 16-bit up-down counter used to input counter mode and QEP&RSD mode. The counter clock source is the effective edge of the external input signal
4. Input filter module
5. Edge detection module
6. The output module generates PWM
7. Interrupt event generation

14.1.1 Clock controller

The clock controller is used to generate the counting clock source of the basic timer. The counting clock is divided by the predivider.The predivider is based on an 8-bit counter controlled by a 3-bit register PSC. 8 divider coefficients can be selected. The clock source is the internal clock.Since there is no buffer in this control register, the divider coefficient changes immediately, so the divider coefficient should be updated when the basic timer does not work.

The frequency of the counter can be calculated by the following formula:

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

Suppose the MCU clock is 24MHz (41.67ns)

Table 14-1 Register T2PSC different values correspond to different clock frequencies

T2PSC	Coefficient (16bit)	CLK (Hz)
000	0x1	24 M
001	0x2	12 M
010	0x4	6 M
011	0x8	3 M
100	0x10	1.5 M
101	0x20	750 k
110	0x40	375 k
111	0x80	187.5 K

14.1.2 Reading, writing, and counting of TIM2_CNTR

TIM2_CNTR counts only when T2CEN=1.The software writes TIM2_CNTR to change the register value directly, so the software needs to write when the count is not enabled.When the software reads TIM2_CNTR, it reads the high byte first, and the hardware will synchronously cache the low byte at this moment. When the software reads the low byte, it will read the cached data.

14.1.3 The output mode

If configure the T2MOD=01B of the TIM2_CR0 register, TIM2 works in output mode.

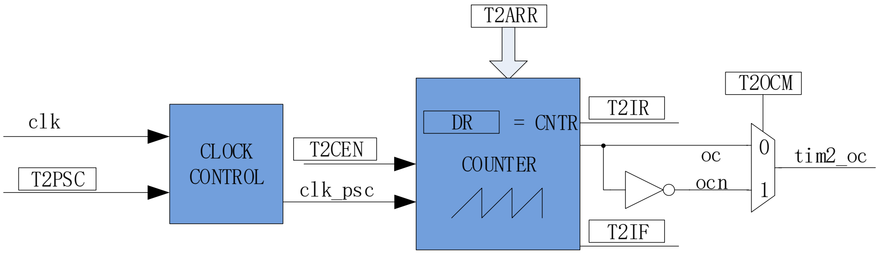


Figure 14-1 Schematic diagram of output mode

The output mode generates an output signal based on the T2OCM of the TIM2_CR0 register and the comparison result, and generates the corresponding interrupt.

14.1.3.1 The reading and writing of TIM2_ARR/TIM2_DR

In output mode, TIM2_ARR/TIM2_DR contains the preloaded registers and shadow registers.When the software writes the TIM2_ARR/TIM2_DR register, the data is saved in the preloaded register and passed to the shadow register when the overflow event T2IF or the counter is not working (T2CEN=0).

TIM2_ARR/TIM2_DR is a 16-bit register. The software needs to write high byte first, and then write low byte. The hardware guarantees that the data in the preloaded register will not be updated to the shadow register after the high byte is written to and before the low byte is written.

14.1.3.2 High/low level output mode

Configure T2OCM=0 of TIM2_CR0 register, TIM2_DR=TIM2_ARR, and output comparison signal TIM2_OC is always at low level; configure T2OCM=1 of TIM2_CR0 register, TIM2_DR= IM2_ARR, and output comparison signal TIM2_OC is always at high level;

It should be noted that only by configuring TIM2_DR=TIM2_ARR can achieve the effect of long-term high/low output. Configuring TIM2_DR=0 will have a pulse of 1 clock cycle.

14.1.3.3 PWM mode

PWM mode determines the PWM cycle according to TIM2_ARR, and TIM2_DR determines the duty cycle, which is equal to TIM2_DR/TIM2_ARR x 100%. T2OCM=0 of the TIM2_CR0 register is configured, and output low level based on the comparison result between the TIM2_DR register and the value TIM2_CNTR (TIM2_CNTR)≤TIM2_DR). T2OCM=1 of the TIM2_CR0 register is configured, and output high level based on the comparison result between the TIM2_DR register and the value TIM2_CNTR (TIM2_CNTR)≤TIM2_DR).

14.1.3.4 Interrupt event

A) when TIM2_CNTR = TIM2_DR, a relatively matching event is generated. The T2IR of the TIM2_CR1 register is set as an interrupt flag, and then the counter continues to count;

B) when TIM2_CNTR = TIM2_ARR, the overflow event is generated, the T2IF of register TIM2_CR1 is set as an interrupt flag, and TIM2_CNTR is cleared and counts again.

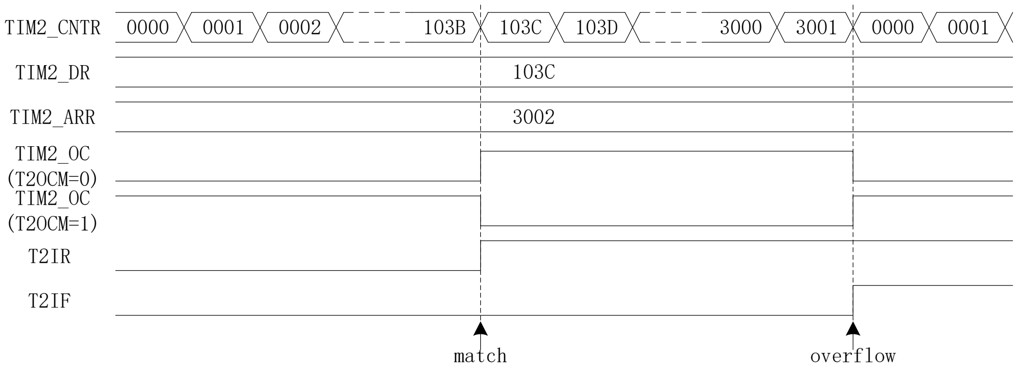


Figure 14-2 Output waveform in output mode

14.1.4 Input signal filtering and edge detection

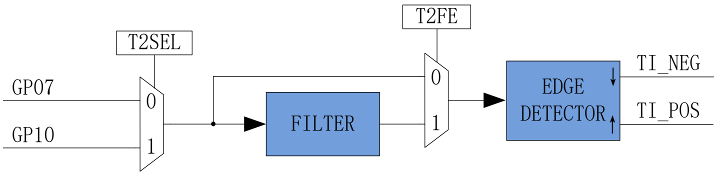


Figure 14-3 block diagram of input signal filtering and edge detection

TIM2 input signal TI comes from GP07 or GP10. According to the selection of T2SEL in PH_SEL register, the input can be selected for noise filtering. Edge detection module detects the rising edge and falling edge of input for the use of the next module.

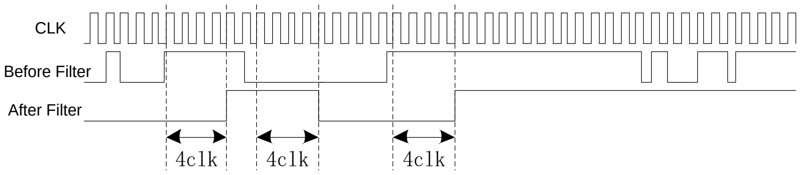


Figure 14-4 timing diagram of filtering module

Filter circuit fixed filter pulse width of 4 clock period of input noise. After T2FE = 1 of the TIM2_CR1 register is configured, even if the filtering function is available, the filtered signal will delay about 4~5 clock cycles compared with the signal before filtered.

14.1.5 Input timer mode

Configure the T2MOD=00B of the TIM2_CR0 register. TIM2 works in the input timer mode.

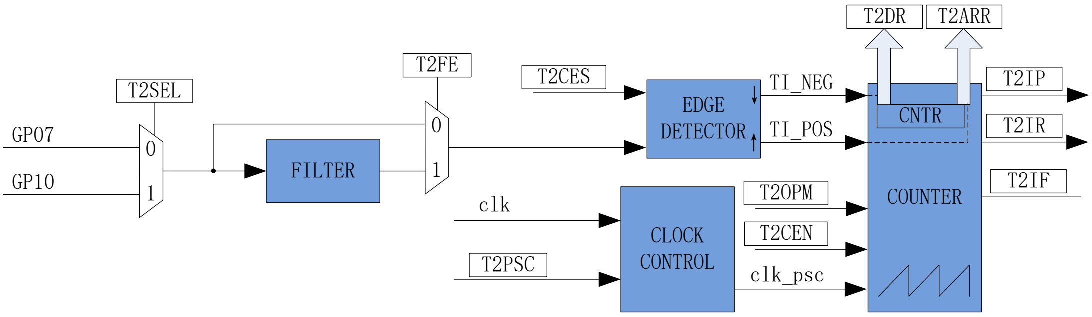


Figure 14-5 Input timer mode principle block diagram

Input timer mode to detect PWM signal pulse width and the duration of one cycle (according to T2CES=0, select two adjacent ascending edges as one cycle, and the ascending edge to the descending edge as the pulse width (high level pulse width); T2CES=1 select two adjacent descending edges as one period, and the descending edge to the ascending edge as the pulse width (low level pulse width)). TIM2_DR and TIM2_ARR are written into TIM2_CNTR respectively. Input signal can be filtered or not;

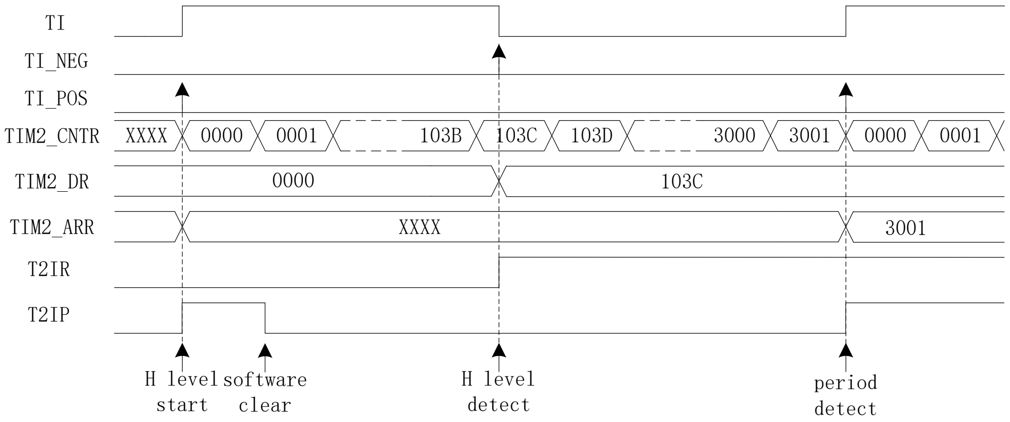


Figure 14-6 Timing diagram of input timer mode (T2CES=0)

Taking T2CES=0 as an example, T2CEN = 1 of TIM2_CR1 register is configured. The counter can be enabled, the counter can be counted upward. When the timer detects the first ascending edge of input (the descending edge is invalid), TIM2_CNTR is reset and counted again.

When the descending edge of the input is detected, that is, the input high level detection is completed. At this time, the value of TIM2_CNTR is saved into TIM2_DR. Meanwhile, the T2IR of the register TIM2_CR1 is set and TIM2_CNTR continues to count.

When the second ascending edge of the input is detected, and when a PWM cycle of the input is detected, the value of TIM2_CNTR is saved into TIM2_ARR at this time. Meanwhile, the T2IP of register TIM2_CR1 is set to 1, and TIM2_CNTR is cleared and counted again.

When the timer has not detected the second ascending edge of the input, the count value TIM2_CNTR reaches 0xFFFF, an overflow event occurs, the T2IF of the TIM2_CR1 register is set to 1, TIM2_CNTR is cleared, and TIM2_CNTR is counted again.

14.1.6 Input counter mode

Configure the T2MOD=10B of the TIM2_CR0 register. TIM2 works in input counter mode.

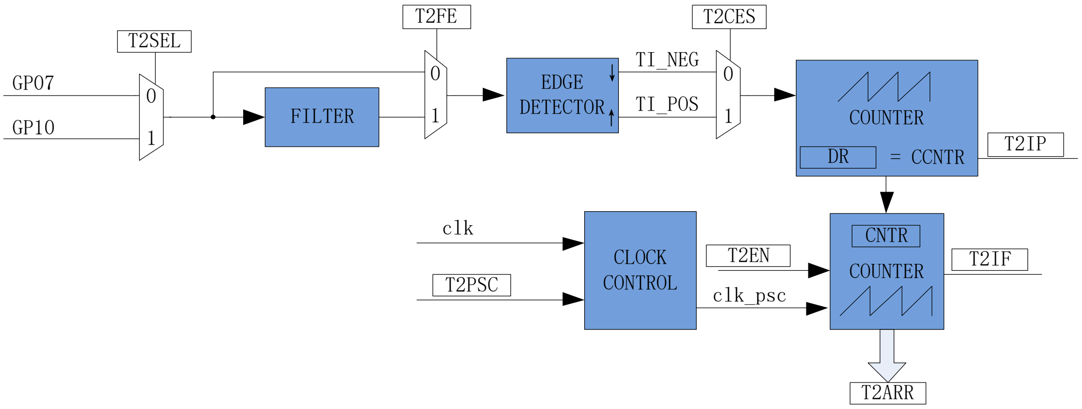


Figure 14-7 Schematic diagram of input counter mode

In input counter mode, TIM2_DR contains the preloaded and shadow registers.When the software writes the TIM2_DR register, the data is saved in the preloaded register and passed to the shadow register when the matching event T2IP, the overflow event T2IF, or the counter is not working (T2CEN=0). TIM2_DR is a 16-bit register. The software needs to write high byte first and then write low byte. The hardware guarantees that the data in the preload register will not be updated to the shadow register after the high byte is written to and before the low byte is written.

Input counter mode to detect the time required for the specified number of PWM, and save TIM2_CNTR of the basic counter into TIM2_ARR; input signal can be filtered or not; If T2CES=1 of the TIM2_CR0 register is configured, the rising edge of the input signal is used as the effective counting edge of the special counter, while the descending edge of the input signal is used as the effective edge.

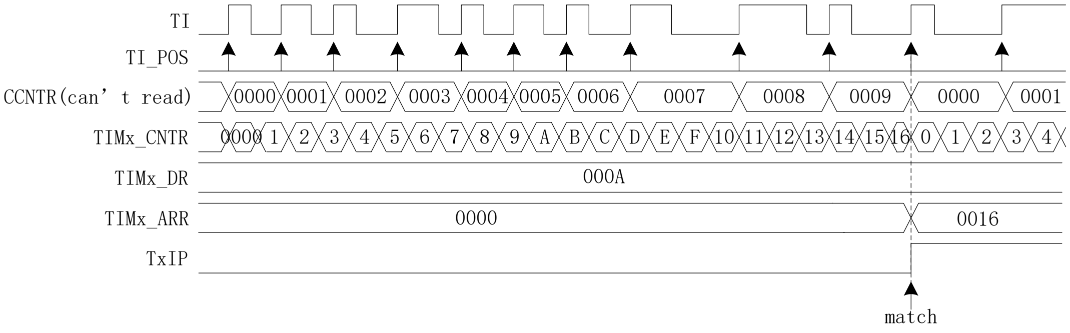


Figure 14-8 Input counter mode timing

Configure T2EN = 1 of TIM2_CR1 register, the counter can be enabled, the counter will be counted up. When the timer detects the first valid edge of input, TIM2_CNTR will be reset and the counter will be recounted.

Whenever the timer detects a valid edge, the count value of the dedicated counter CCNTR plus one; TIM2_DR sets the target value for detecting the number of PWM. When the count value of the special counter reaches the target value, the count value of the basic counter TIM2_CNTR is saved into TIM2_ARR, and the T2IP of the register TIM2_CR1 is set. TIM2_CNTR and TIM2_CCNTR are cleared and recounted.

When the number of PWM input detected has not yet reached the target value, the count value TIM2_CNTR has reached 0xFFFF, overflow event occurs, and the T2IF of TIM2_CR1 register is set.TIM2_CNTR is cleared, CCNTR is not cleared, TIM2_CNTR counts from zero, CCNTR continues to count after the previous value.

14.1.7 QEP&RSD mode

Configure the T2MOD=11B of the TIM2_CR0 register to let TIM2 work in QEP&RSD mode.

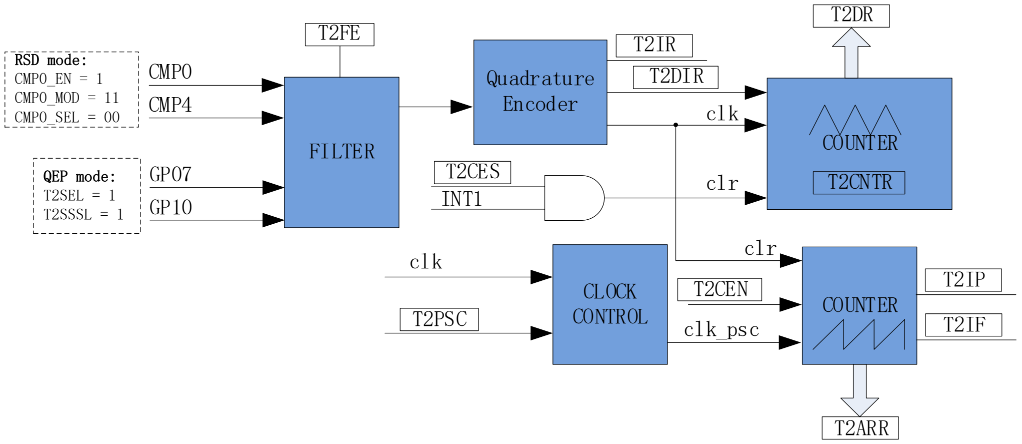


Figure 14-9 Schematic diagram of QEP&RSD mode

QEP&RSD mode obtains relative position, direction and speed information by detecting orthogonal input of two channels GP07 and GP10 (QEP mode, the phase of the two inputs is generally 90 degrees) or CMP0 and CMP4 (RSD mode, the phase of the two inputs is generally 60 degrees) as input, e is sent into the orthogonal decoding module after filtering modul, the effective count along and direction T2DIR, direction change will produce T2IR interrupt flag.

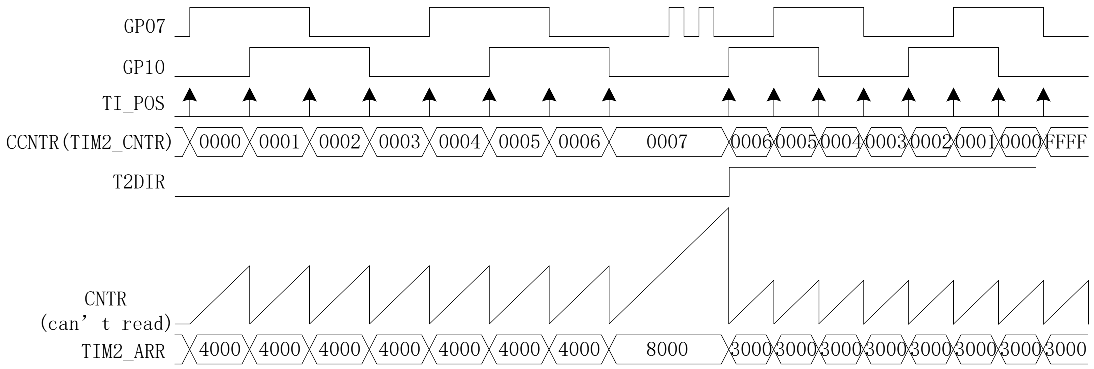


Figure 14-10 Timing diagram of QEP mode (the phase of orthogonal input in RSD mode is different from that in QEP mode)

The dedicated counter is an up or down, counting clock is the effective edge of orthogonal coding module output. If T2DIR=0, the direction is positive, counting up, as effective edges approach, the counter add 1; if T2DIR=1, the direction is negative, counting down, as effective edges approach, the counter minus 1. The dedicated counter can be cleared by external interrupt INT1. The mechanical zero point of the encoder connects to any port the external interrupt 1, enabling INT1 interrupt, configure T2CES=1 at the same time, when the external interrupt 1, the current value of dedicated counter is saved into TIM2__DR, the dedicated counter resets at the same time. After the dedicated counter counts from 0 to 65535 it is cleared automatically, after counting from 65535 to 0, it is set to 65535 automatically. Read the register TIM2_CNTR to get the dedicated counter value.

The base counter is an up counter with a counting clock that can be divided to record the time of two valid counting edges.When the effective edge comes along, the current count value of the base counter is stored in TIM2_ARR, while the base counter is cleared and the T2IP interrupt flag is generated.When the base counter counts to 0xFFFF, the count overflows and the T2IF interrupt flag is generated.

14.1.7.1 RSD comparator sampling

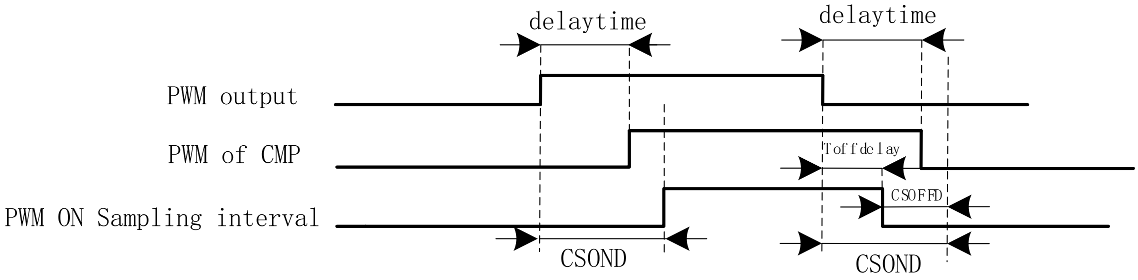


Figure 14-11 PWM ON sampling mode

The interference reflected by PWM output to the comparator has a delay relative to the hopping edge of PWM, which is mainly affected by the following factors: the size of the driving resistance, the switching speed of MOS, the setting of the input delay and hysteresis of the comparison, and the delaytime in the figure is the delaytime between the IC output level and the level detected by the comparator. When conducting high level sampling, the sampling interval should be surrounded by the actual high level read on the comparator, and the delay time CSOND is first set to overcome the delay and the oscillation of the MOS switch. At the same time, if the value of CSOFFD is not set, the end time of the sampling interval is the delay CSOND after the falling edge of PWM output wave. At this time, the actual sampling window has jumped out of the time corresponding to the high level (PWM of CMP) on the comparator, so setting the value of CSOFFD makes the actual sampling window close after the falling edge of PWM out wave is delayed Toffdelay (Toffdelay=CSOND- CSOFFD)

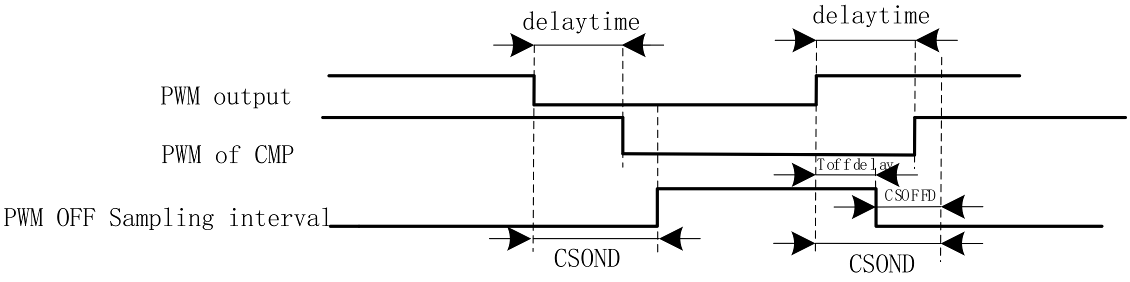


Figure 14-12 PWM OFF sampling mode

Similarly, when performing low level sampling, the sampling interval should be surrounded by the low level actually read on the comparator, and the delay time CSOND is first set to overcome the delay and the oscillation of the MOS switch. At the same time, if the value of CSOFFD is not set, the end time of the sampling interval is the delay CSOND after the rising edge of PWM output wave. At this time, the actual sampling window has jumped out of the time corresponding to the low level (PWM of CMP) on the comparator. Therefore, setting the value of CSOFFD makes the actual sampling window close after the rising edge of PWM out wave is delayed Toffdelay (Toffdelay=CSOND- CSOFFD).

Measure the delay of PWM output to the comparator: set the SAMSEL=00 of CMP_CR3 register to forbid the comparator's sampling function, set the CMPSEL of CMP_CR3 register to output the comparator's comparison value, enable PWM output and comparator, manually rotate the motor to flip the comparator value, and measure the delay between PWM output and the comparator's output.

14.1.8 Step mode

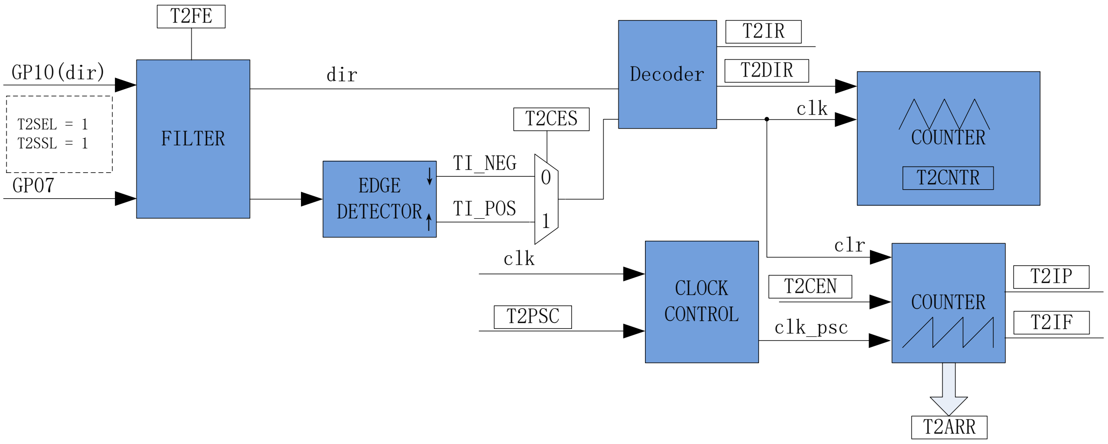


Figure 14-13 Principle block diagram of step mode

Step mode detects the input of two channels to obtain relative position, direction and speed information.GP10 as the direction input, GP07 as the pulse input, select the ascending edge or descending edge as the effective edge according to T2CES. After filtering module, it is sent into the decoding module to get the effective counting edge and direction T2DIR. If the direction changes, T2IR interrupt flag will be generated (note: T2DIR and T2IR only will change when the effective edge of GP07 arrives after GP10 is changed. If you want to generate the interrupt as soon as GP10 changes, you need to use an external interrupt 1)

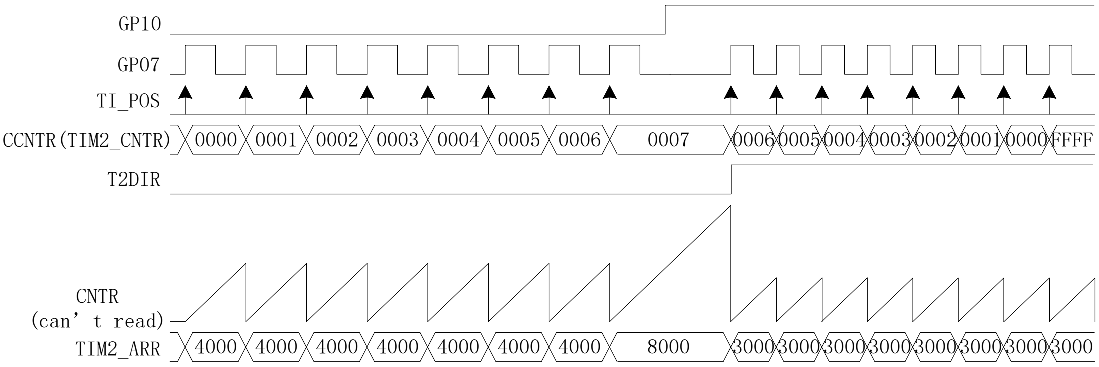


Figure 14-14 Timing diagram of step mode

A dedicated counter is an up-down counter with a counting clock as the effective counting edge of the output of the encoding module.GP10=0, when GP07 comes along effectively, T2DIR=0, direction is positive, count up, count up one;GP10 is equal to 1, when GP07 is effectively coming along, T2DIR is equal to 1, in the opposite direction, count down, count down one.Special counter from 0 to 65535 automatically clear 0, from 65535 to 0 automatically set to 65535, read register TIM2_CNTR is worth to the value of the special counter.

The dedicated counter is an up-down counter. This count clock is a valid count edge outputed by the encoding module. If GP10 = 0, when the valid edge of GP07 comes, T2DIR = 0; at this time, the direction is positive, count up, and the counter is incremented by one. If GP10 = 1, when the valid edge of GP07 comes, T2DIR = 1, at this time, the direction is reverse, count down and the counter decrements by one. The dedicated counter is automatically cleared after it is incremented from 0 to 65535; it is automatically set to 65535 after it is decremented from 65535 to 0. Read the value of register TIM2_CNTR to get the value of the dedicated counter.

The basic counter is an up counter. This count clock can be divided to record the time of two valid count edges. When a valid count edge comes, the current count value of the basic counter is stored in TIM2_ARR, and the counter is cleared, and an interrupt flag T2IP is generated. When the basic counter reaches 0xFFFF, the count overflows and the interrupt flag T2IF is generated.

