

Datasheet

Three-phase Motor Control MCU FU6361Q

Fortior Technology (Shenzhen) Co., Ltd

Contents

Explanation of Symbols.....	19
Abbreviations.....	20
1 System Introduction	22
1.1 Features	22
1.2 Applications	23
1.3 Overview	23
1.4 Functional Block Diagram	24
1.4.1 FU6361Q.....	24
1.5 Memory Organization	25
1.5.1 Program Memory	25
1.5.2 Data Memory.....	25
1.5.3 SFR	26
1.5.4 XSFR	27
2 Pin Definitions	30
2.1 FU6361Q QFN56 Pins.....	30
2.2 FU6361Q QFN56 Pinout Diagram	33
3 Package Information.....	34
3.1 QFN56_7X7	34
4 Ordering Information.....	35
5 Electrical Characteristics	36
5.1 Absolute Maximum Ratings	36
5.2 MR Sensor Electrical Characteristics.....	36
5.3 Global Electrical Characteristics.....	37
5.4 GPIO Electrical Characteristics	37
5.5 Pre-driver IO Electrical Characteristics	38
5.6 ADC Electrical Characteristics	38
5.7 VREF and VHALF Electrical Characteristics.....	39
5.8 Operational Amplifier Electrical Characteristics	39
5.9 BEMF Electrical Characteristics	40
5.10 OSC Electrical Characteristics.....	40
5.11 Reset Electrical Characteristics	40
5.12 LDO Electrical Characteristics	40
5.13 Package Thermal Resistance	40

6 Reset Control	41
6.1 Reset Source (RST_SRC)	41
6.2 Reset Enable	41
6.3 External Reset and Power-on Reset	41
6.4 Low Voltage Detector reset	41
6.5 Watchdog Timer Reset	41
6.6 Flash Error Detector Reset	41
6.7 Reset Registers	42
7 Interrupt Control	43
7.1 Interrupt Registers	43
7.1.1 IE (0xA8)	43
7.1.2 IP0 (0xB8)	43
7.1.3 IP1 (0xC0)	44
7.1.4 IP2 (0xC8)	44
7.1.5 IP3 (0xD8)	44
7.1.6 TCON (0x88)	45
7.2 Interrupt Summary	45
7.3 External Interrupt	46
8 I²C	47
8.1 I ² C Operations	48
8.1.1 Master Mode	48
8.1.2 Slave Mode	48
8.1.3 I ² C Interrupt Sources	49
8.2 I ² C Registers	50
8.2.1 I2C_CR (0x4028)	50
8.2.2 I2C_ID (0x4029)	50
8.2.3 I2C_DR (0x402A)	50
8.2.4 I2C_SR (0x402B)	51
9 SPI	53
9.1 SPI Introduction	53
9.2 SPI Operations	53
9.2.1 SPI Master Mode	53
9.2.1.1 Master Mode Configuration	53
9.2.2 SPI Slave Mode	54
9.2.2.1 Slave Mode Configurations	54

9.2.3 SPI Interrupt Sources	54
9.2.4 SPI Operating Modes	55
9.2.5 Serial Clock Timing.....	56
9.3 SPI Registers.....	58
9.3.1 SPI_CR0 (0x4030).....	58
9.3.2 SPI_CR1 (0x4031).....	58
9.3.3 SPI_CLK (0x4032)	59
9.3.4 SPI_DR (0x4033)	59
10 UART	60
10.1 UART Introductions.....	60
10.1.1 UART Mode0.....	60
10.1.2 UART Mode1.....	60
10.1.3 UART Mode2.....	60
10.1.4 UART Mode3.....	61
10.1.5 UART Interrupt Sources	61
10.2 UART Registers.....	61
10.2.1 UT_CR (0x98)	61
10.2.2 UT_DR (0x99).....	62
10.2.3 UT_BAUD (0x9A,0x9B).....	62
11 MDU.....	63
11.1 Multiplication Configurations	63
11.2 Division Configurations	63
11.3 Important Matters	63
11.4 MDU Registers	65
11.4.1 MDU_CR (0xC1)	65
11.4.2 MD_MBL (0xCA)	65
11.4.3 MD_MBH (0xCB).....	65
11.4.4 MD_MAL (0xC2).....	66
11.4.5 MD_MAH (0xC3).....	66
11.4.6 MD_DA0 (0xC4)	66
11.4.7 MD_DA1 (0xC5)	66
11.4.8 MD_DA2 (0xC6)	66
11.4.9 MD_DA3 (0xC7)	67
11.4.10 MD_DB0 (0xCC)	67
11.4.11 MD_DB1 (0xCD).....	67