14.2 TIM2 register

14.2.1 TIM2_CR0 (0xA1)

Table 14-2 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	Function
[7:5]	T2PSC	Counter clock frequency division selection These bits are used to perform N frequency division on the MCU clock as the counting clock of the basic counter. It is assumed that the MCU clock is 24MHz (41.67ns). 000:0x1 (24MHz) 001:0x2 (12MHz) 010:0x4 (6MHz) 011:0x8 (3MHz) 100:0x10 (1.5Mhz) 101:0x20 (750kHz) 110:0x40 (375kHz) 111:0x80 (187.5kHz)
[4]	T2OCM	Output mode: compare mode selection 0:TIM2_CNTR≤TIM2_DR, output 0;TIM2__CNTR >TIM2_DR, output 1 1:TIM2_CNTR≤TIM2_DR, output 1;TIM2_CNTR>TIM2_DR, output 0 Input counter mode: none Input timer mode: none QEP&RSD mode & step mode: mode selection 0: QEP&RSD mode 1: step mode
[3]	T2IRE	Output mode: compare matched interrupt enablement Input timer mode: pulse width detection interrupt enablement Input counter mode: none QEP&RSD mode & step mode: direction change interrupt enablement

		0: disable event interrupt; 1: enable event interrupt
[2]	T2CES	Input timer mode: cycle edge selection 0: the interval of adjacent two rising edges is used as a period, and the interval between rising edge and the falling edge is used as a pulse width (high level pulse width). 1: the interval of adjacent two falling edges is used as a period, and the interval between falling edge and the rising edge is used as a pulse width (low level pulse width). Input counter mode & step mode: count valid edge selection 0: count the descending edge 1: counting the rising edge QEP&RSD mode: external interrupt 1 (zero) disable pulse counter 0: Disable 1: Enable
[1:0]	T2MOD	Mode selection 00: input timer mode 01: output mode 10: input counter mode 11: QEP&RSD mode & step mode

14.2.2 TIM2_CR1 (0xA9)

Table 14-3 TIM2_CR1 (0xA9)

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

Bit	Name	Function
[7]	T2IR	Output pattern: compare matching flag This bit is set by hardware when the counter value of TIM2_CNTR matches the comparison value TIM2_DR. It is cleared by software. Input timer mode: pulse width detection flag. Timer detects the input pulse width (select counting from the rising edge to the falling edge or from the falling edge to the rising edge depending on T2CES), which is set by hardware.It’s cleared by software. Input counter mode: none QEP&RSD mode & step mode: direction change marker 0: no event occurred; 1: the event occurred.
[6]	T2IP	Output mode: none Input timer mode: PWM cycle detection flag Timer detects the input of a PWM cycle (select counting from the rising edge to the rising edge or from the falling edge to the falling edge depending on T2CES). The

		bit is set by hardware 1.It is cleared by software. Input counter mode: input PWM count matching flag When the number of input PWM reaches the value of TIM2_DR, the bit is set by hardware. It is cleared by software. QEP&RSD mode & step mode: the valid input edge change interrupt flag When the input edge is a valid edge, the bit is set by hardwar. It is cleared by software. 0: no event occurred; 1: the event occurs.
[5]	T2IF	Output mode: overflow mark on counter When the counter value of TIM2_CNTR matches the comparison value of TIM2_ARR, TIM2_CNTR is cleared, and the bit is set by hardware.It clears 0 by software. Input timer mode: overflow mark on counter Timer has not detected the input of a PWM cycle (that is, select counting from the rising edge to the rising edge or from the falling edge to the falling edge depending on T2CES), and the counter value of TIM2_CNTR is added to 0xFFFF to generate the overflow event, TIM2_CNTR is cleared, and the bit is set by hardware.It is cleared by software. Input counter mode: overflow mark on basic counter When the number of PWM input has not reached the value of TIM2_DR, and the value of the basic counter of TIM2_CNTR is added to 0xFFFF, the overflow event is generated, and TIM2_CNTR is cleared, which is set by hardware.It is cleared by software. QEP&RSD mode & step mode: overflow mark on basic counter When the base counter is incremented to 0xFFFF, an overflow event occurs and the basic counter is cleared, and it is set by the hardware to 1.It is cleared by software. 0: no event occurred 1: the event occured
[4]	T2IPE	Output mode: none Input timer mode: enable PWM cycle detect interrupt Input counter mode: enable input PWM count matching interrupt QEP&RSD mode & step mode: input effective edge change interrupts enablement 0: disable event interrupt 1: enable event interrupt
[3]	T2IFE	Output mode: overflow interrupt enabled on counter Input timer mode: enable overflow interrupt of counter Input counter mode: enable the overflow interrupt of basic counter QEP&RSD mode & step mode: enable overflow interrupt of basic counter 0: disable update event interrupt 1: enable update event interrupt
[2]	T2FE	Enable input noise filtering When the noise pulse width is less than 4 clock cycles, the noise will be filtered out.Assuming that the MCU clock is 24MHz(41.67ns), the filter pulse width is

		166.67ns 0: no filtering function 1: enable filtering function
[1]	T2DIR	Only for QEP&RSD& step mode: current direction 0: forward 1: reverse
[0]	T2EN	Counter enable 0: disable counters 1: enable counter

14.2.3 PI_LPF_CR (0xF9)

Table 14-4 PI_LPF_CR (0xF9)

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

Bit	**Name**	**Function**
[7]	T2SS	TIM2 stepper motor mode input mode selection 0: P10 is the direction and P07 is the pulse count 1: P10 is the reverse pulse count, P07 is the positive pulse count
[3]	RSV	Reserved
[2:0]		See Table 10-1

14.2.4 TIM2_CNTR (0xAA,0xAB)

Table 14-5 TIM2_CNTRH (0xAB)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM2_CNTRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 14-6 TIM2_CNTRL (0xAA)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIM2_CNTRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	**Name**	**Function**
[15:0]	TIM2_CNTR	Output mode/input timer mode/input counter mode: the counter value of the basic counter QEP&RSD mode & step mode: the count value of a dedicated counter It will be cleare automatically when TIM2__CNTR counts to 0xFFFF

14.2.5 TIM2_DR (0xAC,0xAD)

Table 14-7 TIM2_DRH (0xAD)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_DRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 14-8 TIM2_DRL (0xAC)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_DRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	TIM2_DR	Output mode: compare matching values (software write) Input timer mode: the count value of the input pulse width (select counting from the rising edge to the falling edge or from the falling edge to the rising edge depending on T2CES) (written by hardware) Input counter mode: the number of input PWM (written by software) QEP&RSD mode: when T2CES=1, when external interrupt 1 (zero) arrives, the value of the dedicated counter (written by hardware) Step mode: none

14.2.6 TIM2_ARR (0xAE,0xAF)

Table 14-9 TIM2_ARRH (0xAF)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_ARRH							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 14-10 TIM2_ARRL (0xAE)

Bit	7	6	5	4	3	2	1	0
Name	TIM2_ARRL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:0]	TIM2_ARR	Output mode: reload value (written by software) Input timer mode: detect the count value of a PWM cycle (which is selected according to T2CES, i.e. ascending edge to ascending edge or descending edge to descending edge) (written by hardware) Input counter mode: input the count value of the basic counter when PWM count matches (written by hardware) QEP&RSD mode & stepping mode: the count value of the basic counter when the input edge is detected as a valid edge (written by hardware)

15 TIM3 / TIM4

15.1 TIM3/TIM4 operation

TIM3/TIM4 has two modes of output and input timer:

1. Output mode: generate output waveform (PWM, single mode)
2. Input timer mode: detect the duration of between high and low level of input PWM, which can be used to calculate the duty cycle of PWM

TIM3/TIM4 mainly includes:

1. The 3-bit programmable frequency divider divides the counting clock of the basic counter
2. A 16-bit upward counting base counter whose counting clock source is the output of the clock controller
3. Input filter module
4. Edge detection module
5. The output module generates PWM and single comparison output
6. Interrupt event generation

15.1.1 Clock controller

The clock controller is used to generate the counting clock source of the basic timer. The counting clock is divided by the Pre-divider.The Pre-divider is based on an 8-bit counter controlled by a 3-bit register PSC. 8 divider coefficients can be selected. The clock source is the internal clock.Since there is no buffer in this control register, the divider coefficient changes immediately, so the divider coefficient should be updated when the basic timer does not work.

The frequency of the counter can be calculated by the following formula:

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

Suppose the MCU clock is 24MHz (41.67ns)

Table 15-1 Registers TxPSC with different values correspond to different clock frequencies

TxPSC	Coefficient (16bit)	CLK (Hz)
000	0x1	24 M
001	0x2	12 M
010	0x4	6 M
011	0x8	3 M
100	0x10	1.5 M
101	0x20	750K
110	0x40	375 K
111	0x80	187.5K

15.1.2 Read, write, and count TIMx_CNTR

TIMx_CNTR counts only when TxEN=1.The software writes TIMx_CNTR to change the register value directly, so the software needs to write when the count is not enabled.When the software reads

TIMx_CNTR, it reads the high byte first, and the hardware caches the low byte at this moment synchronously. When the software reads the low byte, it reads the cached data.

15.1.3 Output mode

Configure the TxMOD=1 of the TIMx_CR0 register, TIM works in output mode.

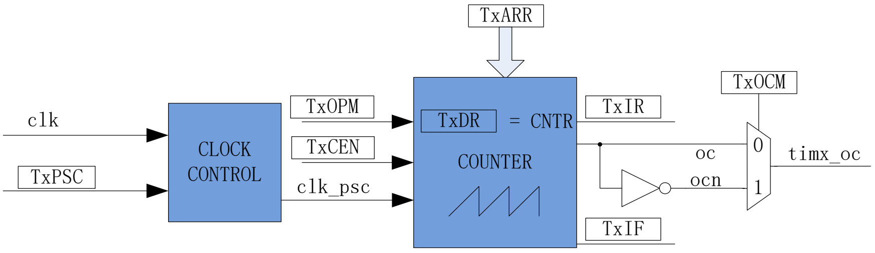


Figure 15-1 Schematic diagram of output mode

The output mode generates an output signal based on the TxOCM of the TIMx_CR0 register configured and the comparison result, and generates the corresponding interrupt.

15.1.3.1 High/low level output mode

Configure TxOCM= 0 of TIMx_CR0 register, TIMx_DR= TIMx_ARR, and output comparison signal TIMx_OC which is always at low level; configure TxOCM= 1 of TIMx_CR0 register, TIMx_DR= TIMx_ARR, and output comparison signal TIMx_OC which is always at high level;

It is important to note that only by configuring TIMx_DR= TIMx_ARR can achieve the effect of long-term high/low output. Configuring TIMx_DR=0 will have a pulse of 1 clock cycle.

15.1.3.2 PWM mode

PWM mode determines the PWM cycle based on TIMx_ARR, and TIMx_DR determines the duty cycle which is equal to TIMx_DR/TIMx_ARRx100%.TxOCM= 0 of the TIMx_CR0 register is configured, and the output is based on the comparison result between the TIMx_DR register and the value TIMx_CNTR (TIMx_CNTR)≤TIMx_DR) outputs low levels and vice versa.TxOCM= 1 of the TIMx_CR0 register is configured, and the output is based on the comparison result between the TIMx_DR register and the value TIMx_CNTR (TIMx_CNTR)≤TIMx_DR) outputs high levels and vice versa.

15.1.3.3 Interrupt event

A) When TIMx_CNTR = TIMx_DR, a relatively matching event is generated, the TxIR of the TIMx_CR1 register is set, and the counter continues to count;

B) When TIMx_CNTR = TIMx_ARR, an overflow event occurs. The TxIF of register TIMx_CR1 is set and the counter is reset. According to whether the TxOPM of register TIMx_CR0 is recounted or not, if TxOPM=1, the count is stopped; if TxOPM=0, the count restarts.

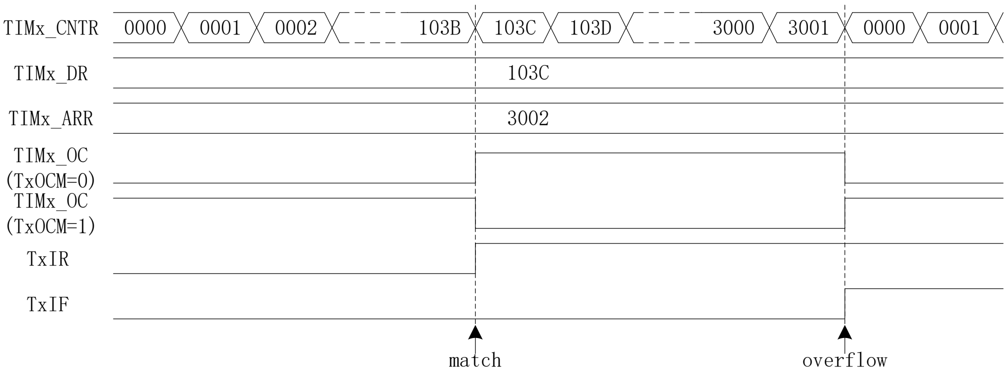


Figure 15-2 Output mode output waveform

15.1.4 Input signal filtering and edge detection

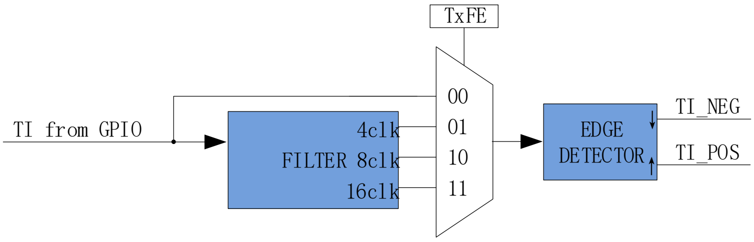


Figure 15-3 Block diagram of input signal filtering and edge detection

The input signal TI of TIM3/TIM4 comes from GP11/GP01, and the input can be selected for noise filtering. Edge detection module detects the rising edge and falling edge of input for the use of the next module.

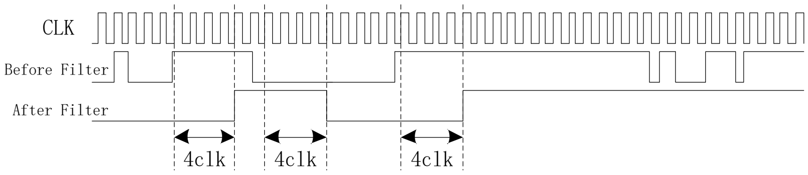


Figure 15-4 Diagram of filtering module filtering 4clk sequence

The filter circuit can be configured to filter input noise with a pulse width of 4/8/16 clock cycles.Configure TxFE = 00B of TIMx_CR1 register to turn off filtering function.Configure the TxFE = 01/10/11B of TIMx_CR1 register to enable the filtering function, the filtered signal will delay about 4~5/8~9/16~17 clock periods compared with the signal before filtering.

15.1.5 Input timer mode

Configure the TxMOD=0 of the TIMx_CR0 register to let TIM3/4 work on the input timer mode

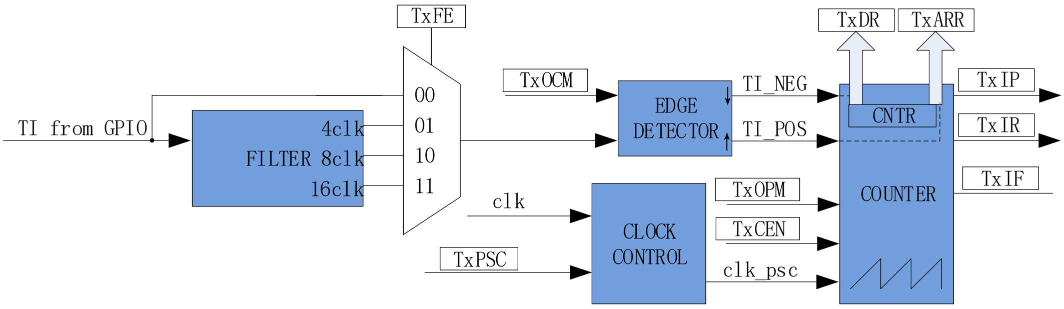


Figure 15-5 Diagram of input timer mode schematic block

Input timer mode to detect PWM signal pulse width and the duration of one cycle (according to TxOCM=0, select two adjacent ascending edges as one cycle, and the ascending edge to the descending edge as the pulse width (high level pulse width); TxOCM=1 select two adjacent descending edges as one period, and the descending edge to the ascending edge as the pulse width (low level pulse width)). TIMx_CNTR values are stored in TIMx_DR and TIMx_ARR respectively. Input signal can be filtered or not by selection;

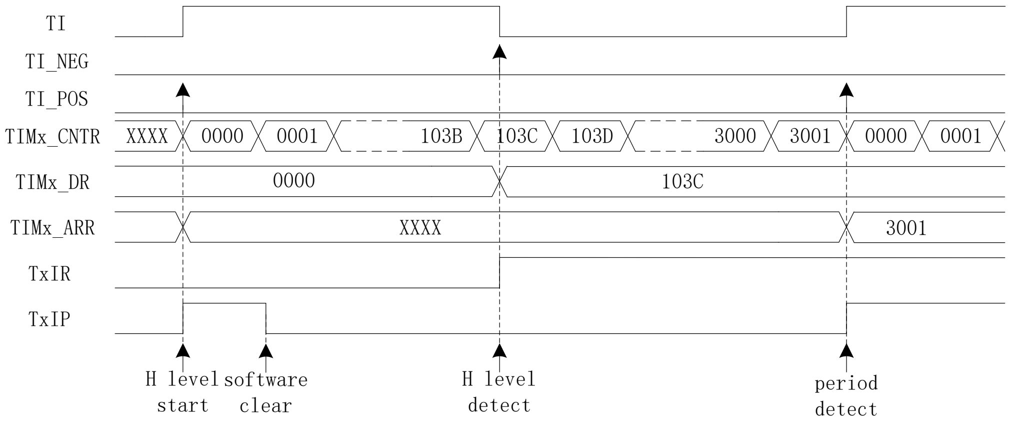


Figure 15-6 Timing diagram of input timer mode (TxOCM=0)

Take TxOCM=0 as an example, enable counter by configuring TxEN of TIMx_CR1 register to 1, and the counter counts up;

When the descending edge of the input is detected, that is, the input high level detection is completed. At this time, the value of TIMx_CNTR is saved into TIMx_DR. Meanwhile, the interrupt flag TxIR of the register TIMx_CR1 is set and TIMx_CNTR continues to count up

When the second rising edge of input is detected, that is, when a PWM cycle of input is detected, the value of TIMx_CNTR is saved into TIMx_ARR at this time. Meanwhile, the interrupt flag TxIP of register TIMx_CR1 is set. TIMx_CNTR is cleared and select whether to restart counting according to TxOPM. If TxOPM=1, stop counting, if TxOPM=0, count again.

When the timer has not detected the second rising edge of input, the count value TIMx_CNTR reaches 0xFFFF, and an overflow event occurs. If the TxIF of the TIMx_CR1 register is set , TIMx_CNTR is cleared.TxOPM=0, count again.

15.2 TIM3/TIM4 registers

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

Table 15-2 TIMx_CR0 (0x9C/0x9E)

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

Bit	Name	Function
[7:5]	TxPSC	Counter clock frequency division selection These bits are used to perform N frequency division on the MCU clock to generate the counting clock of the basic counter. It is assumed that the MCU clock is 24MHz (41.67ns). 000:0x1 (24MHz) 001:0x2 (12MHz) 010:0x4 (6MHz) 011:0x8 (3MHz) 100:0x10 (1.5MHz) 101:0x20 (750kHz) 110:0x40 (375kHz) 111:0x80 (187.5kHz)
[4]	TxOCM	**Output mode:** compare mode selection 0: If TIMx_CNTR≤TIMx_DR, then output 0; if TIMx_CNTR>TIMx_DR, then output 1 1: If TIMx_CNTR≤TIMx_DR, output 1; if TIMx_CNTR>TIMx_DR, then output 0 **Input timer mode:** cycle edge selection 0: the adjacent two rising edges used as a period, and the rising edge to the descending edge used as a pulse width (high level pulse width). 1: the adjacent two descending edges used as a period, and the descending edge to the ascending edge used as a pulse width (low level pulse width).
[3]	TxIRE	**Output mode:** compare matched interrupt enablement Input timer mode: pulse width detection interrupt enablement 0: disable event interrupt; 1: enable event interrupt
[2]	RSV	Reserved
[1]	TxOPM	A single mode The following events occur **Output mode:** counter overflows **Input timer mode:** PWM cycle detection or counter overflow 0: the counter does not stop when an update event occurs; 1: when an update event occurs, the counter stops (clears TxCEN).
[0]	TxMOD	Working mode selection 0: input timer mode 1: output mode

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

Table 15-3 TIMx_CR1 (0x9D/0x9F)

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

Bit	Name	Function
[7]	TxIR	**Output pattern:** compare matching flag This bit is set by hardware when the counter value of TIMx_CNTR matches the comparison value of TIMx_DR.It is cleared by software. **Input timer mode:** pulse width detection flag Timer detects the input pulse width (select counting from the rising edge to the falling edge or from the falling edge to the rising edge depending on TxOCM), which is set by hardware 1.It is cleared by software. 0: no event occurred; 1: the event occured
[6]	TxIP	**Output mode: none** **Input timer mode:** PWM cycle detection flag Timer detects the input of a PWM cycle (select counting from the rising edge to the rising edge or from the falling edge to the falling edge depending on TxOCM). The bit is set by hardware. It is cleared by software. 0: no event occurred; 1: the event occurred.
[5]	TxIF	**Output mode:** overflow mark on counter When the counter value of TIMx_CNTR matches the comparison value of TIMx_ARR, TIMx_CNTR is cleared to zero and the bit is set by hardware. It is cleared by software. **Input timer mode:** overflow mark of counter Timer has not detected the input of a PWM cycle (that is, ascending edge to ascending edge), and the counter value TIMx_CNTR is accumulate to 0xFFFF, resulting in an overflow event. TIMx_CNTR is cleared to zero, and the bit is set by hardware 1.It is cleared by software. 0: no event occurred; 1: the event occurred.
[4]	TxIPE	**Output mode:** none **Input timer mode:** PWM cycle detect interrupt enablement 0: disable event interrupt; 1: enable event interrupt
[3]	TxIFE	**Output mode:** enable overflow interrupt of counter **Input timer mode:** enable overflow interrupt of counter 0: disable update event interrupt; 1: enable update event interrupt

[2:1]	TxFE	Input noise pulse width selection, when the noise pulse width is less than the set value, the noise will be filtered.Suppose the MCU clock is 24MHz(41.67ns) 00: no filtering 01:4 clock cycles, 4 x 41.67ns 10:8 clock cycles, 8 x 41.67ns 11:16 clock cycles, 16 x 41.67ns
[0]	TxEN	Enable basic counter 0: disable counters; 1: enable counter

15.2.3 TIMx_CNTR (0xA2,0xA3/0x92,0x93) (x=3/4)

Table 15-4 TIMx_CNTRH (0xA3/0x93)

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

Table 15-5 TIMx_CNTRL (0xA2/0x92)

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

Bit	Name	Function
[15:0]	TIMx_CNTR	The count of the base counter

15.2.4 TIMx_DR (0xA4,0xA5/0x94,0x95)(x=3/4)

Table 15-6 TIMx_DRH (0xA5/0x95)

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

Table 16-7 TIMx_DRL (0xA4/0x94)

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

Bit	Name	Function
[15:0]	TIMx_DR	Output mode: compare matching values (written by software) Input timer mode: the count of the input pulse width (select counting from the rising edge to the falling edge or from the falling edge to the

		rising edge depending on TxOCM) (written by hardware)

15.2.5 TIMx_ARR (0xA6,0xA7/0x96,0x97) (x=3/4)

Table 16-8 TIMx_ARRH (0xA7/0x97)

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

Table 16-9 TIMx_ARRL (0xA6/0x96)

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

Bit	Name	Function
[15:0]	TIMx_ARR	Output mode: reload value (written by software) Input timer mode: the count value of a PWM cycle (select counting from the rising edge to the rising edge or from the falling edge to the falling edge depending on TxOCM) (written by hardware)

16 SYS_TICK

16.1 Instructions

SYS_TICK is used to generate interrupts at a fixed time. Configure the SYS_TICK register to set the interrupt cycle and set the SYSTIE of the DRV_SR register will enable SYS_TICK and generate an interrupt. The interrupt entry is 10 and is reused with TIM4 interrupt entry.

16.2 Register

16.2.1 DRV_SR (0x4061)

Table 16-1 DRV_SR (0x4061)

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

Bit	Name	Function
[7]	SYSTIF	SYS TICK interrupt flag The bit is set to 1 by the hardware.It clears 0 by software. 0: no event occurred 1: SYS TICK interrupt occurred
[6]	SYSTIE	Enable SYS TICK interrupts SYS TICK interrupt can be generated after enabling 0: Disable 1: Enable
[5:0]		Please refer to 17.2.2

16.2.2 SYST_ARR (0x4064,0x4065)

Table 16-2 SYST_ARRH (0x4064)

Bit	7	6	5	4	3	2	1	0
Name	SYST_ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	1	1	0	1

Table 16-3 SYST_ARRL (0x4065)

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

field	The name of the	describe

[15:0]	SYST_ARR	The period value of SYS_TICK Set this value to determine the period of the SYS_TICK interrupt, and the default period is 1ms The calculation formula: SYS_TICK frequency =24M/ (SYST_ARR+1) The value range is (0,65535)

17 Driver

17.1 Instructions

17.1.1 Operating Introduction

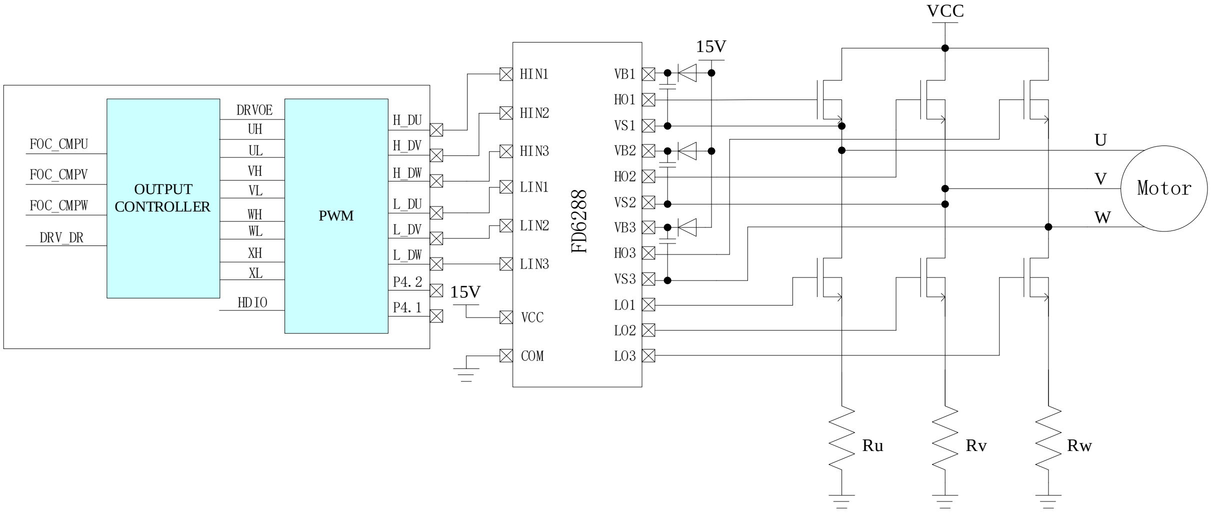


Figure 17-1 Block diagram of DRIVER module of FU6812

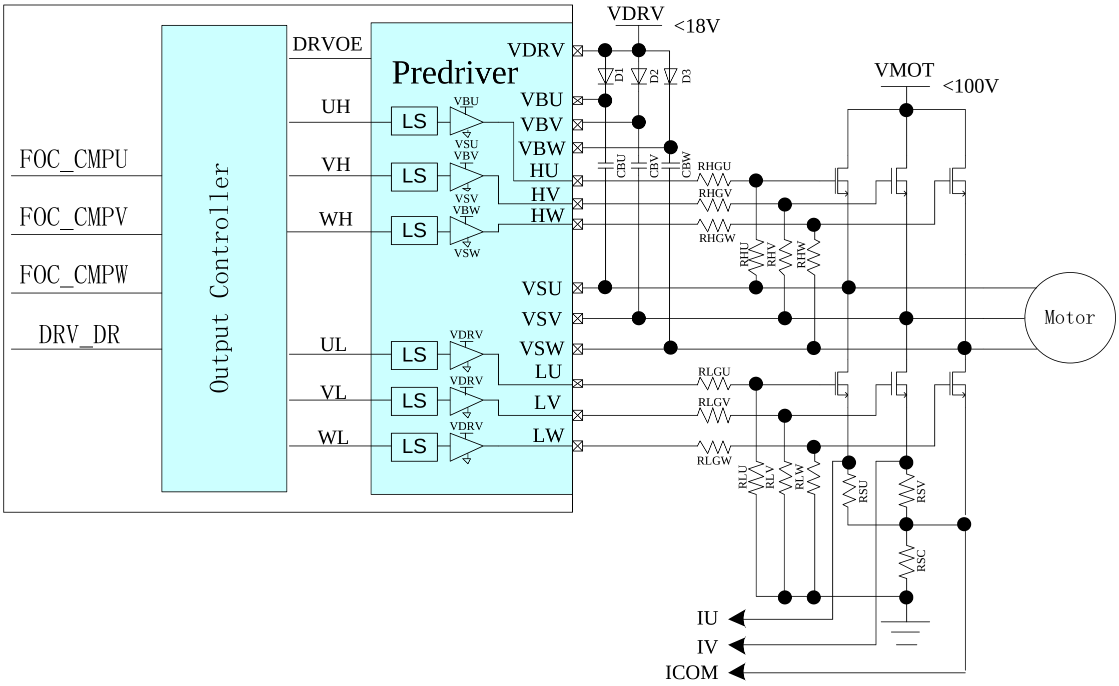


Figure 17-2 Block diagram of the DRIVER module of FU6861

FOC_CMPU/V/W is the three-way comparison value outputed by FOC module, and DRV_DR is the comparison value set by software. The above comparison value outputs four sets of level signal U/V/W/X to PWM output (FU6812) or three sets of level signal U/V/W to Predriver (FU6861) after the output control module.Among them, U/V/W three-way output is used for BLDC control, and U/V/W/X four-way output is used for stepping motor control.

17.1.2 Output control module

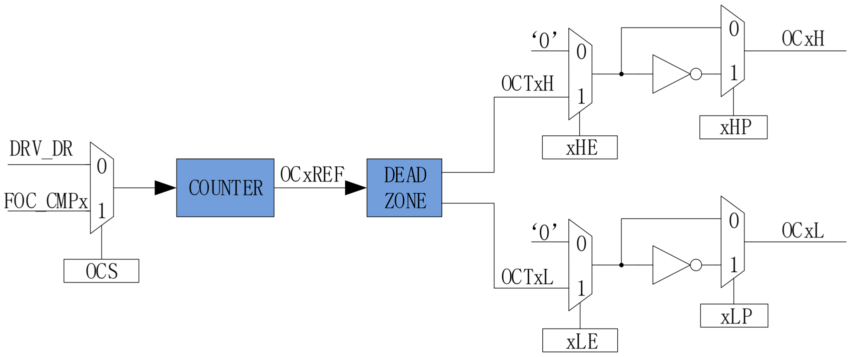


Figure 17-3 output control module front level block diagram

Before configuring Driver module to work, MESEL=1 of DRV_CR register needs to be configured, that is, the motor engine ME chooses FOC/SVPWM/SPWM mode, otherwise, it is BLDC control mode.

When OCS=0, the comparison value of PWM comes from DRV_DR, and the output PWM signal takes OCTxH as the reference. When OCxH and OCxL output at the same time, OCTxL is an inverted output. When OCS=1, the comparison value of PWM comes from FOC, and the output PWM signal takes OCTxL as the reference. When OCxH and OCxL output at the same time, OCTxH is an inverted output.

17.1.2.1 Counting comparison module

Through the OCS of DRV_CR register, the three-way comparison value FOC_CMPU/V/W outputed by FOC module or the comparison value DRV_DR set by software can be selected, and the four-way original PWM signal OCxREF can be obtained after the counter comparison. DRV_DR is used to realize the motor pre-charging, braking and BLDC control.When CNTR is less than the target value, OCxREF outputs high level; otherwise, it outputs low level.

Configure OCS=1 of DRV_CR register, select FOC_CMPU/V/W,which is the three-way comparison value outputed by FOC module, to compare it with the count to generate OC1REF/OC2REF/OC3REF.

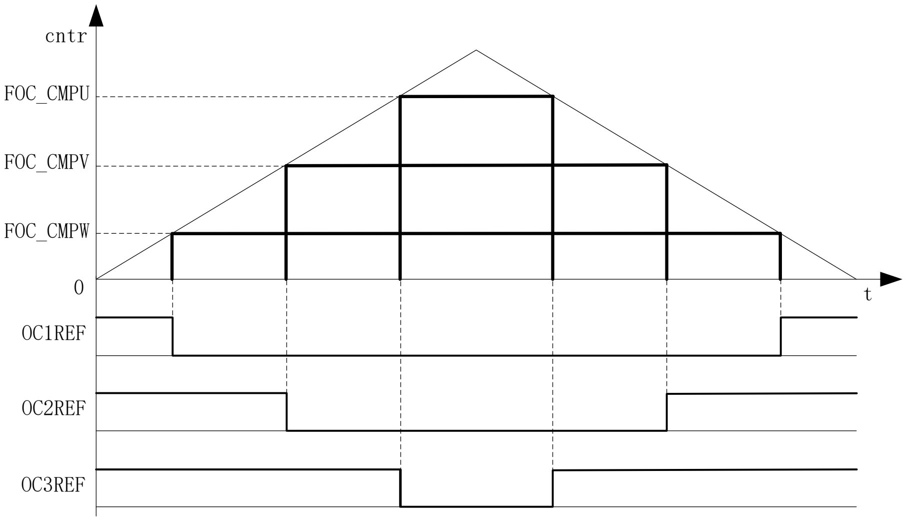


Figure 17-4 Diagram of PWM

Configure OCS =0 of DRV_CR register, select the comparison value DRV_DR set by software and compare it with the count value, and generate OC1REF/OC2REF/OC3REF with the same duty cycle of three channels.

Duty cycle = DRV_DR/ DRV_ARRx 100% (assuming DRV_ARR=750, DRV_DR=375, then duty cycle =50%)

17.1.2.2 Dead zone module

OCxREF supports dead zone insertion.For complementary output, dead zone insertion is enabled if the FOC_DTR register is not equal to 0. Each channel has an 8-bit dead zone generator, and the dead zone delay of the four channels is configured the same through DRV_DTR, and the dead zone time is set through DRV_DTR. When the rising edge of OCxREF occurs, the actual rising edge of OCxL will be delayed for a period of time than the rising edge of OCxREF, the time is determined by DRV_ DTR; When the falling edge of OCxREF occurs, the actual rising edge of OCxH will be delayed for a period of time than the falling edge of OCxREF, the time is determined by DRV_ DTR.

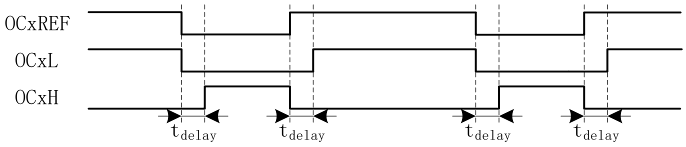


Figure 17-5 Complementary output with dead zone

17.1.2.3 Output enablement and polarity

By configuring the xHE and xLE of the DRV_CMR register, you can choose whether the output mode is an invalid level or the complementary output with the dead zone.Configure the polarity of the output by configuring the xHP and xLP of the DRV_CMR register. DRV_CMR is generally configured by software. When applied to BLDC square wave control, automatic commutation can be achieved by configuring TIMER1 to automatically control DRV_CMR. Configure MESEL=0 of DRV_CR register, that is, motor engine ME selects BLDC control mode. When TIM1 generates commutation event, the relative TIM1_DBRx is updated into DRV_CMR and CMP_CR2[4:3] of cmparator

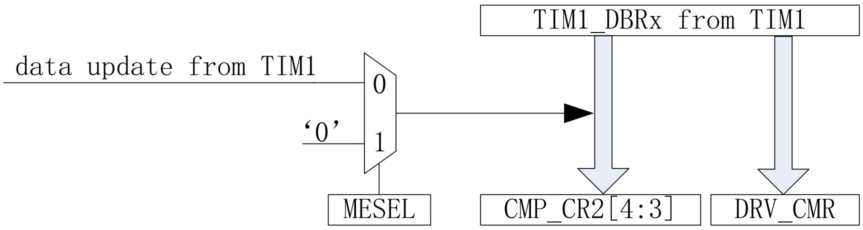


Figure 17-6 TIM1 automatic control DRV_CMR and CMP_CR2[4:3]

With DRV_DR and DRV_ARR register, DRV_CMR register xHE and xLE can achieve pre-charging and braking functions, DRV_DR and DRV_ARR control PWM duty ratio, xHE and xLE control six-way output.

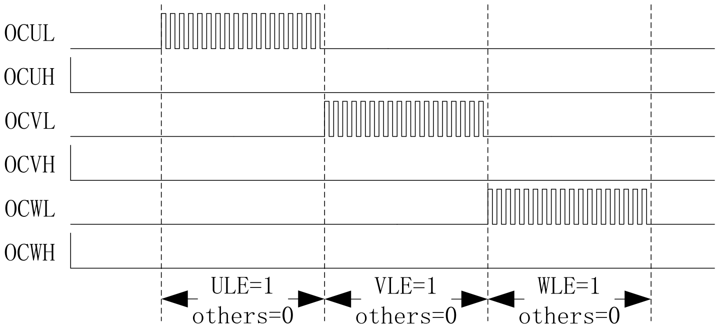


Figure 17-7 Pre-charging diagram

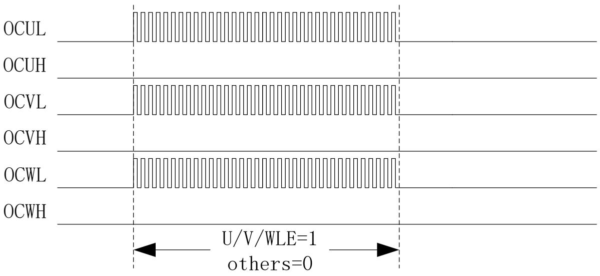


Figure 17-8 Brake diagram

17.1.2.4 Main output enablement MOE

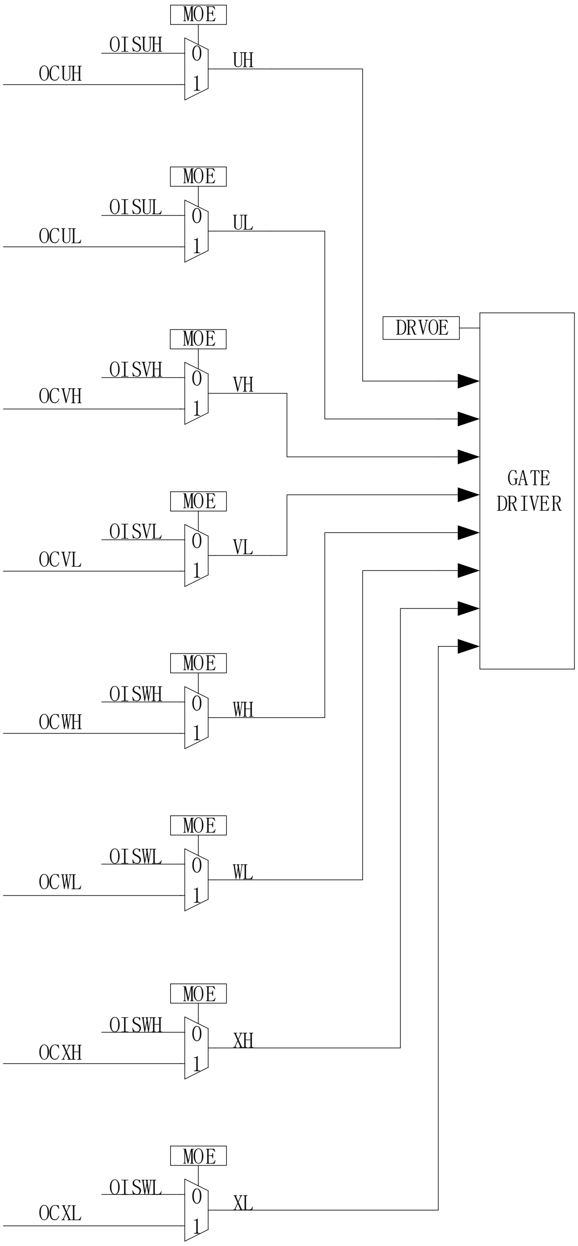


Figure 17-9 output control module after the block diagram

After enabling the MOE, the output comes from the counter comparison value and is used to control the motor output.After MOE is disabled, the output comes from the idle level set by software, which is used to control the motor's non-output, namely, the shutdown state.

17.1.2.5 Interrupt

17.1.2.5.1 Compare-match interrupt

Configure DCIM of DRV_SR register to select compare during eithor counting up or counting down, and configure the comparison match value DRV_COMR. When the count value of the counter is equal to

DRV_COMR, the setting conforming to DCIM will generate an interrupt event, and the interrupt flag DCIF is set by hardware. DCIF can be cleared by software, writing 1 to it is invalid.

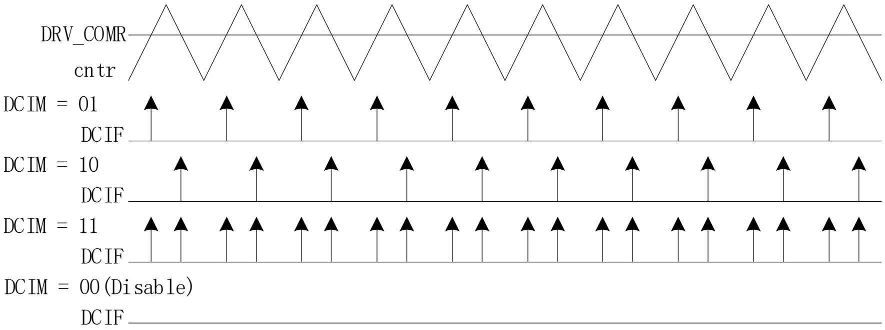


Figure 17-10 DRV compare matching interrupts

17.1.2.5.2 FG interrupt

FGIE of DRV_SR register is set to enable FG interrupt. Every revolution (electrical angle) of the motor will generate an interrupt event. After the interruption event occured, the software needs to be cleared.

17.1.3 PWM output mode (only for FU6812)

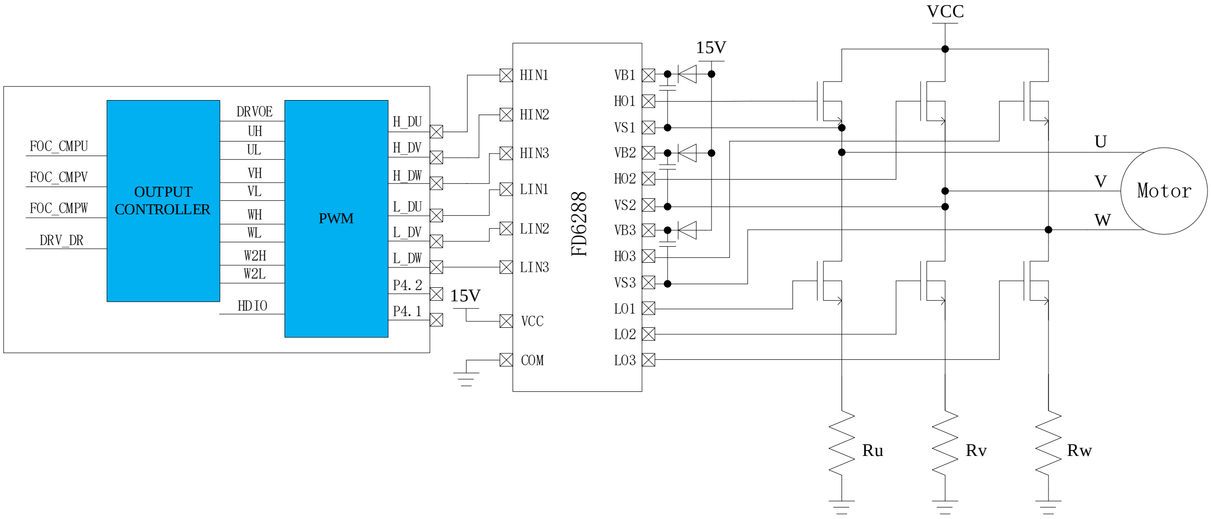


Figure 17-11 Schematic diagram of PWM output mode

FU6812 is the output of PWM, and its functional block diagram is shown in figure 18-11. Entering this mode, DRVOE is the enabling signal of Driver. Different from FU6861's 6N Predriver, the output of PWM connects to HVIC and drives Gate electrode of MOS through HVIC.

17.1.4 6N predriver mode (for FU6861)

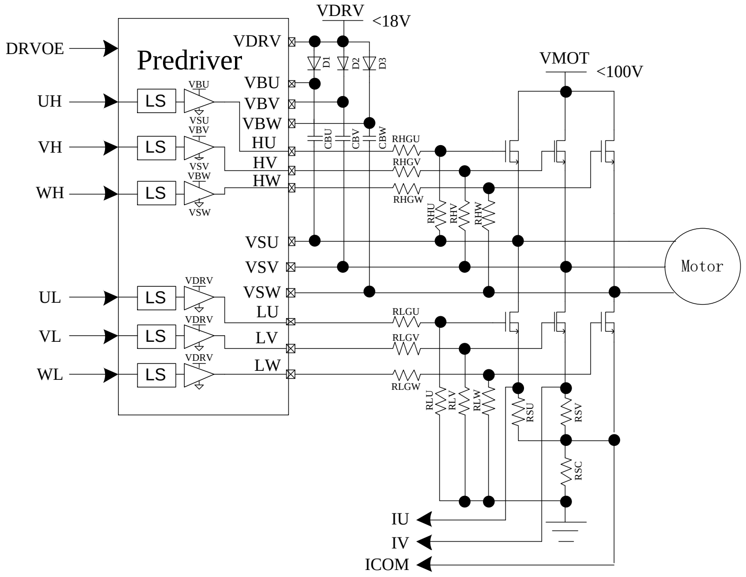


Figure 17-12 schematic diagram of the 6N Predriver mode

The 6N Predriver is as shown in Figure 17-12. Entering this mode, DRVOE is the enabling signal of Predriver, and the output of Predriver connects to 6 NMOS to drive the U/V/W phases of the motor respectively.

Table 17-1 FU6861Q/N/L built-in Predriver singal truth table

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

17.2 Register

17.2.1 DRV_CR (0x4062)

Table 17-2 DRV_CR (0x4062)

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

Bit	Name	Function
[7]	DRVEN	Counter enable 0: Disable 1: Enable

[6]	DDIR	Output direction (forward and backward rotation) Both BLDC and FOC are effective in changing the rotation direction of the motor. If FOC changes this bit, the direction can be changed (sensitive FOC also need to modify angle by software). BLDC without sensors doesn't need to modify TIM1 related parameters, but sensitive BLDC need to change the direction 0: forward 1: reverse
[5]	FOCEN	Enable FOC/SVPWM/SPWM module 0: Disable 1: Enable
[4]	DRPE	Enable DRV_DR preload If enable preload, after software write DRV_DR, the value of the DRV_DR will update after the underflow event. If do not enable preload. After software writes DRV_DR the value is updated immediately 0: Disable 1: Enable
[3]	OCS	Counter compares value source selection 0: DRV_DR 1: FOC/SVPWM/SPWM module
[2]	MESEL	ME module mode selection 0: ME module works in BLDC mode 1. ME module works in FOC/SVPWM/SPWM mode
[1]	RSV	Reserved
[0]	DRVOE	Driver outputs enable 0: Disable 1: Enable

17.2.2 DRV_SR (0x4061)

Table 17-3 DRV_SR (0x4061)

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

Bit	Name	Function
[7]	SYSTIF	SYS TICK interrupt flag The bit is set by hardware.It is cleared by software. 0: no event occurred 1: SYS TICK interrupt generation
[6]	SYSTIE	Enable SYS TICK interrupts SYS TICK interrupts can be generated after enabled

		0: Disable 1: Enable
[5]	FGIF	FG interrupt flag FOC/BLDC generates FGIF flag every circle (electrical cycle) The bit is set to 1 by the hardware.It clears 0 by software. 0: no event occurred 1: FG interrupt is generated
[4]	DCIF	DRV compare match interrupt tags When the count value is equal to DRV_COMR, the counting direction is determined according to DCIM, and the interrupt flag is generated when the count value is equal to DRV_COMR The bit is set by hardware. It is cleared by software. 0: no event occurred 1: comparison interrupt generation
[3]	FGIE	FG interrupts enablement After interruption enabled, FG interrupt flag is generated for every circle (electric cycle) of FOC/BLDC 0: Disable 1: Enable
[2]	DCIP	How many counting cycles to produce an interrupt 0: interruption occurs every time 1: interruption occurs every two times
[1:0]	DCIM	DRV compare match interrupt mode When the count is equal to DRV_COMR, the interrupt flag is determined according to the setting of DCIM 00: no interruption 01: upward direction 10: downward direction 11: upward/downward direction

17.2.3 DRV_OUT (0xF8)

Table 17-4 DRV_OUT (0xF8)

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

Bit	Name	Function
[7]	MOE	Main output enablement This bit is used to select the source of UVWX upper and lower bridge output.The bit can be set and be cleared by software. When the bus current protection is generated (see 27.1.1.1), the hardware will be cleared and close the output. 0: disable, output comes from free levels, OISUH/OISVH/OISWH and

		OISUL /OISVL/OISWL 1: enable the output from the counter comparison value
[6]	RSV	Reserved
[5]	OISWXL	The output idle level of WL and XL Refer to OISUH description
[4]	OISWXH	The output idle level of WH and XH Refer to OISUH description
[3]	OISVL	The output idle level of VL. Refer to OISUH description
[2]	OISVH	The output idle level of VH Refer to OISUH description
[1]	OISUL	The output idle level of UL Refer to OISUH description
[0]	OISUH	The output idle level of UH This bit sets the idle level of output UH.When MOE=0, output the idle level and turn off the corresponding MOS. 0: low level 1: high level

17.2.4 DRV_CMR (0x405C,0x405D)

Note: BLDC control mode, through TIMER1 will automatically control the DRV_CMR register.

Table 17-5 DRV_CMRH (0x405C)

Bit	7	6	5	4	3	2	1	0
Name	XHP	XLP	XHE	XLE	WHP	WLP	VHP	VLP
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 17-6 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	Function
[15]	XHP	X phase upper bridge output polarity 0: high level effective 1: low level effective
[14]	XLP	X phase lower bridge output polarity 0: high level effective 1: low level effective
[13]	XHE	X phase upper bridge output enablement 0: close -- disable output 1: enable -- enable output Note: when XLE and XHE are equal to 1 at the same time, X phase upper and

		lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted
[12]	XLE	X phase lower bridge output enablement 0: close -- disable output 1: enable -- enable output Note: when XLE and XHE are equal to 1 at the same time, X phase upper and lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted
[11]	WHP	W phase upper bridge output polarity 0: high level effective 1: low level effective
[10]	WLP	W phase lower bridge output polarity 0: high level effective 1: low level effective
[9]	VHP	V phase upper bridge output polarity 0: high level effective 1: low level effective
[8]	VLP	V phase lower bridge output polarity 0: high level effective 1: low level effective
[7]	UHP	U phase upper bridge output polarity 0: high level effective 1: low level effective
[6]	ULP	U phase lower bridge output polarity 0: high level effective 1: low level effective
[5]	WHE	W phase upper bridge output enablement 0: disable -- disable output 1: enable -- enable output Note: when WLE and WHE are equal to 1 at the same time, W phase upper and lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted
[4]	WLE	W phase lower bridge output enablement 0: disable -- disable output 1: enable -- enable output Note: when WLE and WHE are equal to 1 at the same time, W phase upper and lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted
[3]	VHE	V phase upper bridge output enablement 0: disable -- disable output

		1: enable -- enable output Note: when VLE and VHE are equal to 1 at the same time, V phase upper and lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted
[2]	VLE	V phase lower bridge output enablement 0: disable -- disable output 1: enable -- enable output Note: when VLE and VHE are equal to 1 at the same time, V phase upper and lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted
[1]	UHE	U phase upper bridge output enablement 0: disable -- disable output 1: enable -- enable output Note: when ULE and UHE are equal to 1 at the same time, U phase upper and lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted
[0]	ULE	U phase lower bridge output enablement 0: disable -- disable output 1: enable -- enable output Note: when ULE and UHE are equal to 1 at the same time, U phase upper and lower bridges complement the output, and the output is automatically inserted with the dead zone.Outputing PWM takes lower bridge for reference, when the tubes of the same phase complement the output, the upper bridge PWM is inverted

17.2.5 DRV_ARR (0x405E,0x405F)

Table 17-7 DRV_ARRH (0x405E)

Bit	7	6	5	4	3	2	1	0
Name	DRV_ARR[11:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 17-8 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	Function
[11:0]	DRV_ARR	Counter reload value that determines carrier period and operation period (central alignment mode) The DRV counter counts from 0 to DRV_ARR, generates an overflow event, and

		then counts down to 0. The calculation formula is $f_{Carrier} = f_{mcu}/2/(DRV\_ARR)$ The value range is (0,4095)

17.2.6 DRV_COMR (0x405A,0x405B)

Table 17-9 DRV_COMRH (0x405A)

Bit	7	6	5	4	3	2	1	0
Name	DRV_COMR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 17-10 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	Function
[11:0]	DRV_COMR	The compare-match value of the counter, when the count value is equal to COMR, whether generate a compare match event based on DCIM of the DRV_SR. The value range is (0,4095)

17.2.7 DRV_DR (0x4058,0x4059)

Table 17-11 DRV_DRH (0x4058)

Bit	7	6	5	4	3	2	1	0
Name	DRV_DR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table 17-12 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	Function
[11:0]	DRV_DR	Software write PWM duty cycle When OCS=0 of DRV_CR, DRV_CNTR is compared with DRV_DR to output PWM.When DRV_CNTR is less than DRV_DR, output 1; otherwise, output 0. Note: when this register is used as the comparison source, the upper bridge is the reference for PWM output, and the lower PWM bridge is inversely output when the same phase tubes are complement the output

		The value range is (0,4095)

17.2.8 DRV_DTR (0x4060)

Table 17-13 DRV_DTR (0x4060)

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

Bit	Name	Function
[7:0]	DRV_DTR	Deadtime DTR is the dead zone duration between inserted complementary outputs.Suppose the MCU clock is 24MHz (41.67ns) DT = (DTR + 1) x 41.67 ns Note: when DTR=0, dead zone is not inserted

18 WDT

Watchdog timer is a timer working in the LS_OSC (interval clock) clock domain, mainly used to monitor the main program running, prevent the MCU crash.How the watchdog works: after the watchdog is started, the watchdog timer starts to run. When the watchdog timer overflows, the watchdog will send a signal to reset the MCU and the main program will run again.Therefore, in the process of running the main program, the watchdog timer should be initialized every once in a while to prevent the watchdog timer overflow, commonly known as "feed dog".

The watchdog of MCU will start timing from 0 after starting. If there is no operation of "feed dog", when the timing reaches FFFC, the watchdog will output a signal with a length of 4 LS_OSC cycles to reset MCU and restart the program.If the program sends the "feed dog" signal to the watchdog regularly during operation, the watchdog timer will count from the initial value set and the watchdog will not reset MCU.

18.1 Notes for using WDT

1、 When MCU enters standby mode or sleep mode, WDT will stop counting, but the count value will remain.

2、 During simulation, WDT will be automatically disabled

3、 WDT overflow resets MCU, then RST_SR[RSTWDT] will be set。

18.2 WDT operation instructions

1、 Configure CCFG1[WDTEN] to start the watchdog, and the watchdog will start counting after the start;

2、 Set WDT_REL, this step can also be used before starting the watchdog;

3、 Set WDT_CR[WDTRF] to initialize the watchdog timer while running the program

18.3 WDT register

18.3.1 WDT_CR (0x4026)

Table 18-1 WDT_CSR (0x4026)

Bit	7	6	5	4	3	2	1	0
Name	RSV						WDTF	WDTRF
Type	R	R	R	R	R	R	R/W0	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:2]	RSV	Reserved
[1]	WDTF	Watchdog reset sign
[0]	WDTRF	1: initialize the watchdog counter; 0: not initialized

18.3.2 WDT_REL (0x4027)

Table 18-2 WDT_REL (0x4027)

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

Bit	Name	Function
[7:0]	WDT_REL	Sets the higher 8 bits of the value of the watchdog counter after reset

19 IO

19.1 IO operation instructions

1、 Port P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.2 map to port registers P0, P1, P2, P3

2、 P0_OE, P1_OE, P2_OE, P3_OE and P4_OE are used to enable the output of P0.0~P4.2

3、 P0.0~P4.2 can enable pull-up resistor by setting the corresponding bits of P0_PU, P1_PU, P2_PU, P3_PU and P4_PU. The pull-up resistance of P0.0~P0.2, P1.3~P1.6, P2.1, P3.6~3.7 is about 5kΩ, pull-up resistance of the rest of the pin is about 33kΩ.The pull-up resistance of P1.5~1.3 can be independently enabled by P1_PU[5:3], and the pull-up resistance will be automatically disable after the rest IO are configured to analog IO

4、 P1.4~P1.7, P2.0~P2.7, P3.0~P3.5 can be configured as analog IO by setting the corresponding bits of P1_AN, P2_AN, P3_AN. After the pin is configured as analog IO, all digital function configuration of corresponding pin fails, and reading corresponding bits of port registers P1, P2 and P3 will get 0.

5、 IO priority:

a) GPIO has the lowest priority for all multiplexed ports

b) P0.1: I2C > TIMER4 > DBG_SIG > GPIO

c) P0.5: SPI > UART > GPIO

d) P0.6: SPI > UART > GPIO

e) P0.7: TIMER2 > CMP > SPI >GPIO

19.2 IO register

19.2.1 P0_OE (0xFC)

Table 19-1 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	Function
[7:0]	P0_OE	Enable P0.0~P0.7 digital output 0: Disable, used for digital input 1: Enable, used for digital output

19.2.2 P1_OE (0xFD)

Table 19-2 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	Function
[7:0]	P1_OE	Enable P1.0~P1.7 digital output 0: Disable, used for digital input 1: Enable, used for digital output

19.2.3 P2_OE (0xFE)

Table 19-3 P2_OE (0xFE)

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

Bit	Name	Function
[7:0]	P2_OE	Enable P2.0~P2.7 digital output 0: Disable, used for digital input 1: Enable, used for digital output

19.2.4 P3_OE (0xFF)

Table 19-4 P3_OE (0xFF)

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

Bit	Name	Function
[7:0]	P3_OE	Enable P3.0~P3.7 digital output 0: Disable, used for digital input 1: Enable, used for digital output

19.2.5 P4_OE (0xE9)

Table 19-5 P4_OE (0xE9)

Bit	7	6	5	4	3	2	1	0
Name	RSV					P4_OE[2]	P4_OE[1]	P4_OE[0]
Type	R					R/W	R/W	R/W
Reset	0					0	0	0

Bit	Name	Function
[7:3]	RSV	Reserved
[2:0]	P4_OE	P4.0~P4.2 digital output enable 0: Disable, used for digital input 1: Enable, used for digital output

19.2.6 P1_AN (0x4050)

Table 19-6 P1_AN (0x4050)

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

<table>
<tr><th>Bit</th><th>Name</th><th>Function</th></tr>
<tr><td>[7:4]</td><td>P1_AN</td><td>Enable P1.7~P1.4 analog mode
0: Disable
1: Enable</td></tr>
<tr><td>[3]</td><td>HBMOD</td><td>The mode configuration of P1.3, determines the functional mode of P1.3 combined with P1_OE.3.
<table>
<tr><td>HBMODE</td><td>P1_OE. 3</td><td>P1.3 mode</td></tr>
<tr><td>0</td><td>0</td><td>Digital input</td></tr>
<tr><td>0</td><td>1</td><td>Digital output</td></tr>
<tr><td>1</td><td>0</td><td>Analog mode</td></tr>
<tr><td>1</td><td>1</td><td>Digital strong drive output, if output high level, it can provide strong drive, if output low level it’s drive ability the same as with '01', digital output mode.</td></tr>
</table></td></tr>
<tr><td>[2]</td><td>HDIO</td><td>The driving ability selection of PWM output, only valid for FU6812’s L_DU, L_DV, L_DW, H_DU, H_DV, H_DW.
0: normal driving ability
1. High driving ability</td></tr>
<tr><td>[1]</td><td>ODE1</td><td>P0.1 open drain enablement
0: Disable
1: Enable</td></tr>
<tr><td>[0]</td><td>ODE0</td><td>P0.0 open drain enablement
0: Disable
1: Enable</td></tr>
</table>

19.2.7 P2_AN (0x4051)

Table 19-7 P2_AN (0x4051)

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

Bit	Name	Function
[7:0]	P2_AN	P2.7~P2.0 analog mode enablement 1: Enable

		0: Disable

19.2.8 P3_AN (0x4052)

Table 19-8 P3_AN (0x4052)

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

Bit	Name	Function
[7:6]	RSV	Reserved
[5:0]	P3_AN	P3.5~P3.0 analog mode enablement 1: Enable 0: Disable

19.2.9 P0_PU (0x4053)

Table 19-9 P0_PU (0x4053)

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

Bit	Name	Function
[7:0]	P0_PU	Enable the pull-up resistance P0.7~P0.0 1: Enable 0: Disable

19.2.10 P1_PU (0x4054)

Table 19-10 P1_PU (0x4054)

Bit	7	6	5	4	3	2	1	0
Name	P1_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:0]	P1_PU	Enable the pull-up resistance of P1.7~P1.0 1: Enable 0: Disable

19.2.11 P2_PU (0x4055)

Table 19-11 P2_PU (0x4055)

Bit	7	6	5	4	3	2	1	0

Name	P2_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:0]	P2_PU	Enable pull-up resistance of P2.7~P2.0 1: Enable 0: Disable

19.2.12 P3_PU (0x4056)

Table 19-12 P3_PU (0x4056)

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

Bit	Name	Function
[7:0]	P3_PU	Enable the pull-up resistance of P3.7~P3.0 1: Enable 0: Disable

19.2.13 P4_PU (0x4057)

Table 19-13 P4_PU (0x4057)

Bit	7	6	5	4	3	2	1	0
Name	RSV					P4_PU[2]	P4_PU[1]	P4_PU[0]
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:3]	RSV	Reserved
[2:0]	P4_PU	Enable pull-up resistance of P4.2~P4.0 1: Enable 0: Disable

19.2.14 PH_SEL (0x404C)

Table 19-14 PH_SEL (0x404C)

Bit	7	6	5	4	3	2	1	0
Name	SPITMOD	UARTEN	UARTCH	T4SEL	T3SEL	T2SEL	T2SSEL	XOE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function

[7]	SPITMOD	SPI slave sender mode 0: as the slave, when SPI is sent and the MISO pin is in the output state. 1: as the slave, after SPI is sent, the MISO pin is in a high resistance state.
[6]	UARTEN	UART enablement 0: Disable 1: Enable
[5]	UARTCH	Enable UART port function transfer 0: UART's port function does not transfer, P0.6 used as the RXD of UART; P0.5 used as the TXD of UART 1: UART port function transfer, P3.3 as the RXD of UART; P3.4 as the TXD of UART
[4]	T4SEL	Enable TIMER4 port 0: P0.1 used as GPIO 1: P0.1 used as the input and output port of TIMER4 Note: the priority of I2C is higher than TIMER4, when enabling I2C, P0.1 used as the port SCL of I2C
[3]	T3SEL	Enable TIMER3 port 0: P1.1 used as GPIO 1: P1.1 used as the input and output port of TIMER3
[2]	T2SEL	Enable TIMER2 port 0: P1.0 used as GPIO 1: P1.0 used as the input and output port of TIMER2
[1]	T2SSEL	Enable TIMER2 port 2 0: P0.7 used as GPIO 1: P0.7 used as the input and output of TIMER2's port 2 Note: TIMER2 has the highest priority, followed by comparator output and SPI's MISO
[0]	XOE	XH/L port enabled 0: P4.2/P4.1 as GPIO 1: P4.2/P4.1 as the XH/XL output, output effective output or idle level OISWH/OISWL according to MOE of the DRVOE register

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

Port output register P0/1/2/3/4 supports read and writing access, RMW (read-modify-write) instruction accesses the value of register (RMW instruction refer to table 20-16, and other instructions access the PORT pin.