12 PI	68
12.1 PI Operations	68
12.2 PI Registers.....	69
12.2.1 PI_LPF_CR (0xF9)	69
12.2.2 PI_EK (0xEA, 0xEB)	69
12.2.3 PI_UK (0xEC, 0xED)	69
12.2.4 PI_KP (0xEE, 0xEF)	70
12.2.5 PI_KI (0xF2, 0xF3)	70
12.2.6 PI_UKMAX (0xF4, 0xF5)	70
12.2.7 PI_UKMIN (0xF6, 0xF7)	71
13 LPF	72
13.1 LPF Operations	72
13.2 LPF Registers	72
13.2.1 PI_LPF_CR (0xF9)	72
13.2.2 LPF_K (0xDD)	72
13.2.3 LPF_X (0xDE, 0xDF)	72
13.2.4 LPF_Y (0xE6, 0xE7)	73
14 SMDU	74
14.1 SMDU Features	74
14.2 SMDU Instructions	74
14.2.1 SMDU Operations.....	74
14.2.2 16-bit Signed Multiplication with the Result Shifted Left by One-bit.....	74
14.2.3 16-bit Signed Multiplication	75
14.2.4 16-bit Unsigned Multiplication.....	75
14.2.5 32-bit/16-bit Unsigned Division.....	75
14.2.6 LPF	76
14.2.7 Coordinate Transformation (SIN/COS)	76
14.2.8 Arctangent	77
14.2.9 PI.....	78
14.2.9.1 PI Introduction	78
14.2.9.2 PI Features.....	78
14.2.9.3 PI Operations.....	78
14.3 SMDU Registers	80
14.3.1 MDU_CR (0xC1)	80
14.3.2 TIM234_CTRL(0x40f1)	80

14.3.3 MUL0_MA (0x03C8, 0x03C9).....	81
14.3.4 MUL0_MB (0x03CA, 0x03CB)	81
14.3.5 MUL0_MC (0x03CC, 0x03CD, 0x03CE, 0x03CF)	82
14.3.6 MUL1_MA (0x03C0, 0x03C1).....	82
14.3.7 MUL1_MB (0x03C2, 0x03C3).....	83
14.3.8 MUL1_MC (0x03C4, 0x03C5, 0x03C6, 0x03C7)	83
14.3.9 MUL2_MA (0x0370, 0x0371).....	84
14.3.10 MUL2_MB (0x0372, 0x0373)	84
14.3.11 MUL2_MC (0x0374, 0x0375, 0x0376, 0x0377).....	84
14.3.12 MUL3_MA (0x0368, 0x0369)	85
14.3.13 MUL3_MB (0x036A, 0x036B).....	85
14.3.14 MUL3_MC (0x036C, 0x036D, 0x036E, 0x036F)	85
14.3.15 DIV0_DA (0x03B4, 0x03B5, 0x03B6, 0x03B7).....	86
14.3.16 DIV0_DB (0x03B8, 0x03B9).....	87
14.3.17 DIV0_DQ (0x03BA, 0x03BB, 0x03BC, 0x03BD)	87
14.3.18 DIV0_DR (0x03BE, 0x03BF).....	87
14.3.19 DIV1_DA (0x03A8, 0x03A9, 0x03AA, 0x03AB).....	88
14.3.20 DIV1_DB (0x03AC, 0x03AD).....	88
14.3.21 DIV1_DQ (0x03AE, 0x03AF, 0x03B0, 0x03B1)	89
14.3.22 DIV1_DR (0x03B2, 0x03B3).....	89
14.3.23 DIV2_DA (0x035C, 0x035D, 0x035E, 0x035F)	89
14.3.24 DIV2_DB (0x0360, 0x0361).....	90
14.3.25 DIV2_DQ (0x0362, 0x0363, 0x0364, 0x0365)	90
14.3.26 DIV2_DR (0x0366, 0x0367).....	91
14.3.27 DIV3_DA (0x0350, 0x0351, 0x0352, 0x0353)	91
14.3.28 DIV3_DB (0x0354, 0x0355).....	92
14.3.29 DIV3_DQ (0x0356, 0x0357, 0x0358, 0x0359).....	92
14.3.30 DIV3_DR (0x035A, 0x035B)	93
14.3.31 SCAT0_COS (0x0346, 0x0347)	93
14.3.32 SCAT0_SIN (0x0348, 0x0349).....	93
14.3.33 SCAT0_THE (0x034A, 0x034B)	94
14.3.34 SCAT0_RES1 (0x034C, 0x034D)	94
14.3.35 SCAT0_RES2 (0x034E, 0x034F)	94
14.3.36 SCAT1_COS (0x033C, 0x033D).....	95
14.3.37 SCAT1_SIN (0x033E, 0x033F).....	95

14.3.38 SCAT1_THE (0x0340, 0x0341).....	95
14.3.39 SCAT1_RES1 (0x0342, 0x0343)	96
14.3.40 SCAT1_RES2 (0x0344, 0x0345)	96
14.3.41 SCAT2_COS (0x0332, 0x0333)	96
14.3.42 SCAT2_SIN (0x0334, 0x0335).....	97
14.3.43 SCAT2_THE (0x0336, 0x0337).....	97
14.3.44 SCAT2_RES1 (0x0338, 0x0339)	97
14.3.45 SCAT2_RES2 (0x033A, 0x033B).....	98
14.3.46 SCAT3_COS (0x0328, 0x0329)	98
14.3.47 SCAT3_SIN (0x032A, 0x032B)	98
14.3.48 SCAT3_THE (0x032C, 0x032D)	99
14.3.49 SCAT3_RES1 (0x032E, 0x032F)	99
14.3.50 SCAT3_RES2 (0x0330, 0x0331)	99
14.3.51 LPF0_K (0x03F8, 0x03F9)	100
14.3.52 LPF0_X (0x03FA 0x03FB)	100
14.3.53 LPF0_Y (0x03FC, 0x03FD, 0x03FE, 0x03FF)	100
14.3.54 LPF1_K (0x03F0, 0x03F1)	101
14.3.55 LPF1_X (0x03F2, 0x03F3)	101
14.3.56 LPF1_Y (0x03F4, 0x03F5, 0x03F6, 0x03F7)	101
14.3.57 LPF2_K (0x03A0, 0x03A1)	102
14.3.58 LPF2_X (0x03A2, 0x03A3)	102
14.3.59 LPF2_Y (0x03A4, 0x03A5, 0x03A6, 0x03A7)	103
14.3.60 LPF3_K (0x0398, 0x0399)	103
14.3.61 LPF3_X (0x039A, 0x039B)	104
14.3.62 LPF3_Y (0x039C, 0x039D, 0x039E, 0x039F)	104
14.3.63 PIO_KP (0x03E0, 0x03E1)	105
14.3.64 PIO_EK1 (0x03E2, 0x03E3)	105
14.3.65 PIO_EK (0x03E4, 0x03E5)	105
14.3.66 PIO_KI (0x03E6, 0x03E7)	106
14.3.67 PIO_UKH (0x03E8, 0x03E9)	106
14.3.68 PIO_UKL (0x03EA, 0x03EB)	106
14.3.69 PIO_UKMAX (0x03EC, 0x03ED)	107
14.3.70 PIO_UKMIN (0x03EE, 0x03EF)	107
14.3.71 PI1_KP (0x03D0, 0x03D1)	107
14.3.72 PI1_EK1 (0x03D2, 0x03D3)	108