Table 19-15 P0/P1/P2/P3/P4

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

Name	Description	R/W	Reset
P0[7:0]	Port register 0	R/W	0x00
P1[7:0]	Port register 1	R/W	0x00
P2[7:0]	Port register 2	R/W	0x00
P3[7:0]	Port register 3	R/W	0x00
P4[2:0]	Port register 4	R/W	0x00

Note: Port P4 has 3 pins, corresponding to P4 output register P4[2:0].

Table 19-16 Read modify write instructions

instruction	Functional description
ANL	Logic AND
ORL	Logic the OR
XRL	Logic exclusive OR
JBC	Jump if bit is set and clear
CPL	Complement bit
INC and DEC	Increment and decrement byte
DJNZ	Decrement and jump if not zero
MOV Px, y, C	Move carry bit to bit y of port x
The CLR Px, y	Clear bit y of port x
SETB Px, y	Set bit y of port x

20 ADC

20.1 ADC function block diagram

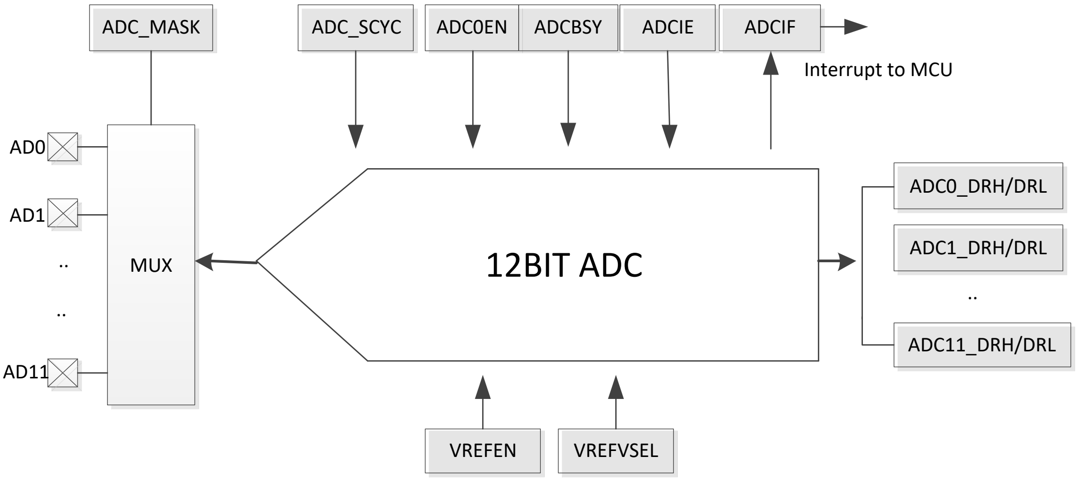


Figure 20-1 ADC function block diagram

20.2 ADC operation instructions

The chip is integrated with a 12-bit suceessive approximation ADC, which supports 12 channels and burst mode sampling.When software writes related startup bit of ADC_CR, the sequential scanning mode will be carried out, and after the FOC function is started, the inner circuit will automatically go into the triggering mode in real time to support the motor drive timing sequence, and the sequential scanning mode will be automatically restored after the completion of the triggering.

20.2.1 Sequential scanning mode

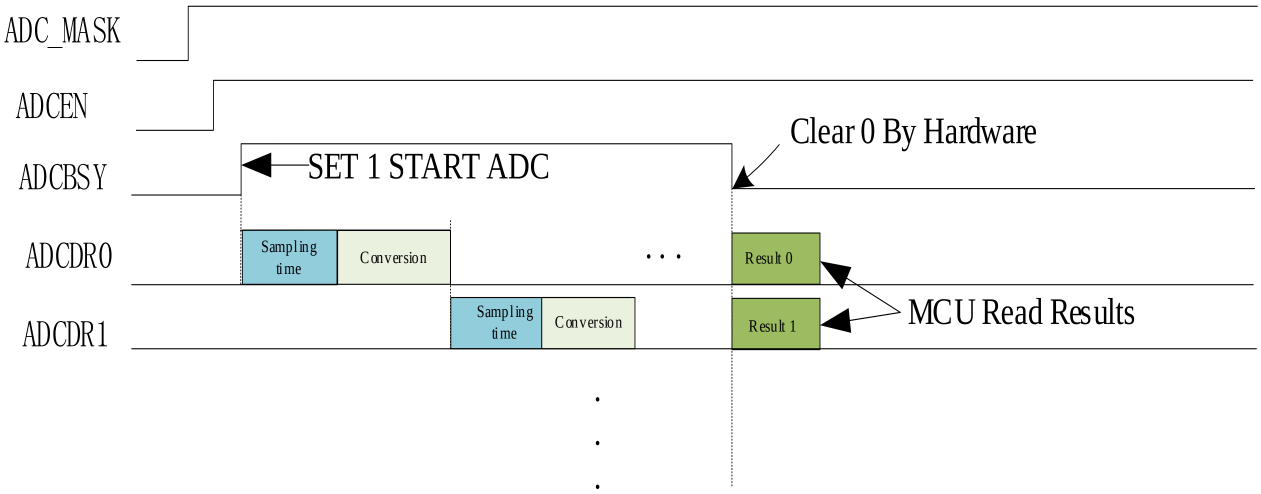


Figure 20-2 ADC sequential scanning timing diagram

As shown in Figure 20-2, start the ADC operation.

- Set the appropriate ADCREF ADC reference voltage.
- Set the corresponding channel, that is, configure register ADC_MASK
- Set the delay of each channel sampling clock cycle (minium is 3).
- Set the ADCEN of register ADC_CR.
- Set the ADCBSY of register ADC_CR, ADC begins to work.
- When read zero from ADCBSY, you can read ADC conversion result.
- ADC conversion sequence according to enabled channel number from low to high (i.e. after enabling CH2/3/4, sample CH2/3/4 and convert successively, then read single conversion results after reading zero from ADCBSY).

There is a built-in 12-bit ADC supporting 12 channel synchronous sampling in chip. Before operating ADC, select the serial number of the corresponding channel by configuring register ADC_MASK, set the appropriate sampling clock cycle delay of each channel (minimum is 3), and set ADCEN and ADCBSY of register ADC_CR, then ADC begin to work.

In addition, the ADC supports trigger function and the priority of trigger function is higher than MCU software operation, the trigger source can be from the FOC module.

An ADC with 12 channels synchronous sampling and 12bit accuracy is integrated into the chip.Before the ADC operation, set the number of the corresponding channel to be sampled, that is, set the register ADC_MASK, set the sampling clock period delay (minimum value is 3) of each channel, set the ADCEN and ADCBSY bits of ADC_STA register to 1, and then the ADC will start to work.

In addition, ADC supports trigger function with higher priority than MCU software operation. The trigger source can be from FOC module.

If FOC function is enabled (FOC_EN=1 of register DRV_CR is configured), the FOC module will automatically start the ADC module and trigger ADC sampling when needed, and send the sampled values to the FOC module automatically.

Note: the trigger function of ADC has a high priority. When the trigger condition meets the need for ADC sampling, if the software is currently operating ADC sampling, the ADC will interrupt the current operation and perform the trigger function. After the sampling of the trigger function, the previous ADC software operation will be automatically resumed.When the ADC trigger function is sampling, if the MCU starts the ADC software operation mode, the ADC first completes the trigger sampling function, and then automatically starts the software operation mode.

20.2.2 ADC trigger mode

After FOC function is started, ADC can automatically trigger ADC sampling when motor drive is required.Triggering mode and sequential scanning mode can be carried out simultaneously. Internal circuit automatically matches the timing sequence of two different modes, but the ADC channel of triggering mode and sequential scanning mode should not overlap.

If FOC function is enabled (FOC_EN=1 of register DRV_CR is configured), the FOC module will automatically start the ADC module and trigger ADC sampling when needed, and send the sampled values to the FOC module automatically.

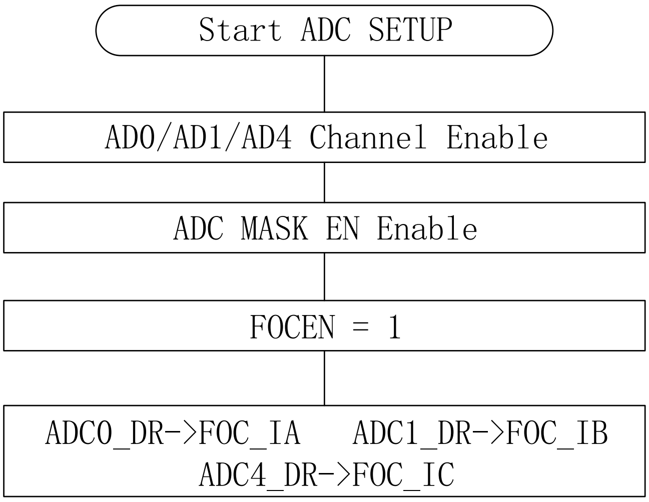


Figure 20-3 ADC trigger sampling timing diagram

As shown in Figure 20-3, after enabling the channel 0/1/4 of ADC and enabling FOC, FOC can trigger sampling and read the result

20.3 ADC register

20.3.1 ADC_CR (0x4039)

Table 20-1 ADC_STA (0x4039)

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

Bit	Name	Function
[7]	ADCEN	Enable ADC function 1: Enable

		0: Disable
[6]	ADCBSY	ADC busy flag. When MCU operates ADC, it should write ADCEN bit first to make the circuit ready, and then write ADCBSY for conversion.MCU write 1 for starting ADC conversion. The bit is cleared automatically by hardware after finished conversion.The MCU can also read this bit to determine if the ADC is in conversion.If this bit is already 1, it is meaningless for MCU to write 1 again.This bit MCU can only write 1, write 0 meaningless, read to represent ADC state. When ADC_MASK is equal to 0, writing 1 to the bit doesn't make sense.
[5]	RSV	Reserved
[1]	ADCIE	ADC interrupt enable.Used to control whether ADCIF initiates an interrupt event to MCU.(TRIG mode interrupt is not included) 0: Disable 1: Enable
[0]	ADCIF	ADC end conversion flag bit. When this ADC conversion is completed, if ADCIE=1, ADC interrupt event will be generated. This bit is not controlled by ADC_ IE. 0: this ADC conversion is not completed. 1: this ADC conversion is completed.

20.3.2 ADC_MASK={ADC_MASKH, ADC_MASKL}(0x4036~0x4037)

Table 20-2 ADC_MASK={ADC_MASKH, ADC_MASKL} (0x4036~0x4037)

ADC_MASKH (0x4036)

Bit	7	6	5	4	3	2	1	0
Name	ADC_SCYC[8]				CH11EN	CH10EN	CH9EN	CH8EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	0	0

ADC_MASKL (0 x4037)

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	Function
[11]	CH11EN	Enable ADC channel 11 Please refer to the ADC_MASK[CH0EN]
[10]	CH10EN	Enable ADC channel 10 Please refer to the ADC_MASK[CH0EN]
[9]	CH9EN	Enable ADC channel 9 Please refer to the ADC_MASK[CH0EN]
[8]	CH8EN	Enable ADC channel 8 Please refer to the ADC_MASK[CH0EN]
[7]	CH7EN	Enable ADC channel 7

		Please refer to the ADC_MASK[CH0EN]
[6]	CH6EN	Enable ADC channel 6 Please refer to the ADC_MASK[CH0EN]
[5]	CH5EN	Enable ADC channel 5 Please refer to the ADC_MASK[CH0EN]
[4]	CH4EN	Enable ADC channel 4 Please refer to the ADC_MASK[CH0EN]
[3]	CH3EN	Enable ADC channel 3 Please refer to the ADC_MASK[CH0EN]
[2]	CH2EN	Enable ADC channel 2 Please refer to the ADC_MASK[CH0EN]
[1]	CH1EN	Enable ADC channel 1 Please refer to the ADC_MASK[CH0EN]
[0]	CH0EN	Enable ADC channel 0 0: disable channel 1: enable channel

20.3.3 ADC_SCYC={ADC_MASKH[7:4],ADC_SCYCL)(0x4036[7:4],0x4038)

Table 20-3 ADC_SCYCL (0x4038)

Bit	7	6	5	4	3	2	1	0
Name	ADC_SCYC[7:4]				DC_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	Function
[11:8]	ADC_SCYC[11:8]	The sampling period setting of ADC, which is shared with channel AD8, AD9, AD10 and AD11 When ADC_SCYC[11] = 0, The number of sampling cycles of channel AD8, AD9, AD10 and AD11 is ADC_SCYC[10:8] When ADC_SCYC[11] = 1, The number of sampling cycles of channel AD8, AD9, AD10 and AD11 is (ADC_SCYC[10:8]*8 + 7).
[7:4]	ADC_SCYC[7:4]	The sampling period setting of ADC, which is shared with channel AD5, AD6, AD7 When ADC_SCYC[7] = 0, The number of sampling cycles of channel AD5, AD6 and AD7 is ADC_SCYC[6:4] When ADC_SCYC[7] = 1, The number of sampling cycles of channel AD5, AD6 and AD7 are (ADC_SCYC[6:4]*8 + 7)
[3:0]	ADC_SCYC[3:0]	The sampling period setting of ADC, which is shared with channel AD0, AD1, AD2, AD3 and AD4 When ADC_SCYC[3] = 0,

		The number of sampling cycles of channel AD0, AD1, AD2, AD3 and AD4 is ADC_SCYC[2:0]. When ADC_SCYC[3] = 1, The number of sampling cycles of channel AD0, AD1, AD2, AD3 and AD4 are (ADC_SCYC[2:0]*8 + 7).

20.3.4 ADC0_DR={ADC0_DRH,ADC0_DRL}(0x0300~0x0301)

Table 20-4 ADC0_DR={ADC0_DRH,ADC0_DRL} (0x0300~0x0301)

ADC0_DRH (0x0300)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC0_DRL (0x0301)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 0
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 0

20.3.5 ADC1_DR={ADC1_DRH,ADC1_DRL}(0x0302~0x0303)

Table 20-5 ADC1_DR={ADC1_DRH, ADC1_DRL}(0x0302~0x0303)

ADC1_DRH (0x0302)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC1_DRL (0x0303)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 1
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 1

20.3.6 ADC2_DR={ADC2_DRH,ADC2_DRL}(0x0304~0x0305)

Table 20-6 ADC2_DR={ADC2_DRH, ADC2_DRL} (0x0304~0x0305)

ADC2_DRH (0x0304)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC2_DRL (0x0305)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 2
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 2

20.3.7 ADC3_DR={ADC3_DRH,ADC3_DRL}(0x0306~0x0307)

Table 20-7 ADC3_DR={ADC3_DRH, ADC3_DRL}(0x0306~0x0307)

ADC3_DRH (0x0306)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC3_DRL (0x0307)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 3
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 3

20.3.8 ADC4_DR={ADC4_DRH,ADC4_DRL}(0x0308~0x0309)

Table 20-8 ADC4_DR={ADC4_DRH, ADC4_DRL} (0x0308~0x0309)

ADC4_DRH (0x0308)

Bit	7	6	5	4	3	2	1	0

Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC4_DRL (0x0309)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of the channel 4
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 4

20.3.9 ADC5_DR={ADC5_DRH,ADC5_DRL}(0x030A~0x030B)

Table 20-9 ADC5_DR={ADC5_DRH, ADC5_DRL}(0x030A~0x030B)

ADC5_DRH (0x030A)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC5_DRL (0x030B)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of the channel 5
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of the channel 5

20.3.10 ADC6_DR={ADC6_DRH,ADC6_DRL}(0x030C~0x030D)

Table 20-10 ADC6_DR={ADC6_DRH, ADC6_DRL} (0x030C~0x030D)

ADC6_DRH (0x030C)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC6_DRL (0x030D)

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

Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 6
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 6

20.3.11 ADC7_DR={ADC7_DRH,ADC7_DRL}(0x030E~0x030F)

Table 20-11 ADC7_DR={ADC7_DRH, ADC7_DRL} (0x030E~0x030F)

ADC7_DRH(0x030E)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC7_DRL (0x030F)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 7
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 7

20.3.12 ADC8_DR={ADC8_DRH,ADC8_DRL}(0x0310~0x0311)

Table 20-12 ADC8_DR={ADC8_DRH, ADC8_DRL} (0x0310~0x0311)

ADC8_DRH (0x0310)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC8_DRL (0x0311)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved

[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 8
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 8

20.3.13 ADC9_DR={ADC9_DRH,ADC9_DRL}(0x0312~0x0313)

Table 20-13 ADC9_DR={ADC9_DRH, ADC9_DRL} (0x0312~0x0313)

ADC9_DRH (0x0312)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC9_DRL (0x0313)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 9
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 9

20.3.14 ADC10_DR={ADC10_DRH,ADC10_DRL}(0x0314~0x0315)

Table 20-14 ADC10_DR={ADC10_DRH, ADC10_DRL} (0x0314~0x0315)

ADC10_DRH (0x0314)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC10_DRL (0x0315)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 10
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 10

20.3.15 ADC11_DR={ADC11_DRH,ADC11_DRL}(0x0316~0x0317)

Table 20-15 ADC11_DR={ADC11_DRH, ADC11_DRL}(0x0316~0x0317)

ADC11_DRH(0x0316)

Bit	7	6	5	4	3	2	1	0
Name	RSV				DH[3]	DH[2]	DH[1]	DH[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

ADC11_DRL(0x0317)

Bit	7	6	5	4	3	2	1	0
Name	DL[7]	DL[6]	DL[5]	DL[4]	DL[3]	DL[2]	DL[1]	DL[0]
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:12]	RSV	Reserved
[11:8]	DH	After the ADC conversion is completed, the higher 4-bit data of channel 10
[7:0]	DL	After the ADC conversion is completed, the lower 8-bit data of channel 10

21 DAC

21.1.1 DAC function block diagram

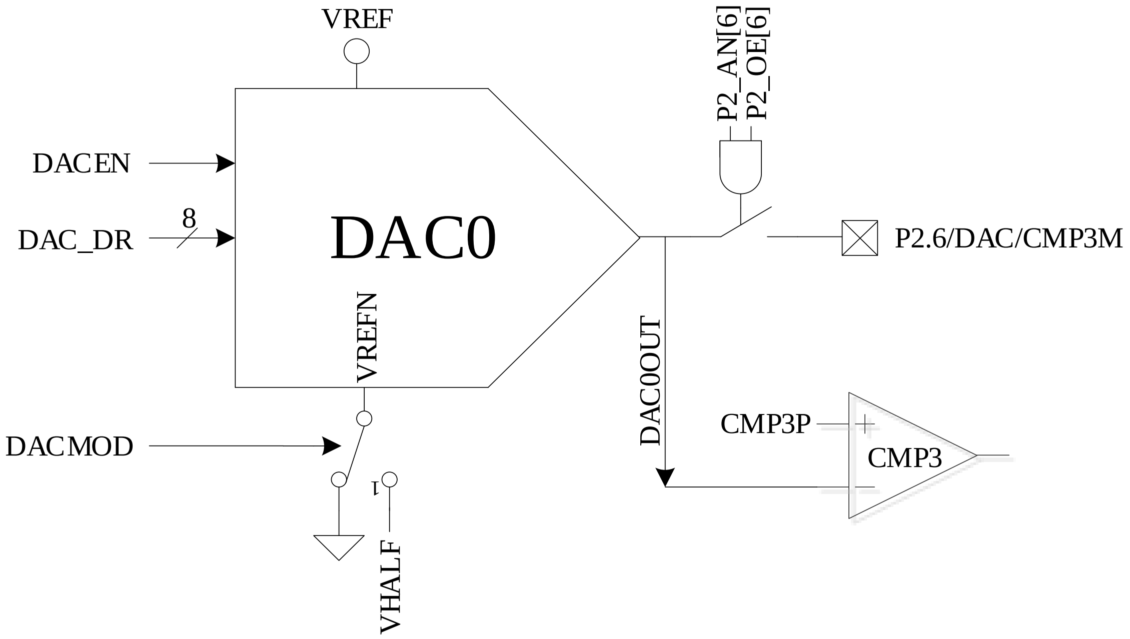


Figure 21-1 DAC function block diagram

Note:

1. DAC0 output has no current driving capacity and can only carry capacitive load. If it needs to be used as resistive load outside the chip, it needs to be followed by operational amplifier voltage for output
2. For DAC0 output to P2.6/DAC pin, P2_AN[6]=1 and P2_OE[6]=1 should be configured
3. DAC0 uses VREF as the reference voltage, enabling DAC by configure VREFEN=1, DACEN=1,

21.1.2 DAC_CR (0x4035)

Table 21-1 DAC_CR (0x4035)

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

Bit	Name	Function
[7]	DACEN	DAC enabled 0: Disable. 1: Enabled.
[6]	DACMOD	DAC mode settings 0: normal mode, the voltage range of DAC output is 0 to VREF. 1: semi-voltage conversion mode, DAC output voltage range is VHALF to VREF.

[5:0]	RSV	Reserved

21.1.3 DAC_DR (0x404B)

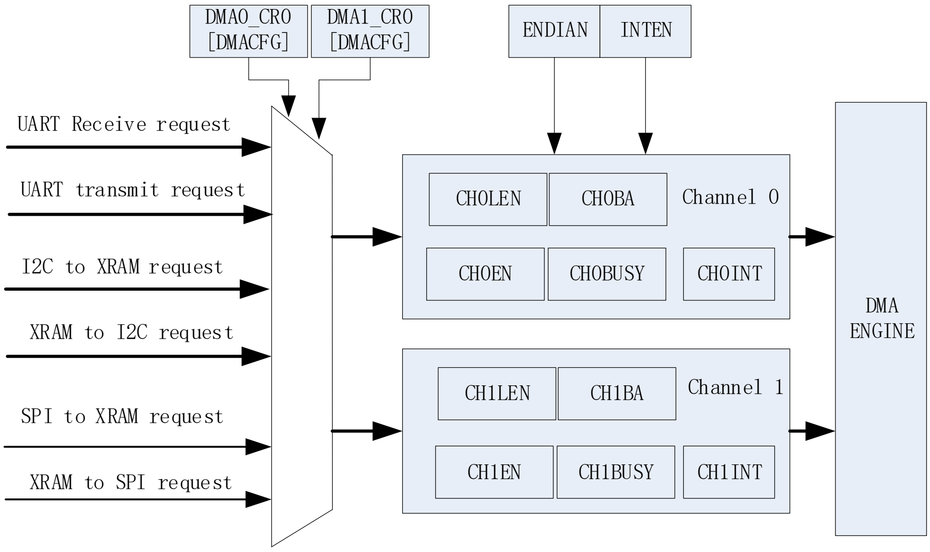
Table 21-2 DAC_DR (0x404B)

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

Bit	Name	Function
[7:0]	DAC_DR	DAC controller data input.

22 DMA

22.1 DMA functions and instructions



DMA sub-module contains a dual-channel DMA controller, which realizes the direct data transfer between peripheral (SPI, UART, I2C) and XRAM. During the transmission, DMA's access to XRAM does not interfere with MCU's normal reading and writing operation of XRAM.The length of the transmission and the starting address of XRAM access can be set, the small end mode or the big end mode is configurable during the transmission, and interrupt enablement is supported.

The operation process of starting DMA is generally as follows: first, the peripheral is configured and enabled; then, according to the requirements, DMA takes over the input and output channels of peripheral through DMAx_CR0[CFG]; set DMA interrupt, transmission length, transmission length and XRAM starting address; then, write DMAx_CR0[EN] and DMAx_CR0[BSY] as 1 to start DMA. After data transmission is completed, the corresponding DMA interrupt can be cleared. When DMA is started again, there is no need to set the configuration bit again. Just write DMAx_CR0[BSY] as 1 to start DMA again.

22.2 DMA register

ADDR	XSFR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0x403A	DMA0_CR0	DAMEN	DMABSY	DMACFG[2:0]			DMA_IE	ENDIAN	DMAIF
0x403B	DMA1_CR0	DMAEN	DMABSY	DMACFG[2:0]			DBG_SW	DBG_EN	DMAIF
0x403C	DMA0_CR1	CH0LEN[5:0]						CH0BA[9:8]	
0x403D		CH0BA[7:0]							
0x403E	DMA1_CR1	CH1LEN[5:0]						CH1BA[9:8]	
0x403F		CH1BA[7:0]							

22.2.1 DMA0_CR0 (0x403A)

Table 22-1 DMA0_CR0 (0x403A)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DMAIE	ENDIAN	DMAIF
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7]	DMAEN	DMA channel 0 enablement
[6]	DMABSY	DMA channel 0 busy state/start Read: 0: the transmission of channel 0 is completed or not in transmission state. 1: the transmission of channel 0 is transmitting. Write: 0: no sense. 1: start channel 0 to start transmission.
[5:3]	DMACFG[2-0]	Channel 0 peripheral selection 000: from UART to XRAM 001: from XRAM to UART. 010: from I2C to XRAM 011: from XRAM to I2C 100: from SPI to XRAM 101: from XRAM to SPI Cannot be changed when channel 0 state is busy.
[2]	DMAIE	DMA interrupt request enablement 0: DMA is prohibited from making interrupt requests to MCU. 1: enable the interrupt request sent to MCU by DMA. When the interrupt flag CH0INT or CH1INT is 1, an interrupt request is issued to MCU.
[1]	ENDIAN	DMA data transfer sequence mode. 0: high byte is received or sent first. 1: the low byte is received or sent first. This bit setting is for 16-bit data mode and should be configured to 0 for 8-bit data mode. Cannot be changed when channel 0 or channel 1 is busy.
[0]	DMAIF	DMA channel 0 transfers completion interrupt flag. It’s set by hardware, it’s cleared by software. 0: no interrupt is generated. 1: channel 0 transmission completion flag. (when INTEN=1, an interrupt request is made to the MCU, otherwise only flags are generated)

22.2.2 DMA1_CR0 (0x403B)

Table 22-2 DMA1_CR0 (0x403B)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DBGSW	DBGEN	DMAIF
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7]	DMAEN	DMA channel 1 enablement
[6]	DMABSY	DMA channel 1 busy state/start Read: 0: channel 1 transmission is completed or not in transmission state. 1: channel 1 is transmitting. Write: 0: nonsense. 1: start channel 1 to start transmission.
[5:3]	DMACFG[2-0]	Channel 1 peripheral selection 000: from UART to XRAM 001: from XRAM to UART. 010: from I2C to XRAM 011: from XRAM to I2C 100: from SPI to XRAM 101: from XRAM to SPI Cannot be changed when channel 1 state is busy.
[2]	DBGSW	DEBUG mode area selection 0: DEBUG area selects XSFR(export address space: 0x4020 ~ 0x40FF) 1:DEBUG area selects XRAM(export address space: 0x0000 ~ 0x0317)
[1]	DBGEN	DEBUG mode enabled 0: normal mode 1: the DEBUG mode When CH1CFG is configured to 101 and DBG_EN=1, DMA will enter DEBUG mode, and when SPI_EN=1, SPI will be 3-line only-send master mode (MISO line is invalid). DMA automatically and repeatedly sends the relevant data of DBG_SW region through SPI MOSI, and CH1BA/CH1LEN is used to specify the address in the region. When sending, NSS automatically becomes low, and after each transmission cycle, NSS automatically becomes high level once, and then continues to send in the next cycle. When you enter DEBUG mode, DMA0CH1 interrupts are automatically turned off.
[0]	DMAIF	DMA channel 1 transfers complete interrupt flag. It’s set by hardware, it’s cleared by software. 0: no interrupt is generated. 1: channel 1 transmission completion flag. (when INTEN=1, an interrupt request is made to the MCU, otherwise only flags are generated)

22.2.3 DMA0_CR1 (0x403C)

Table 22-3 DMA0_CR1H (0x403C)

Bit	7	6	5	4	3	2	1	0
Name	CH0LEN						CH0BA[9:8]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table DMA0_CR1L (0x403D)

Bit	7	6	5	4	3	2	1	0
Name	CH0BA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:10]	CH0LEN[15:10]	Write: the XRAM data transfer length of DMA channel 0. Cannot be changed when channel 0 state is busy. When ENDIAN=1, CH0LEN is recommended to be odd. Read: which byte is currently transferred by DMA channel 0 (0 represents the first byte).
[9:0]	CH0BA[9:0]	The XRAM first address of DMA channel 0. Cannot be changed when channel 0 state is busy. Note that the XRAM address space region transmitted by channel 0 is CH0BA[9:0] ~ (CH0BA[9:0] + CH0LEN[5:0]).