14.3.73 PI1_EK (0x03D4, 0x03D5)	108
14.3.74 PI1_KI(0x03D6, 0x03D7)	108
14.3.75 PI1_UKH (0x03D8, 0x03D9)	109
14.3.76 PI1_UKL (0x03DA, 0x03DB)	109
14.3.77 PI1_UKMAX (0x03DC, 0x03DD)	109
14.3.78 PI1_UKMIN (0x03DE, 0x03DF)	110
14.3.79 PI2_KP (0x0388, 0x0389)	110
14.3.80 PI2_EK1 (0x038A, 0x038B).....	110
14.3.81 PI2_EK (0x038C, 0x038D).....	111
14.3.82 PI2_KI (0x038E, 0x038F)	111
14.3.83 PI2_UKH (0x0390, 0x0391).....	111
PI2_UKL (0x0392, 0x0393).....	112
14.3.84 PI2_MAX (0x0394, 0x0395)	112
14.3.85 PI2_MIN (0x0396, 0x0397)	112
14.3.86 PI3_KP (0x0378, 0x0379)	113
14.3.87 PI3_EK1 (0x037A, 0x037B).....	113
14.3.88 PI3_EK (0x037C, 0x037D).....	113
14.3.89 PI3_KI (0x037E, 0x037F)	114
14.3.90 PI3_UKH (0x0380, 0x0381)	114
14.3.91 PI3_UKL (0x0382, 0x0383).....	114
14.3.92 PI3_UKMAX (0x0384, 0x0385).....	115
14.3.93 PI3_UKMIN (0x0386, 0x0387)	115
15 FOC/SVPWM	116
15.1 FOC/SVPWM Overview	116
15.1.1 FOC Introduction	116
15.1.2 Reference Voltage (VREF) Input	116
15.1.3 PI Controller.....	117
15.1.4 Coordinate Transformation	117
15.1.4.1 Inverse Park Transformation.....	117
15.1.4.2 Inverse Clarke Transformation.....	118
15.1.4.3 Clarke Transformation	118
15.1.4.4 Park Transformation	118
15.1.5 SVPWM.....	119
15.1.5.1 Continuous SVPWM	120
15.1.5.2 Discontinuous SVPWM.....	120

15.1.6 Overmodulation	121
15.1.7 Deadtime Compensation	121
15.1.8 Current and Voltage Sampling	121
15.1.8.1 Single-shunt Current Sampling Mode	121
15.1.8.2 Dual/Triple-shunt Current Sampling Mode	122
15.1.8.3 Current Sampling Offset	123
15.1.9 Angle Mode	123
15.1.9.1 Forced Ramping Angle	124
15.1.9.2 Forced Pulling Angle	124
15.1.9.3 Estimator Output Angle	125
15.1.10 Motor Real-time Parameters	126
15.1.10.1 Tailwind/headwind Detection	127
15.1.10.2 BEMF Detection	127
15.1.10.3 Motor Power	127
15.1.11 FG Generation	127
15.2 FOC Registers	128
15.2.1 FOC_CR1 (0x40A0)	128
15.2.2 FOC_CR2 (0x40A1)	128
15.2.3 FOC_TSMIN (0x40A2)	129
15.2.4 FOC_TGLI (0x40A3)	129
15.2.5 FOC_TBLO (0x40A4)	130
15.2.6 FOC_TRGDLY (0x40A5)	130
15.2.7 FOC_CSO (0x40A6, 0x40A7)	130
15.2.8 FOC__RTHESTEP (0x40A8, 0x40A9)	131
15.2.9 FOC_RTHEACC (0x40AA, 0x40AB)	131
15.2.10 FOC_RTHECNT (0x40AC)	132
15.2.11 FOC_THECOR (0x40AD) (Shared with BLDC Motors)	132
15.2.12 FOC_THECOMP (0x40AE, 0x40AF)	132
15.2.13 FOC_DMAX (0x40B0, 0x40B1)	132
15.2.14 FOC_DMIN (0x40B2, 0x40B3)	133
15.2.15 FOC_QMAX (0x40B4, 0x40B5)	133
15.2.16 FOC_QMIN (0x40B6, 0x40B7)	133
15.2.17 FOC__UD (0x40B8, 0x40B9)	134
15.2.18 FOC__UQ (0x40BA, 0x40BB)	134
15.2.19 FOC__ID (0x40BC, 0x40BD)	134

15.2.20 FOC__IQ (0x40BE, 0x40BF)	135
15.2.21 FOC__IBET (0x40C0, 0x40C1)	135
15.2.22 FOC__VBET (0x40C2, 0x40C3)	135
15.2.23 FOC__VALP (0x40C4, 0x40C5)	136
15.2.24 FOC__IC (0x40C6, 0x40C7)	136
15.2.25 FOC__IB (0x40C8, 0x40C9)	136
15.2.26 FOC__IA (0x40CA, 0x40CB).....	137
15.2.27 FOC__THETA (0x40CC, 0x40CD)	137
15.2.28 FOC__ETHETA (0x40CE, 0x40CF)	137
15.2.29 FOC__EALP (0x40D0, 0x40D1).....	138
15.2.30 FOC__EBET (0x40D2, 0x40D3).....	138
15.2.31 FOC__EOME (0x40D4, 0x40D5).....	138
15.2.32 FOC__ESQU (0x40D6, 0x40D7).....	139
15.2.33 FOC__POW (0x40D8, 0x40D9).....	139
15.2.34 FOC__EKP (0x4074, 0x4075) (Shared with BLDC Motors).....	139
15.2.35 FOC__EKI (0x4076, 0x4077) (Shared with BLDC Motors).....	140
15.2.36 FOC__EBMFK (0x407C, 0x407D) (Shared with BLDC Motors).....	140
15.2.37 FOC__KSLIDE (0x4078, 0x4079) (Shared with BLDC Motors).....	140
15.2.38 FOC__EKLPMIN (0x407A, 0x407B) (Shared with BLDC Motors).....	141
15.2.39 FOC__OMEKLPF (0x407E, 0x407F)	141
15.2.40 FOC__FBASE (0x4080, 0x4081).....	141
15.2.41 FOC__EFREQACC (0x4082, 0x4083) (Shared with BLDC Motors).....	142
15.2.42 FOC__EFREQMIN (0x4084 0x4085) (Shared with BLDC Motors)	142
15.2.43 FOC__EFREQHOLD (0x4086, 0x4087) (Shared with BLDC Motors)	142
15.2.44 FOC__EK3 (0x4088, 0x4089).....	143
15.2.45 FOC__EK4 0x408A, 0x408B)	143
15.2.46 FOC__EK1 (0x408C, 0x408D)	144
15.2.47 FOC__EK2 (0x408E, 0x408F).....	144
15.2.48 FOC__IDREF (0x4090, 0x4091) (Shared with BLDC Motors)	144
15.2.49 FOC__IQREF (0x4092, 0x4093) (Shared with BLDC Motors)	145
15.2.50 FOC__DQKP (0x4094, 0x4095) (Shared with BLDC Motors).....	145
15.2.51 FOC__DQKI (0x4096, 0x4097) (Shared with BLDC Motors).....	145
15.2.52 FOC__UDCFLT (0x4098, 0x4099)	146
15.2.53 FOC__CR3 (0x40ee)	146
15.2.54 FOC__DKP (0x409c, 0x409d)	147