22.2.4 DMA1_CR1 (0x403E)

Table 22-4 DMA1_CR1H (0x403E)

Bit	7	6	5	4	3	2	1	0
Name	CH1LEN						CH1BA[9:8]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Table DMA1_CR1L (0x403F)

Bit	7	6	5	4	3	2	1	0
Name	CH1BA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[15:10]	CH1LEN[15:10]	Write: the XRAM data transfer length of DMA channel 1. Cannot be changed when channel 1 state is busy. When ENDIAN=1, CH1LEN is recommended to be odd. Read: which byte is currently transferred by DMA channel 1 (0 represents the first byte).

[9:0]	CH1BA[9:0]	The XRAM first address of DMA channel 1. Cannot be changed when channel 1 state is busy. Note that the XRAM address space region transmitted by channel 1 is CH1BA[9:0] ~ (CH1BA[9:0] + CH1LEN[5:0]).

When DMA channel peripherals are selected as I2C (including I2C to XRAM and XRAM to I2C), the START+Address interrupt of I2C communication still needs to be cleared by software.When I2C is slave machine, if STOP is encountered, the software needs to clear register I2C_SR[STOP] to clear the I2C interrupt, and DMA transfer should be restarted.

23 VREF

23.1 Operation instructions for the VREF module

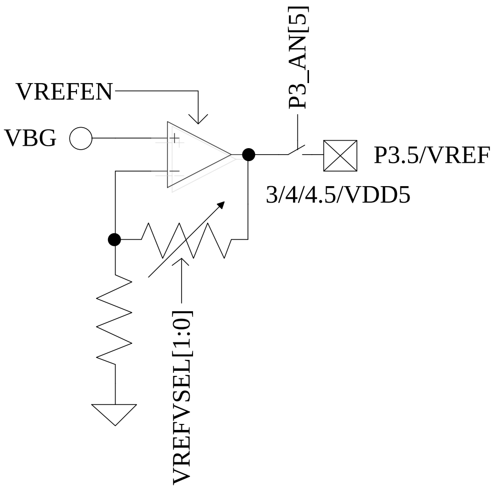


Figure 23-1 Port input and output of the VREF module

The port condition of the VREF module is shown in Figure 23-1. VREF is a reference voltage generation module that provides an internal reference to the ADC.

To make the VREF work, the registers need to be configured as follows: VREFEN=1, VREFVSEL selects the output voltage, see Table 23-1 for details. If VREF voltage needs to be output to chip pin, configure P3_AN[5]=1.VREF voltage is used for the reference voltage of ADC and DAC in the chip. (Note: FU6812S can only use VDD5 as reference voltage.

23.2 Registers in the VREF module

23.2.1 VREF_VHALF_CR (XRAM: 0x404F)

Table 23-1 VREF_CR (0x404F)

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

Bit	Name	Function
[7:6]	VREFVSEL	VREF module output reference voltage selection terminal 01: VDD5 00: 4.5 V 11: 4V 10: 3V
5	RSV	Reserved
4	VREFEN	The VREF module enable signals used for providing an internal reference to the ADC 0: Disable internal VREF references, such as setting P3_AN[5]=1, and external references input from P3.5 1: Enable internal VREF reference, such as setting P3_AN[5]=1, ouput internal VREF reference to P3.5 pin, connect 0.1~1uF capacitance to improve VREF stability
3:1	RSV	Reserved
0	VHALFEN	Enable VHALF 0: Disable 1: Enable

24 VHALF

24.1 VHALF module operation instructions

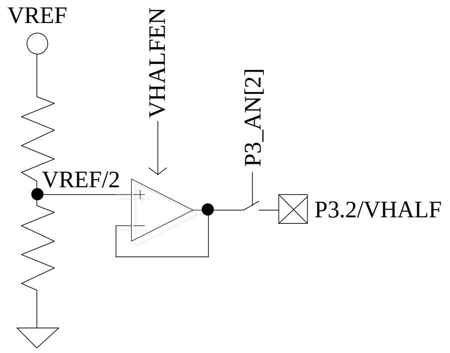


Figure 24-1 Port input and output of the VHALF module

Figure 24-1 shows the ports of the VHALF module.The role of VHALF is to generate a reference voltage.

To make the VHALF module work properly, configure the register as follows: VHALFEN=1;P3_AN[2]=1, output VHALF voltage to P3.2 pin.

24.2 Register of the VHALF module

The VHALF configuration register refer to Table 23-1 .

25 Op amp

FU6812L/61Q integrate three high-speed independent operational amplifiers (op amps), AMP0, AMP1 and AMP2. Each op amp has a separate enable bit. FU6812N/S、FU6861N integrate only one op amp AMP0.

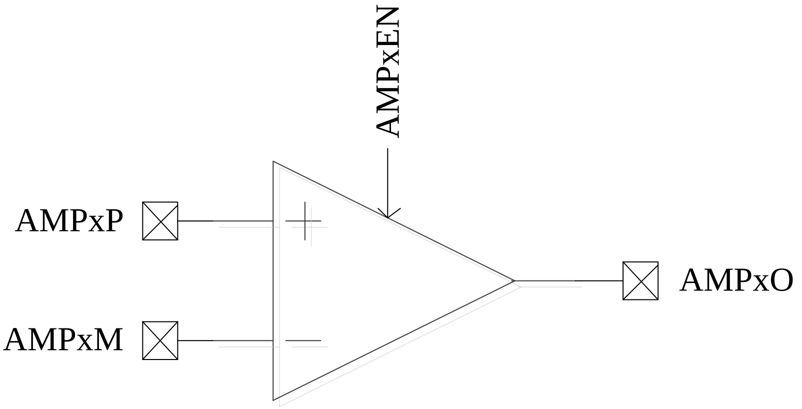


Figure 25-1 Schematic diagram of operation amplification module

25.1 Op amp operation instructions

25.1.1 Bus current op amp（AMP0）

The AMP0 connection is shown in Figure 25-2.

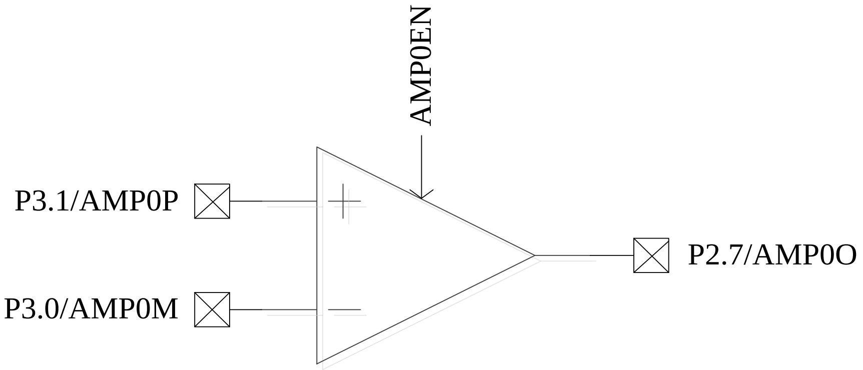


Figure 25-2 Bus current operation (AMP0)

For enabling functioning of the block, the AMP0EN bit must be set to 1.

The pins corresponding to the input and output terminals of the bus current operational amplifier are shown in Figure 25-2. Before enabling AMP0, all three GPIO ports related to AMP0 should be turned into analog mode, namely P2.7, P3.0 and P3.1, and P2_AN[7]=1 and P3_AN[1:0]=11B.

25.1.2 Phase current op amp (AMP1/AMP2)

25.1.2.1 AMP1

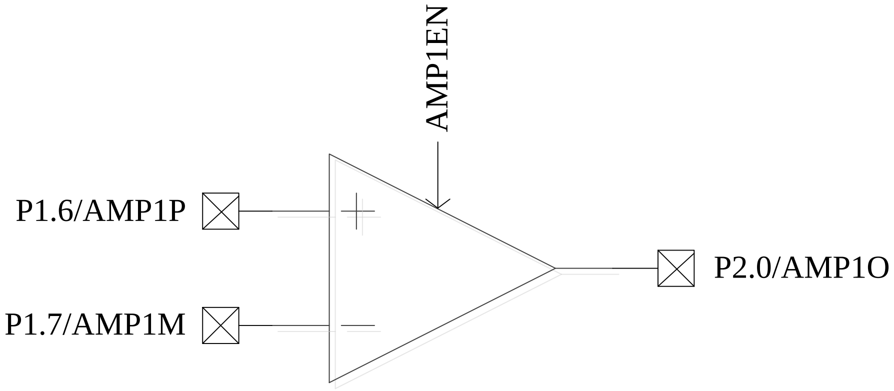


Figure 25-3 AMP1 input/output related pins

If enabling phase current AMP1, AMP1EN=1 must be configured.

The pin corresponding to the input and output end of the phase current operational amplifier is shown in Figure 25-3. Before enabling AMP1 amp, all three GPIO ports related to AMP1 amp should be converted into analog mode, namely P1.6, P1.7 and P2.0, and P1_AN[7:6]=11B and P2_AN[0]=1B.

25.1.2.2 AMP2

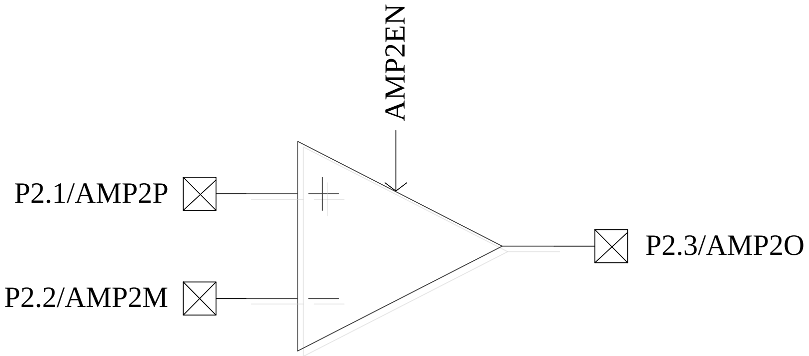


Figure 25-4 AMP2 input/output related pad

AMP2EN=1 is required if AMP2 is to be enabled.

The pin corresponding to the input and output terminals of the phase current operational amplifier is shown in Figure 25-4. Before enabling AMP2 op amp, all three GPIO ports related to this op amp should be made into analog mode, that is, P2.1, P2.2 and P2.3 should be made into analog mode, and P2_AN[3:1]=111B.

25.2 Op amp register

25.2.1 AMP_CR (0x404E)

Table 25-1 AMP_CR (0x404E)

Bit	7	6	5	4	3	2	1	0
Name	RSV					AMP2EN	AMP1EN	AMP0EN
Type	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:3]	RSV	Reserved
[2]	AMP2EN	Enable AMP2
[1]	AMP1EN	Enable AMP1
[0]	AMP0EN	Enable AMP0

26 Comparator

26.1 Comparator operation instructions

26.1.1 Comparator CMP3

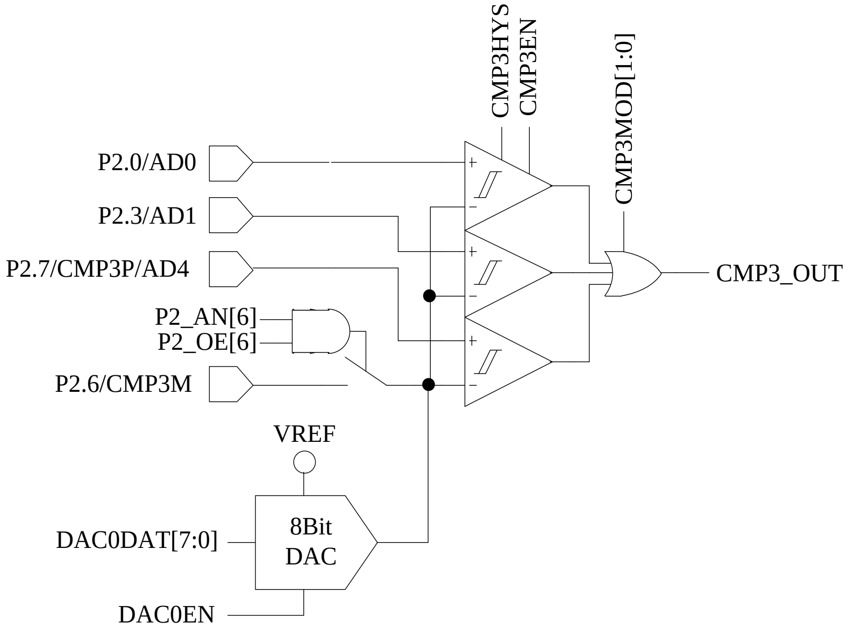


Figure 26-1 Input and output related signals of comparator CMP3

If you want to enable the comparator CMP3, you need to configure:

1. Configure CMP3MOD and select one of single input, double input and three input modes
2. According to the configuration of 1, select the corresponding pin as the analog mode, Px_AN[x]=1, such as single input mode: configure P2_AN[7]=1
3. With negative input reference voltage, it can come from the DAC output inside the chip or from the outside of the chip.If from DAC, enable DAC, refer to the section about DAC. If from outside the chip, configure P2_AN[6]=1, P2_OE[6]=1, input reference voltage from P2.6 pin
4. Configure CMP3HYS to select whether to use hysteresis
5. Enable comparator 3 by set CMP_CR2 register CMP3EN.

The pins corresponding to the input and output terminals of the comparator are shown in Figure 26-1. CMP3HYS controls the hysteresis voltage of the comparator CMP3.There are three input modes for CMP3, which are determined by CMP3MOD[1:0].

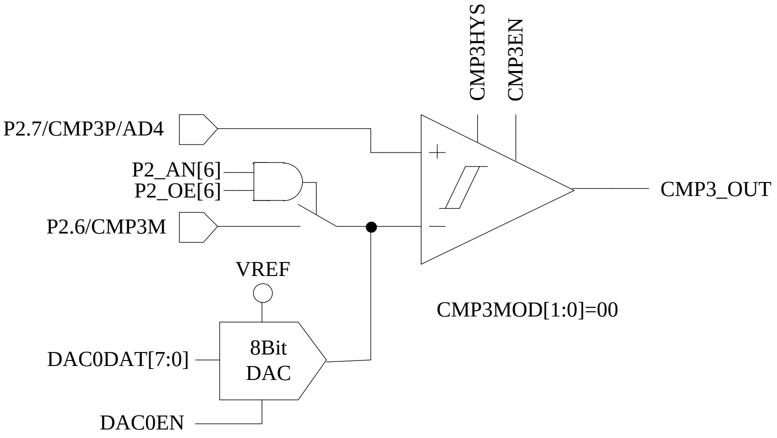


Figure 26-2 CMP3MOD[1:0]=00B, select the single comparator input mode

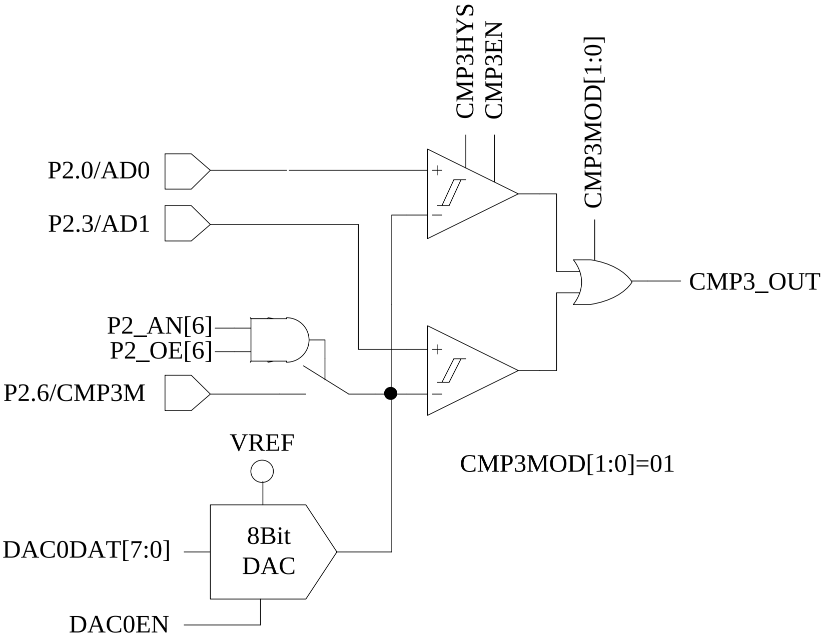


Figure 26-3 CMP3MOD[1:0]=01B, select double comparators input mode

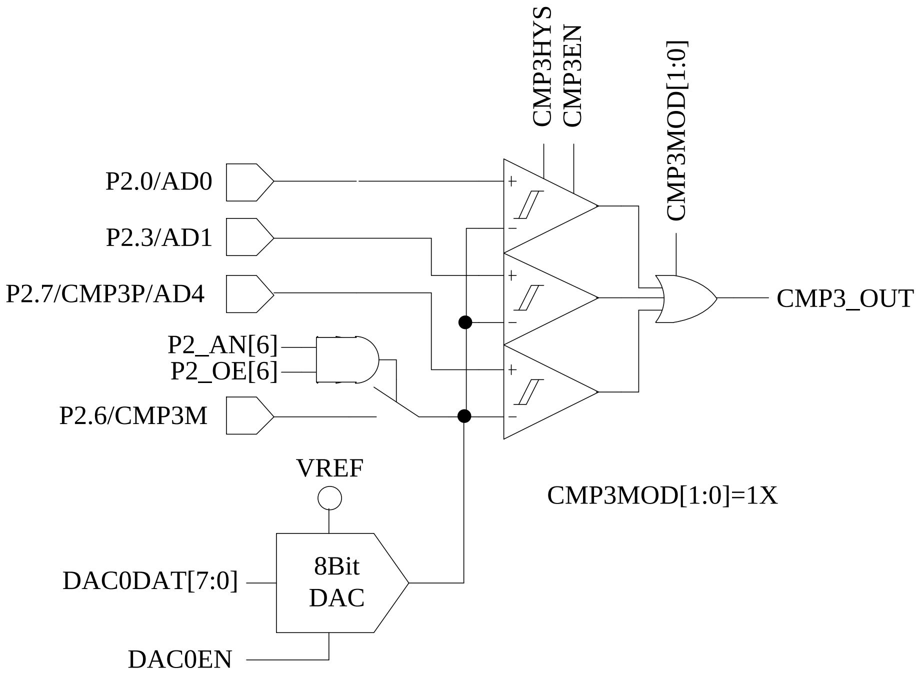


Figure 26-4 CMP3MOD[1:0]=1XB, select three comparators input mode

26.1.1.1 Bus current protection

Bus current protection is based on the generation of bus current protection signal, automatic shutdown motor output, in order to achieve the purpose of protecting the chip and motor. MOEMD=01 of the EVT_FILT register is configured to generate a protection event, close the output automatically and generate protection interruption. Configure the MOEMD=00 of the EVT_FILT register to generate a protection event that will not automatically close the output, but will generate a protection interrupt.

According to the EVT_FILT register's EFSRC, bus current protection events can be generated by comparator CMP3 or by external interrupt INT0(P0.0).When the chip is externally connected to IPM module to control the motor, FALUT signal of IPM is connected to P0.0. If configure EFSRC=1 of EVT_FILT register, and bus current protection signal is generated through external interrupt INT0, at which time the protection interrupt is external interrupt INT0. If configure EFSRC=0 of the EVT_FILT register. The bus current protection signal is generated by the comparator CMP3. By comparing the sampled voltage on the bus, generate the protection signal at which time the protection interrupt is CMP3's interrupt.

The input signal of bus current protection event can be filtered by configuring EVT_FILT register's EFDIV!=0, select one of filtering width of 4/8/16 clock cycles by configuring EFDIV=01/10/11 of EVT_FILT. After enabling filtering function, the filtered signal will delay about 4~5/8~9/16~17 clock cycles compared with the signal before filtering.

When CMP3 used for bus current protection, cooperate internal module to close directly the output to motor, to achieve protection.

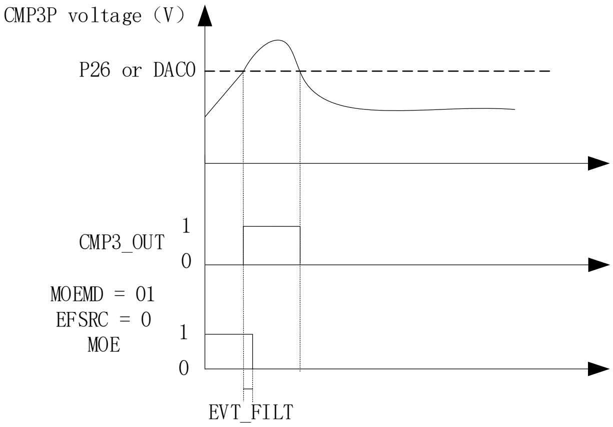


Figure 26-5 Diagram of clearing MOE when the bus current protection is generated

According to CMP3IM, generate the interrupt during rising edges. If EVT_FILT’s MOEMD=01, produce protection events, shut down output automatically and generate the interrupt for protectiong. If EVT_FILT’s MOEMD=0, bus protection events is generated by CMP3. By comparing the sampling voltage value of bus, produce protection signals. Configuration EVT_FILT register’s EFDIV=01/10/11 to select filter width from one of 4/8/16 clock cycles, filtered signal delays about 4~5/8~9/16~17 clock cycles compared with the signal before filter. As shown in figure 27-5, when the comparator is the input voltage of the positive terminal voltage is higher than the negative input, CMP3_OUT1 is set, produce CMP3 compare interrupt, hardware automatically clear MOE after EFDIV filtering (MOE), 6 outputs are shut off, so as to protect.

26.1.1.2 Cycle-by-cycle current limit

Cycle-by-cycle current limit is mainly applied to square wave control of BLDC. MOEMD=10 of EVT_FILT register is configured to generate the protection event, close automatically the output and automatically enable MOE of overflow event of DRV counter. MOEMD=11 of EVT_FILT register is configured to generate the protection event and close automatically the output, and MOE is set automatically every 5us and at overflow and underflow event of DRV counter.

Cycle-by-cycle current limit requires the CMP3IM of CMP_CR0 register to be configured, and the interrupt of CMP3 will be generated at the same time. If no interrupt is needed, the interrupt priority of CMP3 can be set to the lowest, and an empty CMP3 interrupt processing function can be written at the same time.

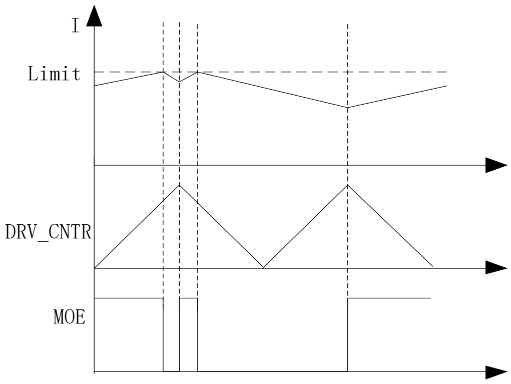


Figure 26-6 Cycle-by-cycle in MOEMD=10 mode

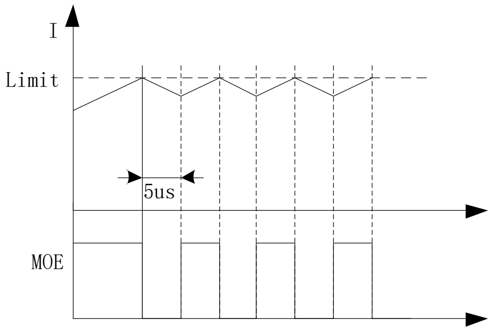


Figure 26-7 Cycle-by-cycle in MOEMD=11 mode

26.1.2 Comparator CMP4

Comparator CMP4 is a hysteresis comparator. Refer to Figure 26-8,judge whether the flip of CMP4 is triggered only by external interrupt 0. Select one of comparators's output according CMPSEL. When CMP4 is enabled, CMP3MOD[1:0] cannot be 01B. CMP4 is generally not used alone, with CMP3 to rechive cycle-by-cycle current limit at BLDC square wave control.

To enable CMP4, the usage of configuring CMP4 is as follows:

1. P2_AN[3]=1, P2_AN[7]=1, P2.3/CMP4P and P2.7/CMP4M pins are configured as analog mode
2. CMP4EN=1, enable CMP4
3. Configure LVSR[EXT0CFG] to use CMP4 to trigger an external interrupt INT0
4. Clear the external interrupt INT0 flag and set the trigger edge of INT0 to enable the external interrupt INT0.
5. The external interrupt INT0 is triggered when the comparator output is flipped from 1 to 0

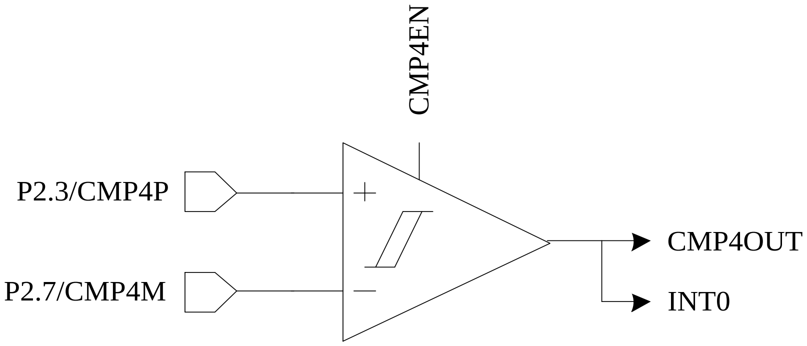


Figure 26-8 Schematic diagram of CMP4 module

26.1.3 Comparator CMP0

The comparator CMP0 has a variety of comparison modes, which can automatically enable CMP0, CMP1 and CMP2 according to the configuration, and can be used for real-time detection of the position and speed of the motor rotor.

CMP0OUT~CMP2OUT of 3-channel output signal of the comparator CMP0 first passes TIM1 filtering and sampling module, so TIM1_CR3[T1TIS]=01B should be ensured before using CMP0. The CMP_SR result about CMP0 is generated based on CMP0, otherwise the CMP_SR result about CMP0 is generated based on HALL signal, refer 13.1.2

CMP0MOD[1:0]=00B, select no built-in resistance 3-comparator mode.Refer to Figure 26-9, which can be used for motor BEMF detection if off-chip virtual central point resistors exist. The negative input end is fixed to P1.5/CMP0M, and the positive input end is P1.4/CMP0P, P1.6/CMP1P, P2.1/CMP2P. The corresponding output is sent to CMP0OUT, CMP1OUT and CMP2OUT,

No built-in resistance 3-comparator mode register configuration:

1. TIM1_CR3[T1TIS]=01B, select comparator as input
2. CMP0MOD[1:0]=00B, select no built-in resistance 3-comparator mode
3. P1_AN[6:4]=111B, P2_AN[1]=1 configure the relevant ports to be analog mode
4. P1_PU[5:4]=00B, and the reset value is 00B. If there is no modification, this step can be ignored
5. Configure CMP0HYS[2:0], set the required hysteresis voltage, and the reset value is 000B
6. CMP0EN=1, enable comparator
7. The program configures CMP0SEL[1:0] based on requirement, selects one or more input channels, and refers to the description of CMP0SEL in Table 26-3
8. Comparator’s output is saved into bits CMP2OUT~CMP0OUT respectively,that is, CMP_SR[2:0]

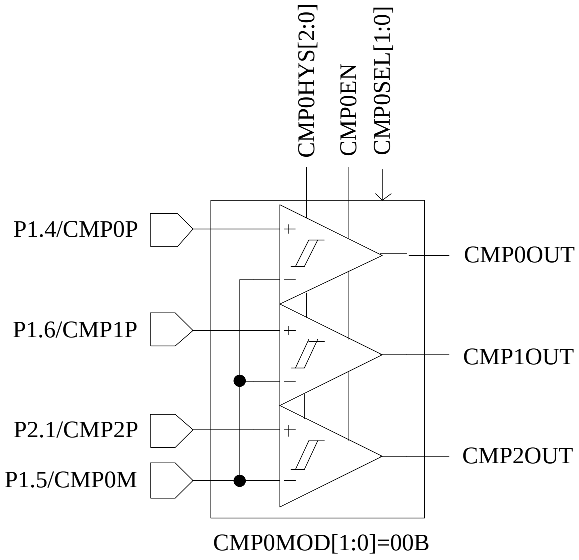


Figure 26-9 CMP0MOD[1:0]=00B,select no built-in resistance 3-comparator mode, which can be used for motor BEMF detection with off-chip virtual central point resistors

CMP0MOD[1:0]=01B, select the mode with built-in resistance 3 comparator, refer to Figure 26-10, which can be used for motor BEMF detection with built-in virtual central resistance. The negative input terminal is connected with the built-in resistance central point in the chip, and the positive input terminal is P1.4/CMP0P, P1.6/CMP1P, P2.1/CMP2P. The corresponding output is saved into CMP0OUT, CMP1OUT and CMP2OUT, respectively.

Built-in resistance 3-comparator mode register configuration:

1. TIM1_CR3[T1TIS]=01B, select comparator as input
2. CMP0MOD[1:0]=01B, select the built-in resistance 3 comparator mode
3. P1_AN[6]=1, P1_AN[4]=1, P2_AN[1]=1 configure the relevant ports as analog mode
4. P1_PU[4]=0, and the reset value is 0. If there is no modification, this step can be ignored
5. Configure CMP0HYS[2:0], set the required hysteresis voltage, and the reset value is 000B
6. CMP0EN=1, enable comparator
7. The program configures CMP0SEL[1:0] according to requirement, selects one or more input channels, and refers to the description of CMP0SEL in Table 26-3
8. Comparator’s output is saved to bits CMP2OUT~CMP0OUT respectively, that is, CMP_SR[2:0]

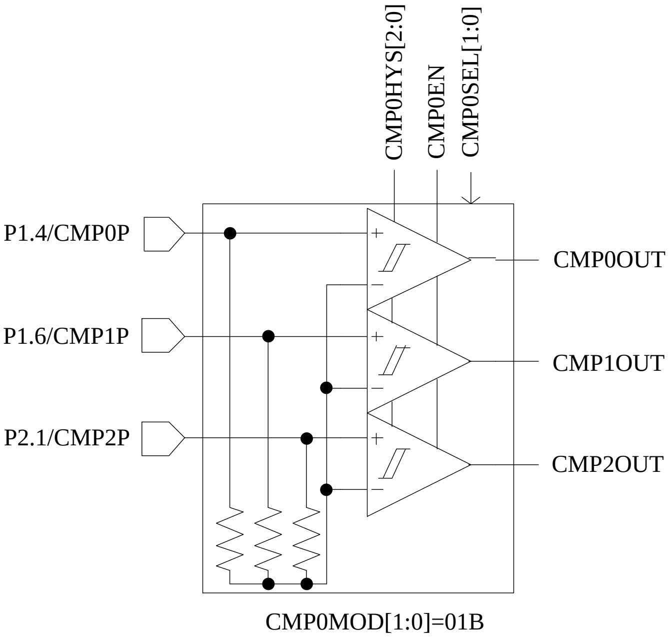


Figure 26-10 CMP0MOD[1:0]=01B, select the built-in resistance 3 comparator mode, which can be used for motor BEMF detection with built-in virtual central point resistors

CMP0MOD[1:0]=10B, select 3 differential comparators mode, refer to Figure 26-11, which can be used to detect the motor rotor position when differential HALL input. The negative input terminal is P1.5/CMP0M, P1.7/CMP1M, P2.1/CMP2M, and the corresponding positive input terminal is P1.4/CMP0P, P1.6/CMP1P, P2.1/CMP2P, and the corresponding output is sent to CMP0OUT, CMP1OUT and CMP2OUT, respectively.