15.2.55 FOC_DKI (0x409e, 0x409f)	147
15.2.56 FOC__IAMAX (0x40da, 0x40db)	147
15.2.57 FOC__IBMAX (0x40dc, 0x40dd)	148
15.2.58 FOC__ICMAX (0x40de, 0x40df)	148
15.2.59 FOC__EMF (0x40e0, 0x40e1)	149
15.2.60 FOC__UDCPS (0x40e2, 0x40e3)	149
15.2.61 FOC__UQCPS (0x40e4, 0x40e5)	149
15.2.62 FOC__UQEX (0x40e6, 0x40e7)	150
15.2.63 FOC_ID_LPFK (0x40e8)	150
15.2.64 FOC_IQ_LPFK (0x40e9)	150
15.2.65 FOC_KFG (0x40ea, 0x40eb)	151
16 Timer1.....	152
16.1 Timer1 Operations	152
16.1.1 Timer1 Counter Module	153
16.1.1.1 Prescaler	153
16.1.1.2 Base Timer	153
16.1.1.3 Reload Timer	154
16.1.2 Input Filtering and Sampling	155
16.1.2.1 Filtering	155
16.1.2.2 Sampling	155
16.1.3 Position Detection Event	157
16.1.4 Write Timing Interrupt	158
16.1.5 Timer1 Interrupt	158
16.2 Square-wave Control Based BLDC Motor	159
16.2.1 Six-Step Phase Commutation of Square Wave Control	160
16.2.2 Working Principle of BLDC Motor	161
16.2.2.1 60° Commutation Base Time	161
16.2.2.2 Phase Commutation	161
16.2.2.3 Forced Commutation at 60°	162
16.2.2.4 Diode Freewheeling Masking	162
16.2.2.5 Angle of ZCP to Phase Commutation (Delayed Commutation)	163
16.2.2.6 Cycle-by-cycle Current Limiting	163
16.2.3 BLDC Motor Debugging	163
16.2.3.1 Comparator Debugging	163
16.2.3.2 Debugging on Diode Freewheeling Masking and Phase Commutation	164

16.3 Phase Commutation Signals by ADC	164
16.4 Timer1 Registers	165
16.4.1 TIM1_CR0 (0x4068)	165
16.4.2 TIM1_CR1 (0x4069)	165
16.4.3 TIM1_CR2 (0x406A)	166
16.4.4 TIM1_CR3 (0x406B)	166
16.4.5 TIM1_CR4 (0x406C)	167
16.4.6 TIM1_IER (0x406D)	167
16.4.7 TIM1_SR (0x406E).....	168
16.4.8 TIM1_BCOR (0x4070, 0x4071).....	169
16.4.9 TIM1_DBRx (x=1~7)(0x4074+2*x, 0x4075+2*x).....	169
16.4.10 TIM1__BCNTR (0x4082, 0x4083).....	171
16.4.11 TIM1__BCCR (0x4084, 0x4085)	171
16.4.12 TIM1__BARR (0x4086, 0x4087)	172
16.4.13 TIM1__RARR (0x4088, 0x4089)	172
16.4.14 TIM1__RCNTR (0x408A, 0x408B)	172
16.4.15 TIM1__ITRIP (0x4098, 0x4099).....	173
16.4.16 TIM1__UCOP (0x408C, 0x408D).....	173
16.4.17 TIM1__UFLP (0x408E, 0x408F)	173
16.4.18 TIM1__URES (0x4090, 0x4091)	174
16.4.19 TIM1__UIGN (0x4092, 0x4093)	174
16.4.20 TIM1_KF (0x4094, 0x4095).....	174
16.4.21 TIM1_KR (0x4096, 0x4097).....	175
16.4.22 EXT0 (0x40f0).....	175
17 Timer2.....	176
17.1 Timer2 Instructions.....	176
17.1.1 Prescaler	176
17.1.2 Reading, Writing and Counting of TIM2__CNTR	177
17.1.3 Output Mode	177
17.1.3.1 Reading and Writing of TIM2__ARR/TIM2__DR	177
17.1.3.2 High-/Low-level Output Mode	177
17.1.3.3 PWM Generation	177
17.1.3.4 Interrupt Event.....	178
17.1.4 Input Signal Filtering and Edge Detection	178
17.1.5 Input Capture Mode	179