Configuration of differential comparator mode register:

1. TIM1_CR3[T1TIS]=01B, select comparator as input
2. CMP0MOD[1:0]=10B, select 3 difference comparator mode
3. P1_AN[7:4]=1111B, P2_AN[2:1]=11B, configure the relevant ports as analog mode
4. P1_PU[4]=0, and the reset value is 0. If there is no modification, this step can be ignored
5. Configure CMP0HYS[2:0], set the required hysteresis voltage, and the reset value is 000B
6. CMP0EN=1, enablecomparator
7. The program configures CMP0SEL[1:0] as requirement, selects one or more input channels, and refers to the description of CMP0SEL in Table 26-3
8. Comparator output is sent to bits CMP2OUT~CMP0OUT respectively, that is, CMP_SR[2:0]

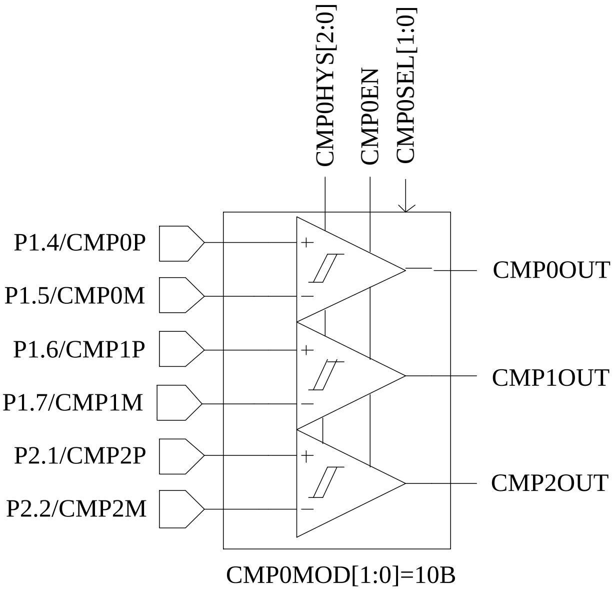


Figure 26-11 CMP0MOD[1:0]=10B, select 3 differential comparators mode,
which can be used to detect the rotor position of the motor when differential HALL input.

CMP0MOD[1:0]=11B, select the double-comparator mode, refer to Figure 26-12, the negative input end is connected with P1.5/CMP0M, the positive input end is P1.4/CMP0P, P1.3/CMP1PS, and the corresponding output is sent to CMP0OUT and CMP1OUT, respectively.

Dual-comparator mode register configuration:

1. TIM1_CR3[T1TIS]=01B, select comparator as input
2. CMP0MOD[1:0]=11B, select double comparators mode
3. P1_AN[5:3]=111B, P1_OE[3]=0B configure the relevant ports to be analog mode
4. The pull-up resistor of P1[5:3] can be selected to turn on by configuring P1_PU[5:3]=111B or turn off by configuring P1_PU[5:3]=000B, and the reset value is 0. If no modification is made, this step can be ignored. In double comparators mode, the pull-up resistors of P1[5:3] is only used for special occasions, and is normally turned off
5. Configure CMP0HYS[2:0], set the required hysteresis voltage, and the reset value is 000B
6. CMP0EN=1, enabling comparator
7. The program configures CMP0SEL[1:0] according to requirement, selects one or more input channels, and refers to the description of CMP0SEL in Table 26-3
8. Comparator's output is sent to bits CMP1OUT and CMP0OUT respectively, that is, CMP_SR[1:0]

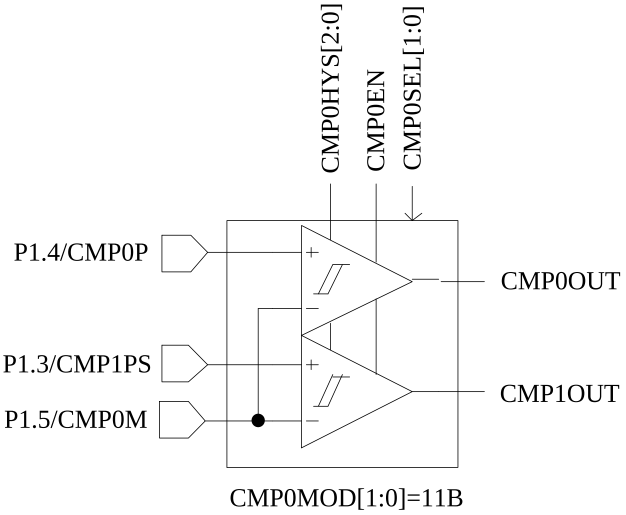


Figure 26-12 dual comparators mode

Hysteresis voltage of comparator 0 can be set up by the register, according to actual demand configure reasonable positive and negative hysteresis voltage. .

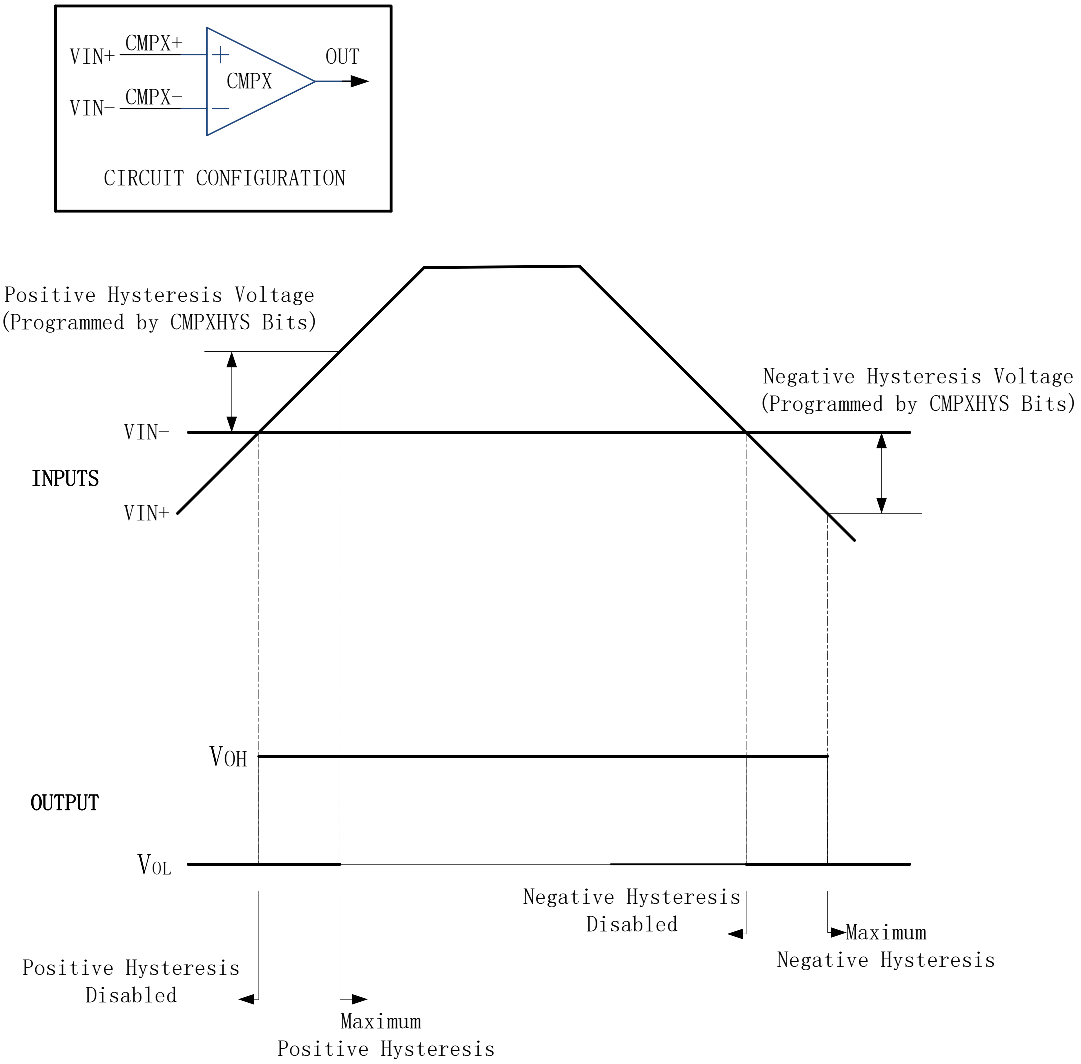


Figure 26-13 Diagram of hysteresis input and output of CMP0

26.1.4 Comparator sampling

Comparator sampling function is mainly used for BLDC drives and RSD function, is used to eliminate the interference from the switch. If you apply to the BLDC drives motors please refer to Sampling. If you apply to RSD please refer to RSD comparator sampling.

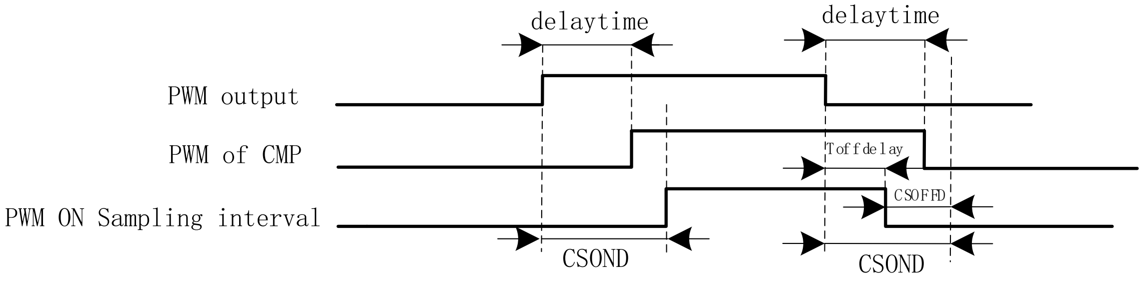


Figure 26-14 PWM ON sampling mode

The interference reflected in the output of PWM to the comparator is delayed relative to the jump edge of PWM, which is mainly affected by the following factors: the size of driving resistance, MOS switching speed, input delay and hysteresis setting of comparator. As shown in figure 27-14, delaytime is the delay from the output level of IC to the time where comparator detects level. When sample on the high level, the sampling interval should be surrounding by the high level read by the comparator in actual, first set the value of CSOND to exceed the oscillation of the MOS switch. At the same time, if the value of CSOFFD is not set, the end time of the sampling interval is delayed CSOND compared with the falling edge of PWM output wave. At this time, the actual sampling window has jumped out of the time corresponding to the high level (PWM of CMP) on the comparator, so setting the value of CSOFFD makes the actual sampling window close Toffdelay(Toffdelay=CSOND- CSOFFD) after the falling edge of PWM out wave.

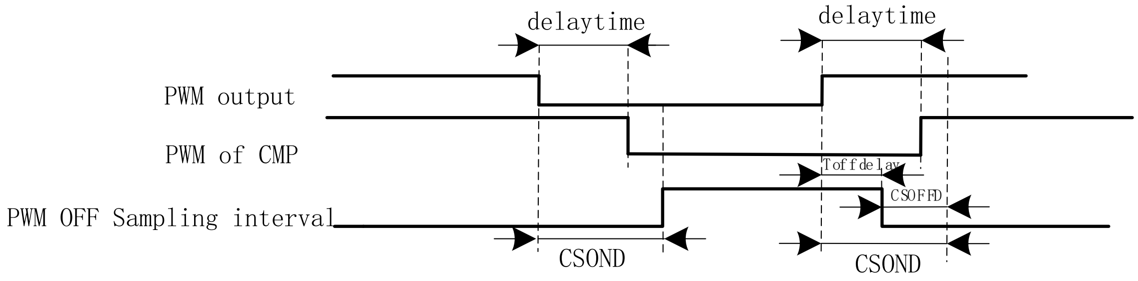


Figure 26-15 PWM OFF sampling mode

Similarly, when performing low level sampling, the sampling interval should be enveloped by the low level actually read on the comparator, and the delay time CSOND is first set to overcome the delay and the oscillation of the MOS switch. At the same time, if the value of CSOFFD is not set, the end time of the sampling interval is the delay CSOND after the rising edge of PWM output wave. At this time, the actual sampling window has jumped out of the time corresponding to the low level (PWM of CMP) on the comparator. Therefore, setting the value of CSOFFD makes the actual sampling window close Toffdelay (Toffdelay=CSOND- CSOFFD) after the rising edge of PWM out wave

Measure the delay of PWM output to the comparator: clear SAMSEL of CMP_CR3 register to forbid the comparator's sampling function, set the CMPSEL of CMP_CR3 register to output the comparator's comparison value, enable PWM output and comparator, manually rotate the motor to flip the comparator value, and measure the delay between PWM output and the comparator's output.

26.1.5 Comparator output

The output of the four comparators is connected to the multiplexer, and at the same time, the CMP_CR2 register’s CMPSEL selects one of the comparators to output the signal to a specific pin (P0.7),.

26.2 Comparator register

26.2.1 CMP_CR0 (0xD5)

Table 26-1 CMP_CR0 (0xD5)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IM		CMP2IM		CMP1IM		CMP0IM	
Type	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:6]	CMP3IM	Comparator CMP3 interrupt mode Refer to the CMP0IM description
[5:4]	CMP2IM	Comparator CMP2 interrupt mode Refer to the CMP0IM description
[3:2]	CMP1IM	Comparator CMP1 interrupt mode Refer to the CMP0IM description
[1:0]	CMP0IM	Comparator CMP0 interrupt mode 00: no interruption 01: interruption occurs at the rising edge 10: interruption occurs at the descending edge 11: both rising and descending edges are interrupted

26.2.2 CMP_CR1 (0xD6)

Table 26-2 CMP_CR1 (0xD6)

Bit	7	6	5	4	3	2	1	0
Name	HALLSEL	CMP3MOD		CMP3EN	CMP3HYS	CMP0HYS		
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7]	HALLSEL	HALL signal input selection 0: P0.2 / P3.7 / P3.6 1: P1.4 / P1.6 / P2.1
[6:5]	CMP3MOD	The positive input of comparator 3 is selected, refer to Figure 26-1, and the negative input is connected to P2.6 or DAC output. 00: single positive input mode, P2.7 is connected to the positive input terminal, refer to Figure 26-2 01: double comparators mode, P2.0 and P2.3 are connected to the positive input terminal, refer to Figure 26-3 1X: 3 comparators mode, P2.0, P2.3 and P2.7 are connected to the positive input terminal, refer to Figure 26-4
[4]	CMP3EN	Enable CMP3 0:Disable 1: Enable
[3]	CMP3HYS	Hysteresis configuration for comparator 3 0: Disable hysteresis 1: Enable hysteresis
[2:0]	CMP0HYS	CMP0 hysteresis voltage selection 000: no hysteresis 001: ±2.5mV 010: -5 mV 011: +5 mV 100: ±5 mV 101: -10 mV 110: +10 mV 111: ±10 mV

26.2.3 CMP_CR2 (0xDA)

Table 26-3 CMP_CR2 (0xDA)

Bit	7	6	5	4	3	2	1	0
Name	CMP4EN	CMP0MOD		CMP0SEL		RSV		CMP0EN
Type	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function

<table>
<tr><td>[7]</td><td>CMP4EN</td><td>Enable CMP4
0: Disable
1: Enable</td></tr>
<tr><td>[6:5]</td><td>CMP0MOD</td><td>Comparator CMP0/1/2 mode settings
00: no built-in resistor 3 comparators mode, refer to Figure 26-9
01: with built-in resistor 3 comparators mode, refer to Figure 26-10
10:3 differential comparators mode, refer to Figure 26-11
11: double comparator mode, CMP0, CMP1 works, CMP2 does not work, refer to Figure 26-12</td></tr>
<tr><td>[4:3]</td><td>CMP0SEL</td><td>Port combination selection of comparator 0, used in combination with CMP0MOD, the customer generally configures CMP0SEL=00.Under BLDC application, TIM1 will automatically control CMP0SEL. Please refer to Output enablement and polarity
<table>
<tr><th>CMP0MOD</th><th>CMP0SEL</th><th>Functional description</th></tr>
<tr><td rowspan="4">00</td><td>00</td><td>The comparator cmp0/1/2 works at the same time, and the input ends of the three comparators are all connected with CMP0M. The hardware automatically compares the positive input end CMP0P, CMP1P and CMP2P with the common negative input end CMP0M, and the output results are sent to CMP0OUT, CMP1OUT and CMP2OUT, respectively</td></tr>
<tr><td>01</td><td>Comparator CMP0 works, CMP1/2 is idle, positive end is connected to CMP0P, negative end is connected to CMP0M, and output is connected to CMP0OUT</td></tr>
<tr><td>10</td><td>Comparator CMP1 works, CMP0/2 is idle, positive end is connected to CMP1P, negative end is connected to CMP0M, and output is connected to CMP1OUT</td></tr>
<tr><td>11</td><td>Comparator CMP2 works, CMP0/1 is idle, positive end is connected to CMP2P, negative end is connected to CMP0M, and output is connected to CMP2OUT</td></tr>
<tr><td>01</td><td>00</td><td>The comparator CMP0/1/2 works at the same time. The input terminals of the three comparators are connected to the center point of built-in BEMF resistance. The hardware automatically compares the positive input CMP0P, CMP1P and CMP2P with the common negative input CMP0M respectively, and the output results are sent to CMP0OUT, CMP1OUT and CMP2OUT,</td></tr>
</table>
</td></tr>
</table>

					respectively
				01	Comparator 0 selects the port combination corresponding to CMP0, the positive end is connected to CMP0P, the negative end is connected to the center point of built-in BEMF resistance, and the output is connected to CMP0OUT
				10	Comparator 0 selects the port combination corresponding to CMP1, the positive end is connected to CMP1P, the negative end is connected to the center point of built-in BEMF resistance, and the output is connected to CMP1OUT
				11	Comparator 0 selects the port combination corresponding to CMP2, the positive end is connected to CMP2P, the negative end is connected to the center point of built-in BEMF resistance, and the output is connected to CMP2OUT
			10	00	The comparator CMP0/1/2 works at the same time. The positive input end of the three comparators is connected to CMP0P, CMP1P and CMP2P respectively, and the corresponding negative input end is connected to CMP0M, CMP1M and CMP2M respectively. The output results are sent to CMP0OUT, CMP1OUT and CMP2OUT respectively
				01	Comparator 0 selects the port combination corresponding to CMP0, the positive input end is connected to CMP0P, the negative input end is connected to CMP0M, and the output end is connected to CMP0OUT
				10	Comparator 0 selects the port combination corresponding to CMP1, the positive input end is connected to CMP1P, the negative input end is connected to CMP1M, and the output end is connected to CMP1OUT
				11	Comparator 0 selects the port combination corresponding to CMP2, the positive input end is connected to CMP2P, the negative input end is connected to CMP2M, and the output end is connected to CMP2OUT
			11	00	The comparator CMP0/1 works at the same

				time. The positive input end is connected to CMP0P and CMP1PS, and the negative input end is fixed to CMP0M. The output results are sent to CMP0OUT and CMP1OUT respectively
			01	Comparator 0 selects the port combination corresponding to CMP0, that is, the positive input end is connected to CMP0P, the negative input end is connected to CMP0M, and the output end is connected to CMP0OUT
			10	Comparator 0 selects the port combination corresponding to CMP1, that is, the positive input end is connected to CMP1PS, the negative input end is connected to CMP0M, and the output end is connected to CMP1OUT
			11	Reserved
[2:1]	RSV	Reserved		
[0]	CMP0EN	CMP0 Enable 0: Disable 1: Enable		

26.2.4 CMP_CR3 (0xDC)

Table 26-4 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	Function
[7]	CMPDTEN	Enable comparator dead zone sampling 0: Disable 1: Enable
[6:5]	DBGSEL	DEBUG signal selection Select one debug signal output to port P01 00: debug signal is not enabled 01: square wave freewheeling mask end and zero crossing signal. Refer to Debug of Freewheeling Mask and Commutation 10: ADC trigger signal 11: comparator sampling interval, refer to Comparator debugging
[4:3]	SAMSEL	Enable comparator CMP0,CMP1,CMP2 and ADC in PWM on/off sampling function, refer to Comparator debugging 00: sampling on on and off, no delay sampling on

		01: only off sampling, the delay of sampling according to CMP_SAMR 10: only on sampling, the delay of sampling according to CMP_SAMR 11: both on and off sampling, the delay of sampling according to CMP_SAMR
[2:0]	CMPSEL	Comparator output selection Select one comparator signal output to port, refer to Comparator debugging 000: no output 001: CMP0 010: CMP1 011: CMP2 100: CMP3 101: CMP4 110: Reserved 111: ADC calculation results compare bits

26.2.5 CMP_SAMR (0x40AD)

Table 26-5 CMP_SAMR (0x40AD)

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

Bit	Name	Function
[7:4]	CSOND	Comparator CMP0,CMP1 and CMP2 delay of the sampling start When PWM off->on or PWM on->off, MOS tube on and off will interfere with the comparator, and CSOND is set to delay sampling start of comparators CMP0, CMP1 and CMP2 to avoid interference. Calculating CSOND need to take into account the delay generated by the drive circuit. Suppose the MCU clock is 24MHz(41.67ns) Delay time = CSOND x 41.67 x 8ns Note: CSOND must >= CSOFFD Please refer to Sampling for application of BLDC driver and refer to RSD comparator sampling for application of RSD.
[3:0]	CSOFFD	Comparator CMP0,CMP1,CMP2 close sampling time When PWM off->on or PWM on->off, MOS tube on and off will interfere with the comparator. PWM reflects the interference of comparator, and CSOFFD is set to shield the interference stage of comparator. Suppose the MCU clock is 24MHz(41.67ns) Close sampling time = CSOFFD x 41.67 x 8ns Note: CSOND must >= CSOFFD Please refer to Sampling of BLDC driver and refer to RSD comparator sampling for application of RSD.

26.2.6 CMP_SR (0xD7)

Table 26-6 CMP_SR (0xD7)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IF	CMP2IF	CMP1IF	CMP0IF	CMP3OUT	CMP2OUT	CMP1OUT	CMP0OUT
Type	R/W0	R/W0	R/W0	R/W0	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7]	CMP3IF	Compare interrupt flag of CMP3 The CMP3 interrupt event is generated and the bit is set by hardware. It is cleared by software. 0: no event occured. 1: interrupt event occured.
[6]	CMP2IF	Compare interrupt flags of CMP2 The CMP2 interrupt event is generated and the bit is set by hardware. It is cleared by software.TIM1_CR3[T1TIS]=01b must be configured, otherwise it is the interrupt flag of HALL. 0: no event occured 1: interrupt event occured
[5]	CMP1IF	Compare interrupt flags of CMP1 The CMP1 interrupt event is generated and the bit is set by hardware. It is cleared by software.TIM1_CR3[T1TIS]=01b must be configured, otherwise it is the interrupt flag of HALL. 0: no event occured 1: interrupt event occured
[4]	CMP0IF	Compare interrupt flag of CMP0 The CMP0 interrupt event is generated and the bit is set by hardware. It is cleared by software.TIM1_CR3[T1TIS]=01b must be set, otherwise it is the interrupt flag of HALL. 0: no event occured 1: interrupt event occured
[3]	CMP3OUT	Compare the results of CMP3 0: the current comparison result of CMP3 is 0 1: the current comparison result of CMP3 is 1
[2]	CMP2OUT	Compare the results of CMP2 comparisons TIM1_CR3[T1TIS]=01b must be set; otherwise, it is the current level of HALL. 0: the current comparison result of CMP2 is 0 1: the current comparison result of CMP2 is 1
[1]	CMP1OUT	Compare the results of CMP1 TIM1_CR3[T1TIS]=01b must be set; otherwise, it is the current level of HALL. 0: the current comparison result of CMP1 is 0

		1: the current comparison result of CMP1 is 1
[0]	CMP0OUT	Compare the results of CMP0 TIM1_CR3[T1TIS]=01b must be set; otherwise, it is the current level of HALL. 0: the current comparison result of CMP0 is 0 1: the current comparison result of CMP0 is 1

26.2.7 EVT_FILT (0xD9)

Table 26-7 EVT_FILT (0xD9)

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

Bit	Name	Function
[7]	TSDEN	The Temperature sensor detects enable. 0: Disable 1: Enable.
[6:5]	TSDADJ	Temperature sensor detect adjustment. 00:105℃ 01:120℃ 10:135℃ 11:150 ℃
[4:3]	MOEMD	MOE signal hardware zeroing and enabling The occurrence of bus current overshoot event will make the MOE hardware reset and enable 00: MOE does not reset automatically 01: MOE is cleared automatically 10: MOE reset automatically, and the overflow event on DRV counter automatically enables MOE (mainly used for square wave) 11: MOE reset automatically, and automatically enables MOE (mainly used for square wave) on the overflow and underflow events of DRV counter and every 5us.
[2]	EFSRC	Bus current protection event filter module input source 0: comparator CMP3, the protection interrupt is CMP3 interrupt 1: external interrupt INT0(P0.0), the protection interrupt is external interrupt INT0
[1:0]	EFDIV	Bus current protection event filtering width: 00: no filtering 01:4 system clock cycles 10:8 system clock cycles 11:16 system clock cycles

27 Power supply module

27.1 LDO

27.1.1 LDO module operation instructions

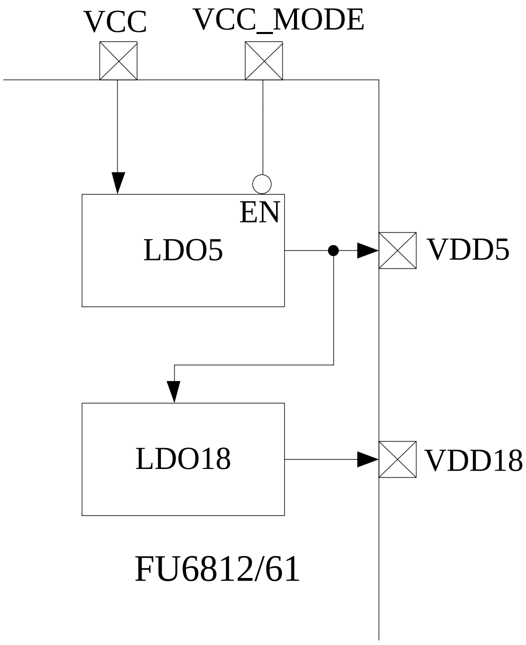


Figure 27-1 Function block diagram of power module

The port corresponding to the LDO module is shown in Figure 27-1. The function of LDO is to reduce the input power supply to 5V VDD5 and 1.8V VDD18, respectively, to supply power to the internal analog and digital modules of the chip. VDD5 can be selected as internal LDO5 generation or external supply, which is determined by VCC_MODE.

FU6812:

Single power supply high voltage mode (VCC_MODE=0). VCC= 5~24V. Refer to Figure 27-2.

Dual power mode (VCC_MODE=1), VCC≥VDD5. VCC=5~36V, VDD5=5V. Refer to Figure 27-3.

Single power supply low-voltage mode (VCC_MODE=1). VCC=VDD5=3~5.5V. Refer to Figure 27-4.