17.1.6 Input Counter Mode	180
17.1.7 QEP&RSD Mode.....	181
17.1.7.1 RSD Comparator Sampling	182
17.1.8 Step Mode	183
17.2 Timer2 Registers	185
17.2.1 TIM2_CR0(0xA1)	185
17.2.2 TIM2_CR1(0xA9)	186
17.2.3 PI_LPF_CR (0xF9)	187
17.2.4 TIM2__CNTR(0xAA,0xAB)	187
17.2.5 TIM2__DR(0xAC,0xAD)	188
17.2.6 TIM2__ARR(0xAE,0xAF).....	188
18 Timer3/Timer4.....	189
18.1 Timer3/Timer4 Instructions.....	189
18.1.1 Prescaler	189
18.1.2 Reading, Writing and Counting of TIMx__CNTR.....	189
18.1.3 Output Mode	190
18.1.3.1 High-/Low-level Output Mode.....	190
18.1.3.2 PWM Generation	190
18.1.3.3 Interrupt Event.....	190
18.1.4 Input Signal Filtering and Edge Detection	191
18.1.5 Input Capture Mode	191
18.1.6 FG Generation (Timer4).....	193
18.2 Timer3/Timer4 Registers	193
18.2.1 TIMx_CR0(0x9C/0x9E) (x=3/4).....	193
18.2.2 TIMx_CR1(0x9D/0x9F) (x=3/4)	194
18.2.3 TIMx__CNTR(0xA2,0xA3/0x92,0x93) (x=3/4)	195
18.2.4 TIMx__DR(0xA4,0xA5/0x94,0x95) (x=3/4)	195
18.2.5 TIMx__ARR(0xA6,0xA7/0x96,0x97) (x=3/4)	195
19 Systick	196
19.1 Systick Instructions	196
19.2 Systick Registers.....	196
19.2.1 DRV_SR (0x4061)	196
19.2.2 SYST_ARR (0x4064,0x4065)	196
20 Driver	197
20.1 Driver Instructions	197

20.1.1 Driver Introduction	197
20.1.2 Output Control Module	197
20.1.2.1 Count and Compare Module.....	198
20.1.2.2 Deadtime Module	198
20.1.2.3 Enable and Polarity of Output Signals.....	199
20.1.2.4 Main Output Enable (MOE).....	200
20.1.2.5 Interrupt	201
20.1.3 6N Pre-driver Mode.....	202
20.2 Driver Registers.....	203
20.2.1 DRV_CR (0x4062).....	203
20.2.2 DRV_SR(0x4061).....	203
20.2.3 DRV_OUT (0xF8)	204
20.2.4 DRV_CMRR (0x405C, 0x405D)	204
20.2.5 DRV_ARR (0x405E, 0x405F)	206
20.2.6 DRV_COMR (0x405A, 0x405B).....	206
20.2.7 DRV_DR (0x4058, 0x4059).....	207
20.2.8 DRV_DTR (0x4060)	207
21 WDT	208
21.1 WDT Notes	208
21.2 WDT Operations.....	208
21.3 WDT Registers.....	209
21.3.1 WDT_CR (0x4026).....	209
21.3.2 WDT_REL (0x4027)	209
22 RTC and Clock Calibration.....	210
22.1 RTC Functional Block Diagram	210
22.2 RTC Operations	210
22.3 RTC Registers	210
22.3.1 RTC_TM (0x402c,0x402d).....	210
22.3.2 RTC_STA (0x402e).....	210
22.4 Clock Calibration	212
22.4.1 Introduction.....	212
22.4.2 Clock Calibration Register.....	212
22.4.2.1 CAL_CR (0x4044,0x4045)	212
23 IO	213
23.1 IO Introduction.....	213