FU6861Q:

Mode 1: VCC_MODE=0, VCC= 5~24V, VDRV=7~18V

Mode 2: VCC_MODE=1, VCC=VDD5=3~5.5V, VDRV=7~18V

FU6861N:

Mode 1: VCC=5~24V, VDRV=7~18V

Note: VCC_MODE=1, the Voltage is VDD5

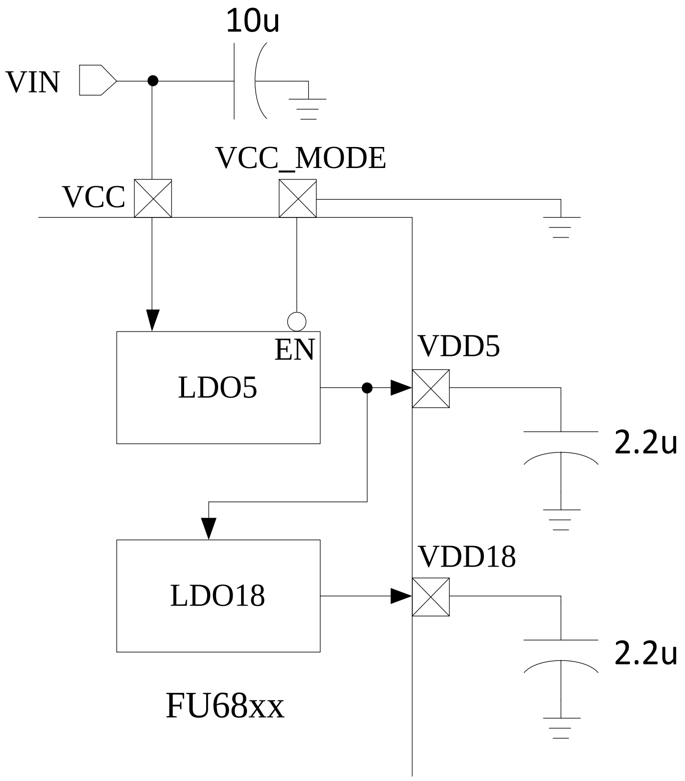


Figure 27-2 Single-supply high-voltage mode power connection

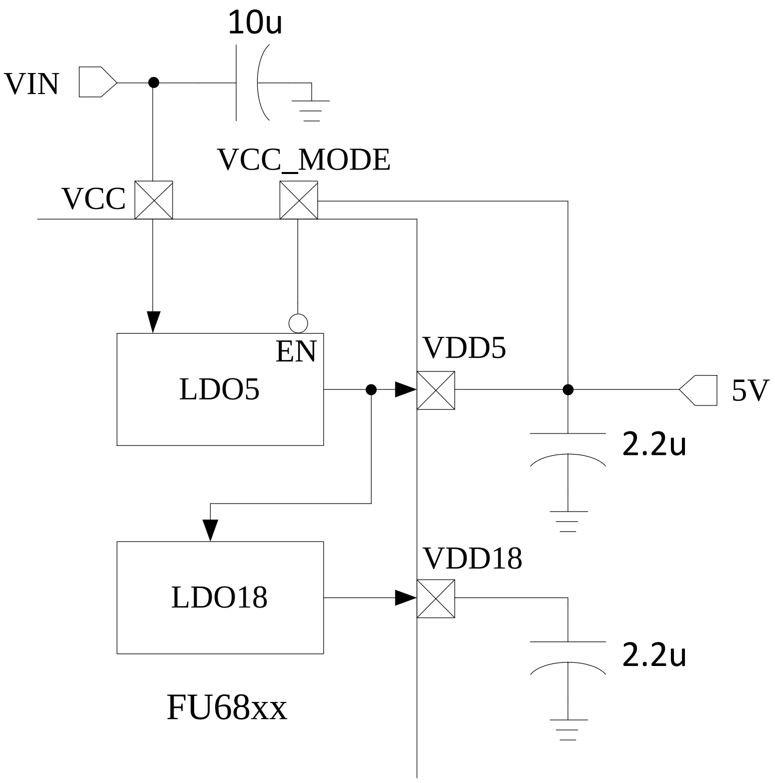


Figure 27-3 Dual power mode power connection

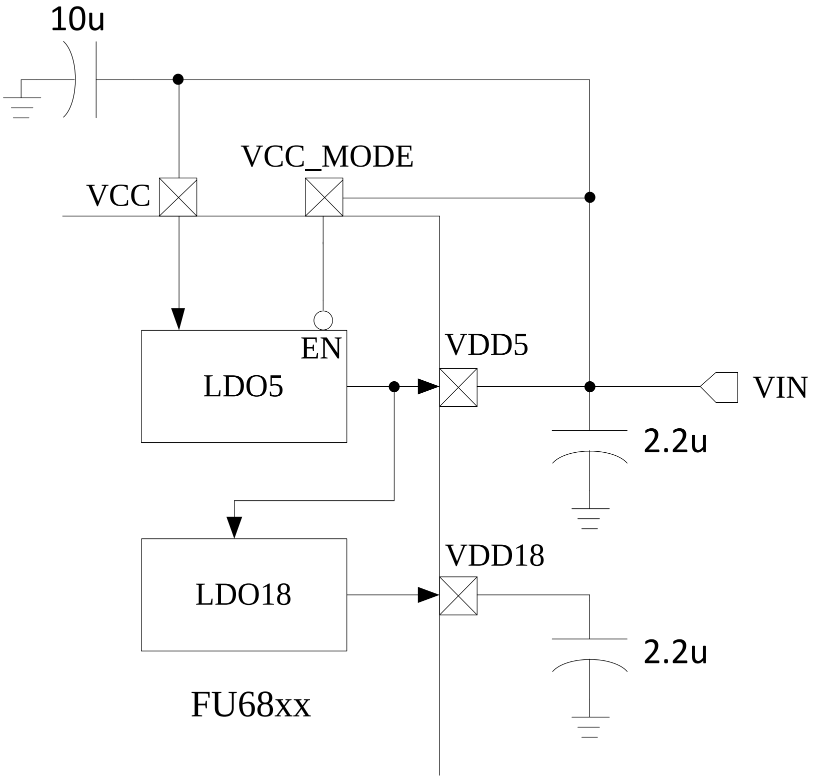


Figure 27-4 Single-supply low-voltage mode power connection

27.2 Low pressure test

27.2.1 Operation instructions of low-voltage detection module

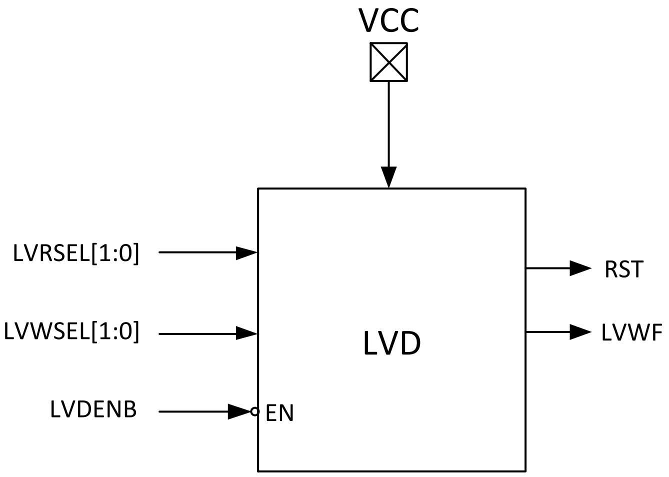


Figure 27-5 Low voltage detection module

To enable the low-voltage detection module, the registers need to be configured as follows: LVDENB=0.

27.2.2 CCFG2:RST_MOD (0x401D)

Table 27-1 Low voltage detection module related register 1

Bit	7	6	5	4	3	2	1	0
Name	LVRSEL		WDTBTEN	WDTRSTEN	RSV		LVWSEL	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function

[7:6]	LVRSEL	Low voltage reset voltage selector. The low voltage reset detects the voltage value of VDD5.
		00: the corresponding VDD5 reset voltage is 2.8V;
		01: the corresponding VDD5 reset voltage is 3.0V;
		10: the corresponding VDD5 reset voltage is 3.5V;
		11: the corresponding VDD5 reset voltage is 3.8V.

[5:2]		Please refer to table 33-2.

[1:0]	LVWSEL	Low voltage warning voltage selector. The low voltage early warning detects the VCC voltage value.
		00: the corresponding VCC warning voltage is 7V;
		01: the corresponding VCC warning voltage is 8V;
		10: the corresponding VCC warning voltage is 9V;
		11: the corresponding VCC warning voltage is 10V.

27.2.3 CCFG1:CK_RST_CFG (0x401E)

Table 27-2 Low-voltage detection module related registers 2

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

Bit	Name	Function
[7]	LVWENB	Low voltage early warning enabled 0:Enable 1:Disable
[6]	LVWIE	VCC low voltage alarm interrupts enabling, and LVWSEL sets the VCC early warning voltage threshold. 0: shut down 1: enable. If low voltage alarm is needed to enable interrupt, it must also enable low voltage reset detection circuit (i.e. LVDENB=0).
[5:0]		Please refer to Table 32-1

27.2.4 LVSR (0xDB)

Table 27-3 LVSR (0xDB)

Bit	7	6	5	4	3	2	1	0
Name	RSV		EXT0CFG			TSDF	LVWF	LVWIF
Type	R	R	R/W	R/W	R/W	R	R	R/W

Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:6]	RSV	Reserved
[5:3]	EXT0CFG	The P0 interface configuration of Interrupt 0 000: configure P0.0 as the external interrupt 0 interface 001: configure P0.1 as the external interrupt 0 interface 010: configure P0.2 as the external interrupt 0 interface 011: configure P0.3 as the external interrupt 0 interface 100: configure P0.4 as the external interrupt 0 interface 101: configure P0.5 as the external interrupt 0 interface 110: configure P0.6 as the external interrupt 0 interface 111: configure the comparator CMP4 output as an external interrupt 0 interface
[2]	TSDF	The over temperature state bit 0: the current temperature does not exceed the set temperature. 1: the current temperature exceeds the set temperature. This flag bit is often used in conjunction with the temperature protection interrupt flag bit (TSDIF, namely TCON[5]), which reflects the dynamic overtemperature state.
[1]	LVWF	VCC low voltage waring flag The low voltage mark reflects whether the voltage state is low or not 0: no alarm at present 1: current low voltage detection alarm
[0]	LVWIF	VCC low voltage interrupt flag This bit reflects whether a low-voltage event has ever occurred. When the low-power detection interrupt is enabled and the low-power detection interrupt is generated, this bit is set and program enters the interrupt at the same time.The bit is set by hardware, and cleared by software. If Low voltage detection interrupt isn’t enabled, this bit will not be set by hardware. 0: no event occurred 1: low voltage detection alarm occurred

28 FLASH

28.1 Main Feature

- A total of 16 KB Flash ROM; .
- Each sector is 128 bytes, a total of 128 sectors;
- The first 127 sectors support sector self-erase/write operation, support online programming and application programming;
- Flash supports sector erase (except the 128th sector);

28.2 FLA_CR: Programmable control register

Table 28-1 FLA_CR (0x85)

Bit	7	6	5	4	3	2	1	0
Name	RSV			FLAERR	FLAACT	RSV	FLAERS	FLAWEN
Type	R	R	R	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:5]	RSV	Reserved
[4]	FLAERR	Programmable error flag, readable 0: When FLASH autowrites, programming or erasing is successful 1: When FLASH autowrites, programming or erasing is failed
[3]	FLAACT	FLASH erase/write operation trigger bits Write 0 is invalid Write 1 to start Flash operations such as programming and erasing
[2]	RSV	Reserved
[1]	FLAERS	Sector erase enable 0: Disable 1: Enable Note: Only when FLAWEN is set to 1, FLAERS makes effects
[0]	FLAWEN	Programming enable 0: Disable 1: Enable Note: Only when FLAWEN is set to 1, FLAERS makes effects

28.3 FLA_KEY: FLASH program unlock register

Table 28-2 FLA_KEY (0x84)

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

Bit	Name	Function
[7:0]	FLA_KEY	FLASH erase/program unlock registers Writing 0x5A and 0x1F sequentially to FLA_KEY turns on the "software programming FLASH" function.If the order is not correct or some other value is written, this function is frozen until the next system reset. Once the lock is unlocked, any action to write FLA_CR will cause FLA_KEY to be locked again.

		Read: the lowest 2 bits reflect the internal state, and the high 6 bits return 0x00: 00: locked 01:0x5A has been written and is waiting for 0x1F to write 11: unlocked 10: freezed

28.4 FLASH self-writing operation

1. Note:

In order to ensure the safety of the FLASH operation, it is strongly recommended that before self-burn ban all interrupt event to avoid wrong FLASH operation caused by the MOVX instruction in interrupt service routines.

2. The steps of software self-erase FLASH sector:

Step0: clear the bit used for enabling all interrupts, EA.

Step1: write 0x03 to register FLA_CR.

Step2: for unlocking FLASH, write 0x5A, 0x1F into the register FLA_KEY in order

Step3: using MOVX instructions, write any values to the FLASH sector required erasure.

Step4: write 1 to FLA_CR.FLAACT then enter into the process of software erasure. After instruction execution erasure is completed automatically, and automatically locked again.

3. Self-write data to FLASH by software

Step0: clear the bit used for enabling all interrupts, EA.

Step1: write 0x01 to register FLA_CR.

Step2: for unlocking FLASH, write 0x5A, 0x1F into the register FLA_KEY in order

Step3: using MOVX instructions, write datas to the given FLASH address

Step4: write 1 to FLA_CR.FLAACT then enter into the process of one byte self-write. After instruction execution data is writen, and automatically locked again.

Note:

1) In order to ensure the safety of user’s program, it’s necessary for disabling all interrupts. It can avoid writing to rom_code caused by the MOVX instruction.

2) All of the above three kinds of operation of flash, internal circuit need a long time to complete, one erase sector needs around 120 ~150 ms.

3) Each sector size of 128 bytes, the final sector (address range: 0x3F80 ~ Ox3FFF) at any time will not be erased. Any instructions in the noprotection area to access (including reading, writing and erasing operation) protected areas, will reset the MCU.

29 CRC (Cyclic Redundancy Check Calculation Unit)

29.1 CRC16 functional block diagram

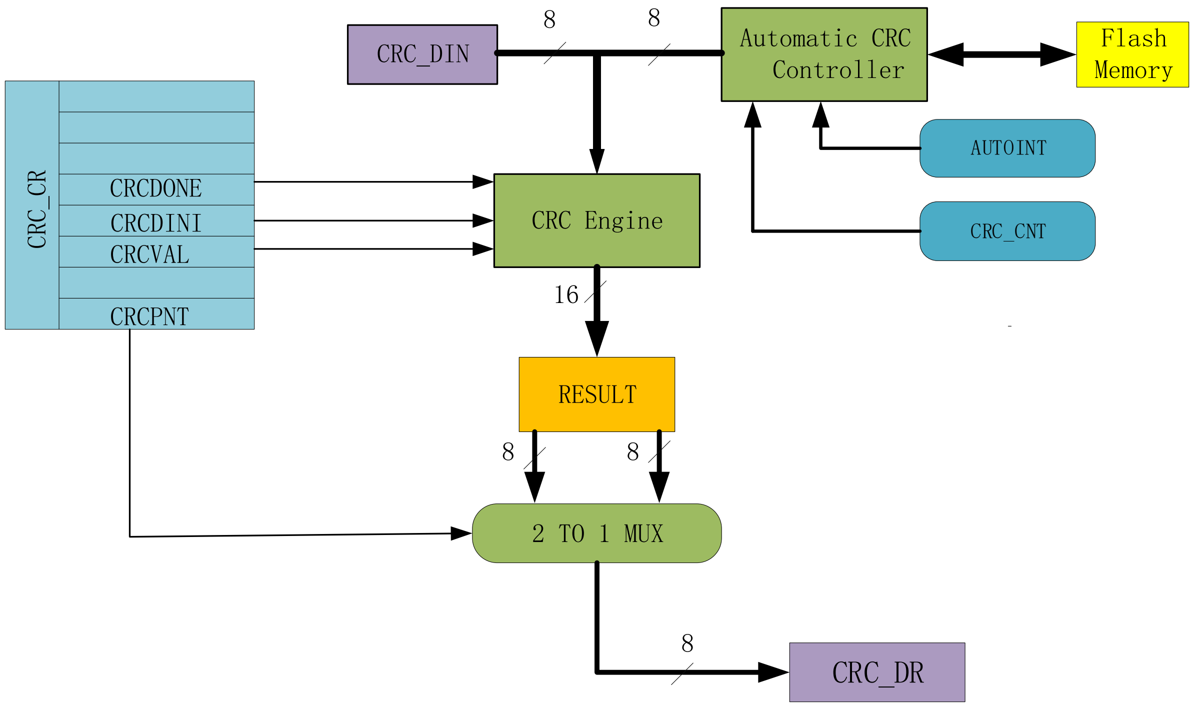


Figure 29-1 CRC functional block diagram

CRC (cyclic redundancy check unit) is according to the generation of fixed polynomial to get any CRC calculation results of 8 bits of data., as shown in Figure 29-1, CRC receives the 8 bits of data of the CRC_DIN register, 16-bit calculation result will be sent to the internal registers after the completion, through CRCPNT and CRC_DR indirectly access to the internal result egisters.

Table 29-1 CRC standard and generator polynomial

Number	CRC Criterion	Generator Polynomial	Hexadecimal
1	CRC12	x^12+x^11+x^3+x^2+x+1	80F
2	CRC16	x^16+x^15+x^2+1	8005
3	CRC16-CCITT	x^16+x^12+x^5+1	1021
4	CRC32	x^32+x^26+x^23+x^22+x^16+x^12 +x^11+x^10+x^8+x^9+x^5+x^4+x+1	04C11DB7

29.2 CRC16 generator polynomial

FU6812/61 selects the generation polynomial based on CRC16-CCITT standard:

$$X^{16}+X^{12}+X^{5}+1$$

29.3 CRC16 basic logic diagram

Figure 29-2 shows the schematic diagram of serial CRC16 circuit. FU6812/61 is implemented by parallel algorithm, and the result can be calculated with a system clock for each input byte.

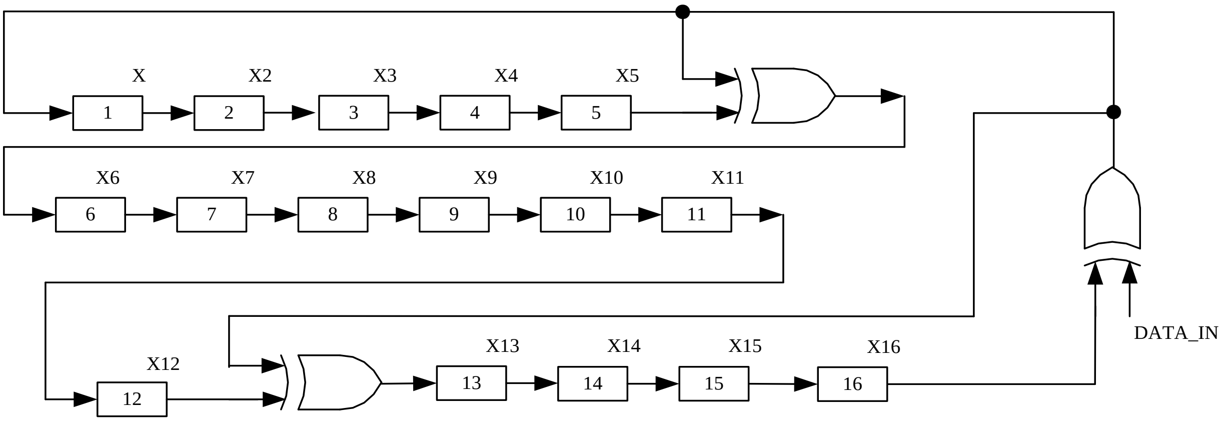


Figure 29-2 Schematic diagram of CRC16 circuit

29.4 Instructions

29.4.1 Calculates the CRC for a single byte

To calculate the CRC value of a single byte, follow these steps:

1、 There are two ways to initial CRC_DR according requirement. If the initial value is 0x0000 or 0xffff, CRC_CR[CRCVAL] can be configured and CRC_CR[CRCDINI] can be set to 1.If you want the initial value to be any value, CRC_CR[CRCPNT] and CRC_DR can be used to initiate CRC.

2、 Writing a data to the input data register CRC_DIN, such as 0x63, the CRC result is computed for the next clock cycle

3、 Read CRC results: write CRC_CR[CRCPNT] bit as 1, the software reads the result output register CRC_DR, get high byte data;write 0 to CRC_CR[CRCPNT], read CRC_DR again, get the low byte number;The combined results are the correct CRC results.

29.4.2 Batch computing ROM data CRC

To calculate the CRC value of a continuous region of ROM data, please follow the following steps:

1、 Initialize CRC_DR, method with single-byte CRC initialization;

2、 Write the appropriate value to CRC_BEG and set the start sector of the ROM to calculate.

3、 Write the appropriate value to CRC_CNT to set the sector offset from the start sector to the end sector

4、 Write 1 to CRC_CR[AUTOINT] and leave the other bits unchanged, then auto computing will start..

5、 The method for reading CRC results is the same as the method for reading single-byte CRC

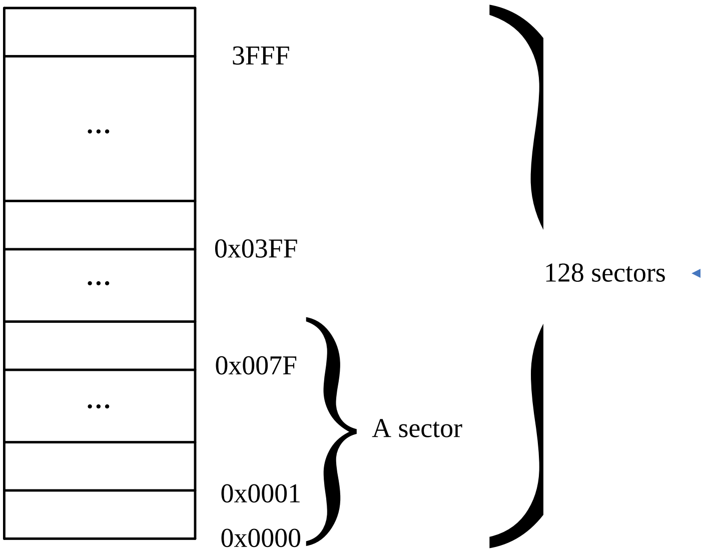


Figure 29-3 ROM access partition diagram

See Figure 29-3. The entire ROM consists of 16K bytes divided into 128 sectors numbered sector0 to sector127. Each sector contains 128 bytes per sector.When performing CRC batch calculation, the initial sector value CRC0BEG can be any value between 0x00~0x7F, including 0x00 and 0x7F. The total number CRC0CNT of sectors to be calculated is 0x00~0x7F, including 0x00 and 0x7F.

It should be noted that the value of CRC_BEG should decrease as the value of CRC_BEG increases. For example, if the value of CRC_BEG is 0x7F, the value of CRC_CNT can only be 0x00, that is, the CRC value in the last sector can only be calculated. At this point, if the value of CRC_CNT is set to 0x01 or higher by accident, the CRC controller hardware will automatically limit the number of bytes computed so that the CRC engine only calculates the CRC value of the data in the last sector.

29.5 CRC register

29.5.1 Control register: CRC_CR

Table 29-2 CRC_CR (0x4022)

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

Bit	Name	Function

[7:5]	RSV	Reserved
[4]	CRCDONE	Automatic CRC calculation completion flag. In the process of automatic CRC calculation mode, the hardware automatically writes 0 to this bit, and the software code will stop executing. In other cases, the hardware automatically sets this position to 1, so when the software reads this bit it always returns 1.
[3]	CRCDINI	CRC results initialization enablement 0: invalid initialization 1: valid initialization When the software writes 1 to this bit, the hardware does not actually write 1 to this bit, but generates synchronously a high level pulse of a clock cycle to the CRC engine as a condition for CRC result initialization.So whatever value the software writes to this bit, it always returns 0 when it reads it.
[2]	CRCVAL	CRC results initializing selection bit. 0: initialize the CRC result to 0x0000 1: initialize the CRC result to 0xFFFF
[1]	AUTOINT	CRC automatically calculates enablement. When you write 1 to this bit, the data in a continuous block of Flash is performed automatically CRC calculations. The starting block of calculation is CRC_BEG. A total of CRC_CNT blocks are calculated. Note: before enabling automatic CRC computing, the other bits should be configured before writing 1 to this bit. In other words, this bit cannot be configured with the other bits at the same time.
[0]	CRCPNT	The pointer of CRC result. 0: when reading CRC_DR register, the low byte (7-0 bit) of 16-bit CRC result is accessed. 1: when reading CRC_DR register, the high byte (15-8 bits) of 16-bit CRC result is accessed.

Note:

As CRC calculation process is divided into two categories, one is a single byte CRC calculation, one is ROM data batch CRC automatic calculation.Writing 1 to the bit[1] of the control register CRC_CR will immediately start the automatic calculation process of CRC. If you want to calculate the CRC value of a single byte written to the CRC_DIN register by software, the bit[1] of the CRC0STA register can only be 0.

29.5.2 Input data register: CRC_DIN

Table 29-3 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	Function

[7:0]	CRC_DIN	CRC module input data. Each time a data is written to this register, the CRC module automatically calculates the new CRC result and overrides the original CRC result according to the input data and the existing CRC result. Note: this register is a virtual register, written data is not saved. 0x00 is returned when this address is read.

29.5.3 Result output register: CRC_DR

Table 29-4 CRC0DAT (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	Function
[7:0]	CRC_DR	CRC result output. Each time the register is read or written, the high or low bytes of the CRC result are accessed according to the result pointer CRC0PNT in the control register CRC_CR.

Note:

Since the value of this register is directly determined by software and can be changed by other signals, it is directly placed in the CRC module rather than in the dedicated register.

29.5.4 Automatic calculation of the starting point register: CRC_BEG

Table 29-5 CRC0BEG (0x4024)

Bit	7	6	5	4	3	2	1	0
Name	RSV	CRC_BEG						
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	Function
[7]	RSV	Reserved
[6:0]	CRC_BEG	Automatically computes the ROM starting sector of CRC. For example: if the value of CRC_BEG[7:0] is 1 and 128 bytes per sector size, the automatic CRC calculation begins with the address 1 x 128=128. It actually begins with the first byte of the second Sector.

29.5.5 Automatic block count register: CRC_CNT

Table 29-6 CRC_CNT (0x4025)

Bit	7	6	5	4	3	2	1	0
Name	RSV	CRC_CNT						
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	Function
7	RSV	Reserved
[6:0]	CRC_CNT	Automatic CRC calculation of sector offset. This value defines the offset of the ROM sector from which the CRC value needs to be computed. Through it, you can determine which the end sector of the automatic CRC calculation.

30 Sleep mode

30.1 PCON register

Table 30-1 PCON (0x87)

Bit	7	6	5	4	3	2	1	0
Name	RSV		GF3	GF2	GF1	RSV	STOP	IDLE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
[7:6]	RSV	Reserved
[5]	GF3	Generic flag bit 3
[4]	GF2	Generic flag bit 2
[3]	GF1	Generic flag bit 1
[2]	RSV	Reserved
[1]	STOP	Write 1 to make the chip into sleep mode, it’s cleared by hardware after waking up
[0]	IDLE	Write 1 to make the chip into standby mode, it’s cleared by hardware after waking up Power consumption modes: {STOP, IDLE} =1x, the system sleeps {STOP, IDLE} =01, the system in the standby state {STOP, IDLE} =00, the system works normally

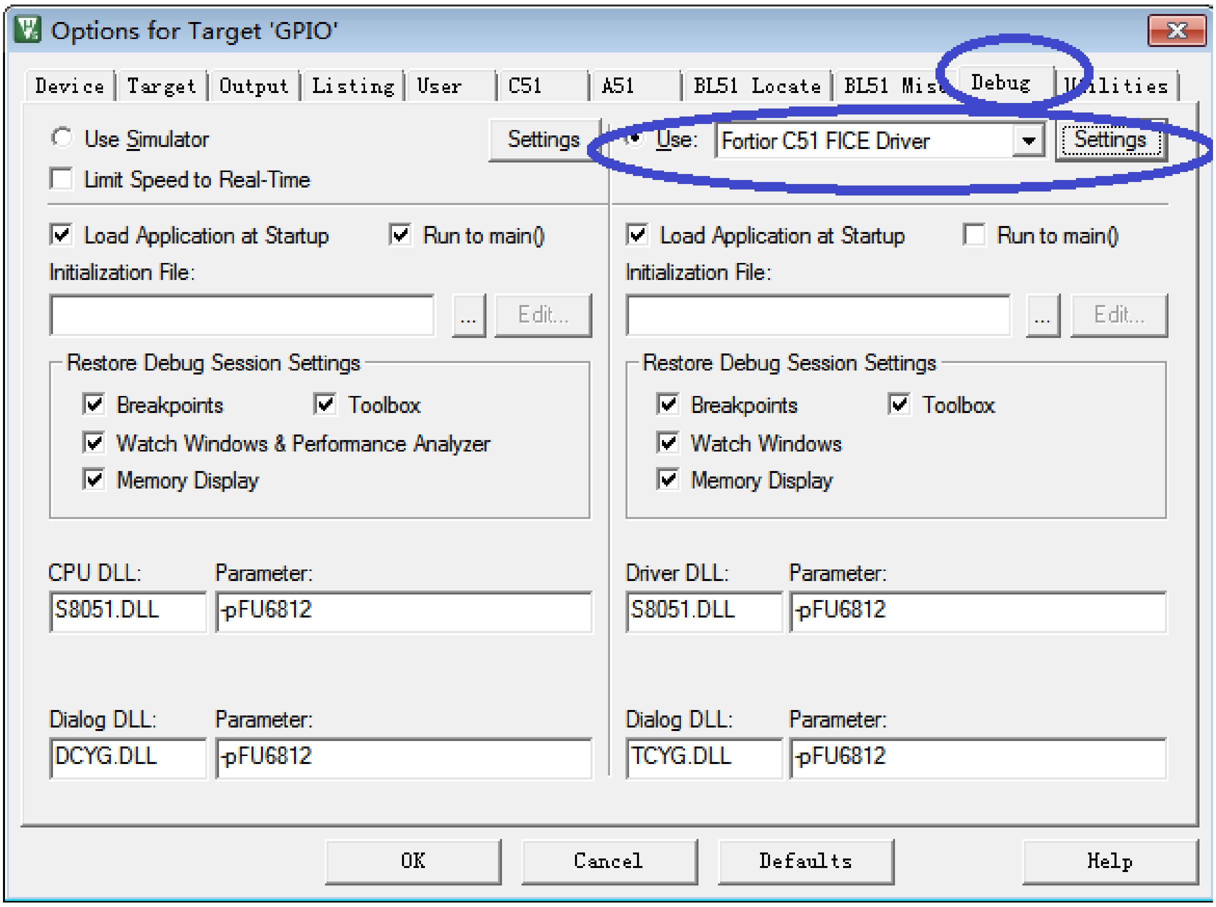
30.2 Power consumption modes

There are three power consumption modes: normal, standby, and sleep. The working conditions of modules under various power consumption modes are summarized as follows:

Table 30-2 power consumption mode

model	describe	Wake up the source	Power consumption performance
Normal	Except for the closed peripherals, other modules work at full speed	NA	High power consumption, Best performance.
IDLE	The CPU clock is gated, and whether other functional modules turn off or work depending on their control bits. The watchdog clock is door-controlled.	Any interrupt External /Debug reset	Low power consumption Flexible performance
Sleep	FLASH Deep Sleep. Analog fast clock circuit shut down, software should pay attention to ensure that ADC, FOC, motor control/drive circuit is idle before entering sleep. The watchdog clock is turned off.	External interrupt, External/Debug reset	Low power consumption Flexible performance

31 Code Protection



Options for Target 'GPIO'

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

○ Use Simulator Settings
□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()
Initialization File:
... Edit...

Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows & Performance Analyzer
☑ Memory Display

CPU DLL: S8051.DLL Parameter: -pFU6812

Dialog DLL: DCYG.DLL Parameter: -pFU6812

● Use: Fortior C51 FICE Driver Settings

☑ Load Application at Startup □ Run to main()
Initialization File:
... Edit...

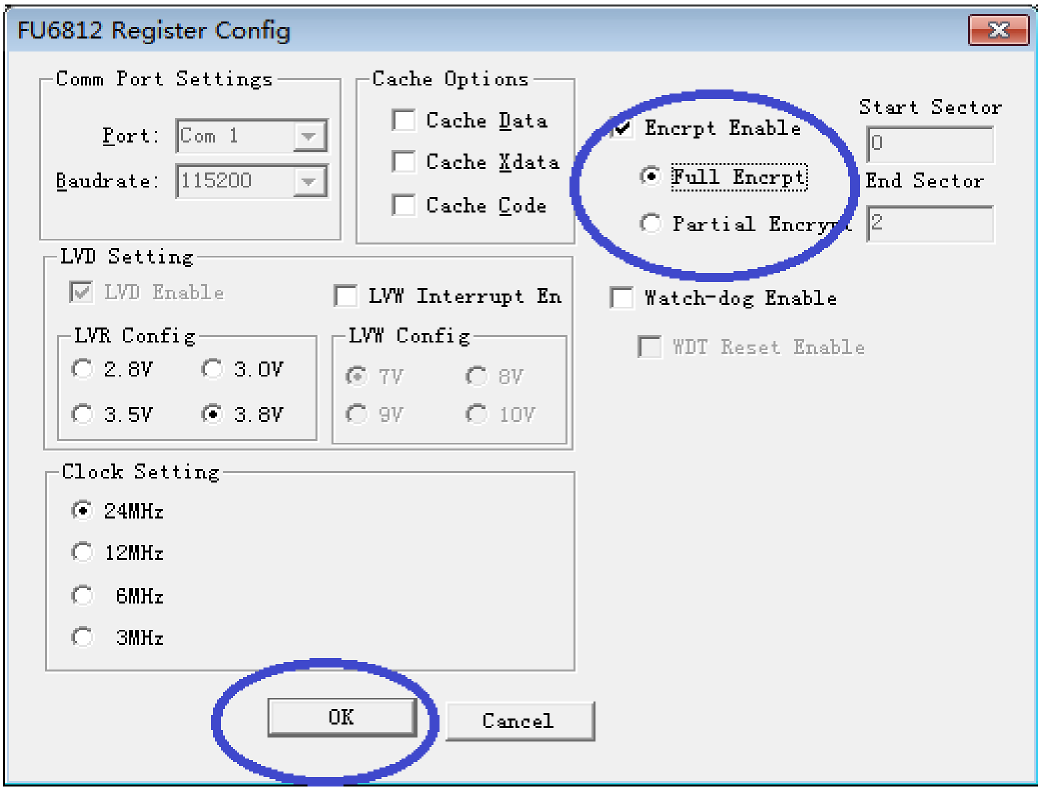
Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows
☑ Memory Display

Driver DLL: S8051.DLL Parameter: -pFU6812

Dialog DLL: TCYG.DLL Parameter: -pFU6812

OK Cancel Defaults Help

Figure 31-1 Step 1 of Code protection



FU6812 Register Config

Comm Port Settings
Port: Com 1
Baudrate: 115200

Cache Options
□ Cache Data
□ Cache Xdata
□ Cache Code

☑ Encrpt Enable
● Full Encrpt
○ Partial Encrypt

Start Sector: 0
End Sector: 2

LVD Setting
☑ LVD Enable □ LVW Interrupt En

LVR Config
○ 2.8V ○ 3.0V
○ 3.5V ● 3.8V

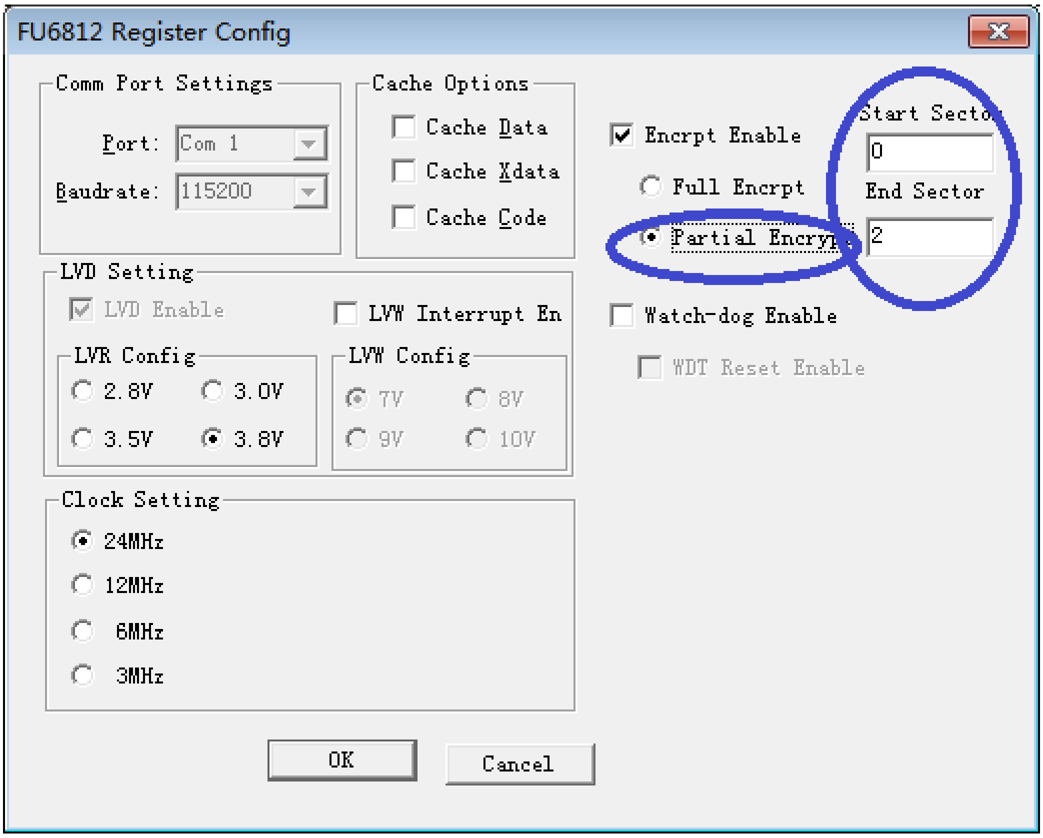
LVW Config
● 7V ○ 8V
○ 9V ○ 10V

□ Watch-dog Enable
□ WDT Reset Enable

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

OK Cancel

Figure 31-2 Code protection on full protection mode



FU6812 Register Config

Comm Port Settings
Port: Com 1
Baudrate: 115200

Cache Options
□ Cache Data
□ Cache Xdata
□ Cache Code

☑ Encrpt Enable
○ Full Encrpt
● Partial Encryp

Start Secto
0
End Sector
2

□ Watch-dog Enable
□ WDT Reset Enable

LVD Setting
☑ LVD Enable
□ LVW Interrupt En

LVR Config
○ 2.8V ○ 3.0V
○ 3.5V ● 3.8V

LVW Config
● 7V ○ 8V
○ 9V ○ 10V

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

OK Cancel

Figure 31-3 Code protection on local protection mode

FU6812/61 supports users to protect intellectual property rights of code burned into FLASH. Methods and steps are as follows:

Step 1:

Open the 8051 IDE tool, go to Target Options and select the Debug TAB before compiling. Select as shown in Figure 31-1 and click Settings to go to the next setting.

Step 2:

Select and set as shown in Figure 31-2, and click OK.Then compile the project and download the.bin file. After burning to FLASH, you can achieve code protection.

It should be noted that the chip has both full code protection mode and local code protection mode. As shown in Figure 31-2, the chip is set to full code protection mode, and all code in FLASH will be protected after setting. As shown in Figure 31-3, the protection mode is set to local code protection mode. After setting the protection mode, only protect the sectors from SECTOR 0 to END SECTOR. The last SECTOR is always protected in any case.

The size of each sector is 128 bytes.

32 Config register

32.1 CCFG customer config register

32.1.1 CCFG1:CK_RST_CFG

Table 32-1 CCFG1 (0x401E)

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

Bit	Name	Function
[7]	LVDENB	Low voltage reset detection enabled 0: enable. If VDD5 is lower than the threshold voltage set by LVRSEL, the system is reset. 1: closed Note: The low-voltage detection function is divided into two parts: one is low-voltage reset detection. Once VDD5 is detected to be lower than the threshold voltage set by LVRSEL, the chip will reset.Second, low-voltage alarm function. Once the system detects that VCC is lower than the threshold voltage set by LVWSEL, low-voltage alarm interruption will be generated.If any of the above functions are required to work, LVDENB must be set to 0 (this is, enable low-voltage reset detection function).And low voltage alarm interrupts are controled by low voltage alarm interrupt enablement.
[6]	LVWIE	VCC low voltage alarm interrupts enabling, and LVWSEL sets the VCC early warning voltage threshold. 0: disable. 1: enable. If low voltage alarm interrupt is needed to enable, it must also enable low voltage reset detection circuit (i.e. LVDENB=0).
[5]	WDTEN	Watch-dog enablement 0: Disable 1: Enable
[4:3]	RSV	Reserved
[2:1]	FCK_SEL	System clock frequency selection 00：24MHz 01：12MHz 10：6MHz 11：3MHz
[0]	RSV	Reserved

This register can be accessed directly by software.The recommended way is to set it up for initialization in the IDE tool first, and after setting it up, software doesn't have to write it again.

32.1.2 CCFG2: RST_MOD

This register is set only through IDE tools. After setting this register, relevant register values are compiled and generated to merge with ROM_CODE to generate burn file BIN file.This register is not writable in software, but its value can be read.

Table 32-2 CFG2 (0x401D)

Bit	7	6	5	4	3	2	1	0
Name	LVRSEL		WDTBTEN	WDTRSTEN	RSV		LVWSEL	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function

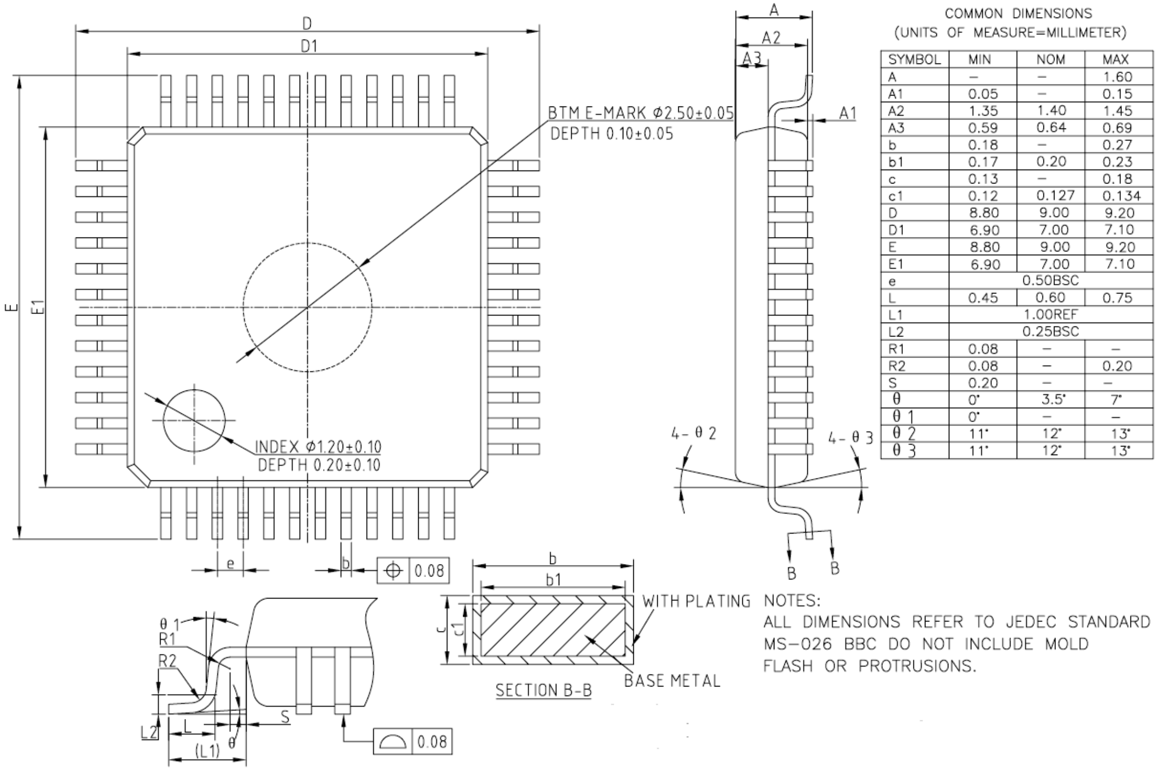
[7:6]	LVRSEL	Low voltage reset voltage selector. The low voltage reset detects the voltage value of VDD5.
		00: the corresponding VDD5 reset voltage is 2.8V;
		01: the corresponding VDD5 reset voltage is 3.0V;
		10: the corresponding VDD5 reset voltage is 3.5V;
		11: the corresponding VDD5 reset voltage is 3.8V.

[5]	WDTBTEN	1: BOOT is started when watch-dog is reset
[4]	WDTRSTEN	Enable watch-dog overflow reset: 1: digital reset is triggered after overflow
[3:2]	RSV	Reserved

[1:0]	LVWSEL	Low voltage warning voltage selector. Low voltage early warning detects the VCC voltage value.
		00: the corresponding VCC warning voltage is 7V;
		01: the corresponding VCC warning voltage is 8V;
		10: the corresponding VCC warning voltage is 9V;
		11: the corresponding VCC warning voltage is 10V.

33 Packaging information

33.1 LQFP48_7X7



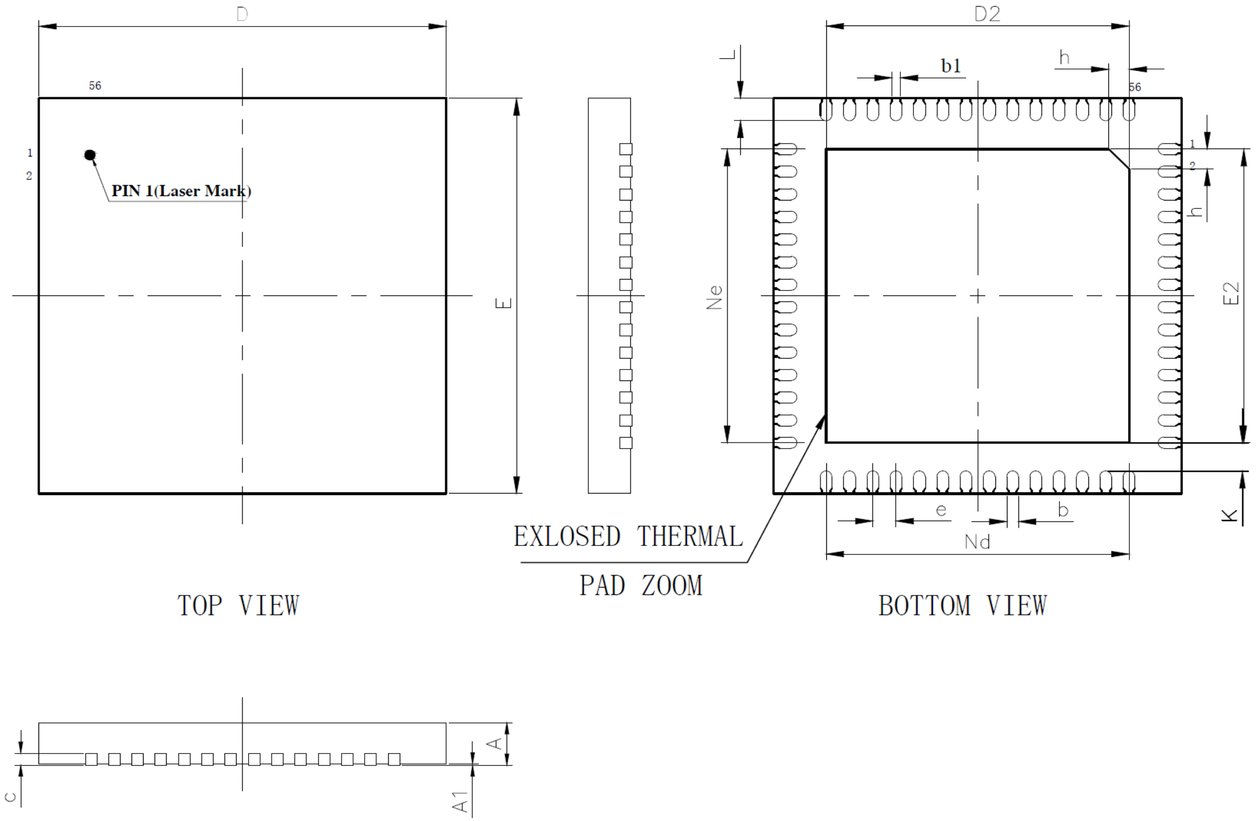
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
e	0.50BSC		
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ 1	0°	–	–
θ 2	11°	12°	13°
θ 3	11°	12°	13°

NOTES:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BBC DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

Figure 33-1 LQFP48_7X7 package size diagram

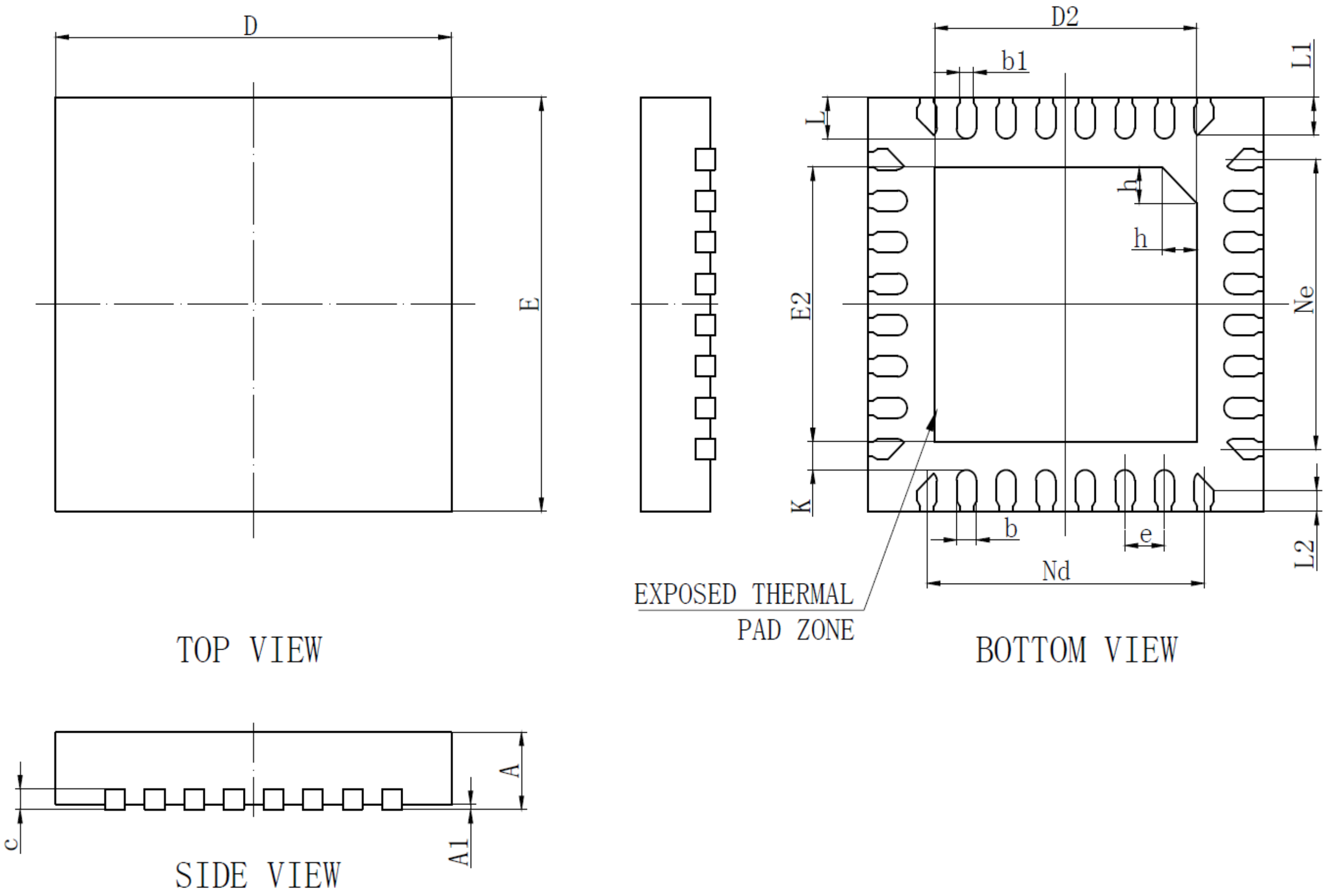
33.2 QFN56_7X7



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0. 70	0. 75	0. 80
A1	—	0. 02	0. 05
b	0. 15	0. 20	0. 25
b1	0. 14REF		
c	0. 18	0. 20	0. 25
D	6. 90	7. 00	7. 10
D2	5. 10	5. 20	5. 30
e	0. 40BSC		
Nd	5. 20BSC		
Ne	5. 20BSC		
E	6. 90	7. 00	7. 10
E2	5. 10	5. 20	5. 30
K	0. 20	--	--
L	0. 35	0. 40	0. 45
h	0. 30	0. 35	0. 40

Figure 33-2 Package size diagram of QFN56_7X7

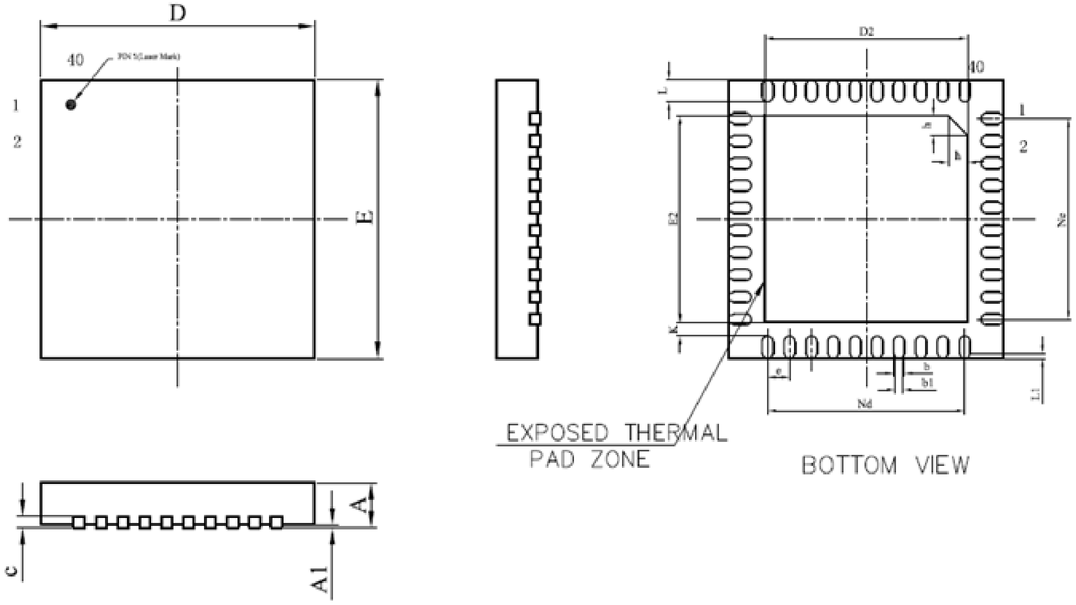
33.3 QFN32_4X4



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0. 70	0.75	0. 80
A1	0	0. 02	0. 05
b	0. 15	0.20	0. 25
b1	0.14REF		
c	0.18	0.20	0.25
D	3. 90	4. 00	4. 10
D2	2. 60	2. 65	2. 70
e	0. 40BSC		
Nd	2. 80BSC		
E	3. 90	4. 00	4. 10
E2	2. 60	2. 65	2. 70
Ne	2. 80BSC		
K	0. 20	-	-
L	0. 35	0. 40	0. 45
L1	0. 30	0. 35	0. 40
L2	0. 15	0. 20	0. 25
h	0. 30	0. 35	0. 40

Figure 33-3 QFN32 4mmx4mmx0.75mm package size diagram

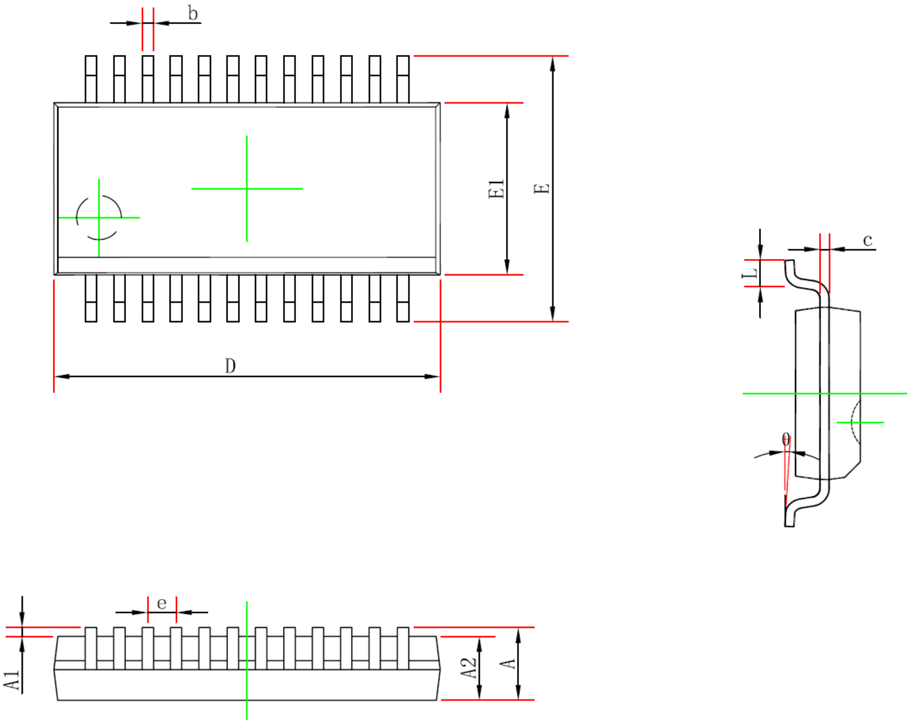
33.4 QFN40_5X5



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	—	0.02	0.05
b	0.15	0.20	0.25
b1	0.14REF		
c	0.18	0.20	0.25
D	4.90	5.00	5.10
D2	3.60	3.70	3.80
e	0.40BSC		
Nd	3.60BSC		
E	4.90	5.00	5.10
E2	3.60	3.70	3.80
Ne	3.60BSC		
L	0.35	0.40	0.45
L1	0.10REF		
K	0.20	–	--
h	0.30	0.35	0.40

Figure 33-4 Package size diagram of QFN40 5mm x 5mm x 0.4mm

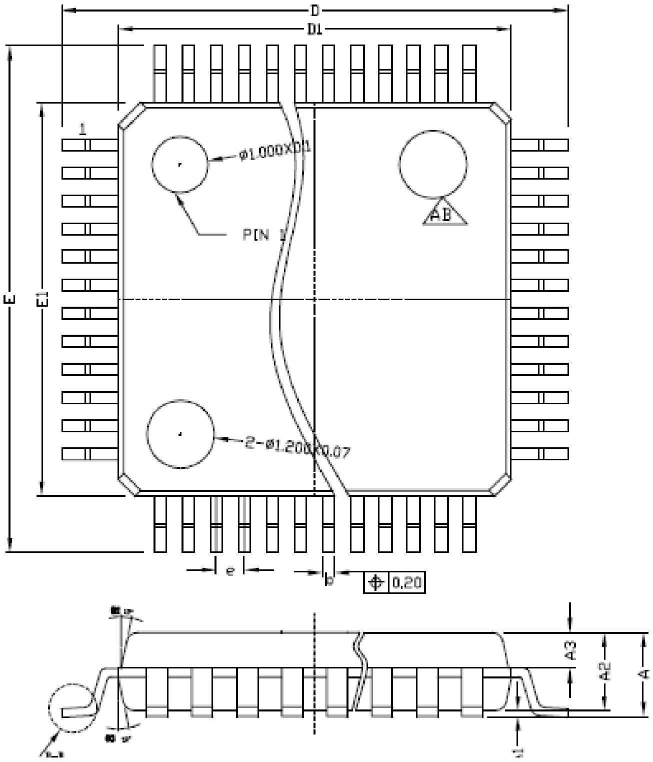
33.5 SSOP24_8.65x3.9

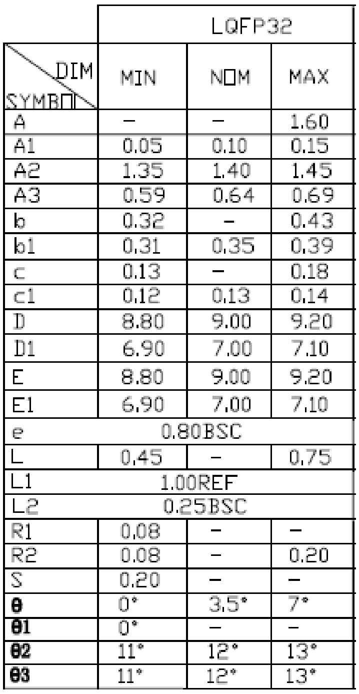


Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	**Max**	**Min**	**Max**
A	——	1. 750	——	0. 069
A1	0. 100	0. 250	0. 004	0. 010
A2	1. 250	——	0. 049	——
b	0. 203	0. 305	0. 008	0. 012
c	0. 102	0. 254	0. 004	0. 010
D	8. 450	8. 850	0. 333	0. 348
E1	3. 800	4. 000	0. 150	0. 157
E	5. 800	6. 200	0. 228	0. 244
e	0. 635 (BSC)		0. 025 (BSC)	
L	0. 400	1. 270	0. 016	0. 050
θ	0°	8°	0°	8°

Figure 33-4 SSOP24 8.65mmx3.9mm package size diagram

33.6 LQFP32_7X7





DIM SYMB□□	LQFP32		
	MIN	N□M	MAX
A	–	–	1.60
A1	0.05	0.10	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.32	–	0.43
b1	0.31	0.35	0.39
c	0.13	–	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
e	0.80BSC		
L	0.45	–	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

Figure 33-5 LQFP32_7X7 package size diagram

34 Order information

Table 34-1 Product Selection Guide

Type	MIPS (Peak)	FLASH (KB)	XRAM (KB)	Clock Source				Driver interface			Driving type			I2C/UART/SPI	DMA	GPIO	Timer	Analog peripherals								Lead-free	Package
				Internal fast clock	External fast clock	Internal slow clock	External slow clock	6 N Predriver	3P3N Predriver	Gate Driver	BLDC	SVPWM	FOC					ADC			DAC		VREF	Amplifier	Comparator		
																		Number	Channels	Bits	Number	Bits					
FU6812L	24	16	0.75	√	-	-	-	-	-	√	√	√	√	√	√	34	6	1	12	12	1	8	√	3	3	√	LQFP48 (7x7 mm)
FU6812N	24	16	0.75	√	-	-	-	-	-	√	√	√	√	√	√	20	5	1	7	12	1	8	√	1	2	√	QFN32 (4 x4 mm)
FU6812S	24	16	0.75	√	-	-	-	-	-	√	√	√	√	UART	√	12	5	1	5	12	1	8	√	1	2	√	SSOP24 (8.65 x3.9mm)
FU6861Q	24	16	0.75	√	-	-	-	√	-	-	√	√	√	√	√	32	5	1	12	12	1	8	√	3	3	√	QFN56 7 x7 (mm)
FU6861N	24	16	0.75	√	-	-	-	√	-	-	√	√	√	√	√	19	5	1	9	12	1	8	√	1	3	√	QFN40 (5x5 mm)
FU6861L	24	16	0.75	√	-	-	-	√	-	-	√	√	√	√	√	27	5	1	11	12	1	8	√	3	3	√	LQFP48 (7x7 mm)
FU6812P	24	16	0.75	√	-	-	-	-	-	√	√	√	√	UART	√	21	6	1	9	12	1	8	√	3	3	√	LQFP32 (7x7 mm)

Copyright Notice

Right to make changes —Fortior Technology (Shenzhen) Co., Ltd RSVs 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 should be aware that the personal computer field is the subject of many patents. Our customers should 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.

Fortior Technology(Shenzhen) Co.,Ltd.
Room203,2/F, Building No.11,Keji Central Road2,
SoftwarePark, High-Tech Industrial Park, Shenzhen, P.R. China 518057
Tel: 0755-26867710
Fax: 0755-26867715
URL: http://www.fortiortech.com

Contained herein