23.2 IO Operations	213
23.3 IO Registers	214
23.3.1 P0_OE (0xFC)	214
23.3.2 P1_OE (0xFD)	214
23.3.3 P2_OE (0xFE).....	214
23.3.4 P3_OE (0xFF).....	214
23.3.5 P4_OE (0xE9)	215
23.3.6 P1_AN (0x4050)	215
23.3.7 P2_AN (0x4051)	216
23.3.8 P3_AN (0x4052)	216
23.3.9 P0_PU (0x4053)	216
23.3.10 P1_PU (0x4054)	216
23.3.11 P2_PU (0x4055)	217
23.3.12 P3_PU (0x4056)	217
23.3.13 P4_PU (0x4057)	217
23.3.14 PH_SEL (0x404C).....	217
23.3.15 P0 (0x80)/P1 (0x90)/P2 (0xA0)/P3 (0xB0)/P4(0xE8).....	218
24 ADC.....	219
24.1 ADC Block Diagram	219
24.2 ADC Operations.....	219
24.2.1 Sequential Sampling Mode.....	219
24.2.2 ADC Triggered Sampling Mode.....	220
24.3 ADC Registers.....	222
24.3.1 ADC_CR (0x4039).....	222
24.3.2 ADC_MASK={ADC_MASKH,ADC_MASKL} (0x4036~0x4037)	222
24.3.3 ADC_MASKH (0x4036)	223
24.3.4 ADC_SCYCL (0x4038)	223
24.3.5 ADC0_DR={ADC0_DRH,ADC0_DRL} (0x0300~0x0301)	224
24.3.6 ADC1_DR={ADC1_DRH,ADC1_DRL} (0x0302~0x0303)	224
24.3.7 ADC2_DR={ADC2_DRH,ADC2_DRL} (0x0304~0x0305)	224
24.3.8 ADC3_DR={ADC3_DRH,ADC3_DRL} (0x0306~0x0307)	225
24.3.9 ADC4_DR={ADC4_DRH,ADC4_DRL} (0x0308~0x0309)	225
24.3.10 ADC5_DR={ADC5_DRH,ADC5_DRL} (0x030A~0x030B).....	226
24.3.11 ADC6_DR={ADC6_DRH,ADC6_DRL} (0x030C~0x030D)	226
24.3.12 ADC7_DR={ADC7_DRH,ADC7_DRL} (0x030E~0x030F)	226

24.3.13 ADC8_DR={ADC8_DRH,ADC8_DRL} (0x0310~0x0311)	227
24.3.14 ADC9_DR={ADC9_DRH,ADC9_DRL} (0x0312~0x0313)	227
24.3.15 ADC10_DR={ADC10_DRH,ADC10_DRL} (0x0314~0x0315)	227
24.3.16 ADC11_DR={ADC11_DRH,ADC11_DRL} (0x0316~0x0317)	228
25 DAC	229
25.1 DAC Functional Block Diagram.....	229
25.2 DAC Registers	230
25.2.1 DAC_CR (0x4035).....	230
25.2.2 DAC_DR (0x404B)	230
26 DMA.....	231
26.1 DMA Instructions	231
26.2 DMA Registers.....	232
26.2.1 DMA0_CR0 (0x403A)	232
26.2.2 DMA1_CR0 (0x403B)	232
26.2.3 DMA0_CR1 (0x403C)	233
26.2.4 DMA1_CR1 (0x403E)	234
27 VREF.....	235
27.1 VREF Instructions	235
27.2 VREF Register	236
27.2.1 VREF_VHALF_CR(XRAM: 0x404F)	236
28 VHALF	237
28.1 VHALF Instructions.....	237
28.2 VHALF Register.....	237
29 Operational Amplifier	238
29.1 Operational Amplifier Instructions	238
29.1.1 Bus Current Sampling Operational Amplifier (AMP0)	238
29.1.2 Phase Current Operational Amplifier (AMP1/AMP2).....	239
29.1.2.1 AMP1.....	239
29.1.2.2 AMP2.....	239
29.1.3 PGA Differential Input Mode	240
29.2 Operational Amplifier Registers.....	242
29.2.1 AMP_CR (0x404E).....	242
29.2.2 CMP_AMP (0x40f2)	242
29.2.3 TSD_ADJ (0x40f3)	243
30 Comparator	244

30.1 Comparator Introduction	244
30.1.1 Comparator CMP3	244
30.1.1.1 Overcurrent Protection (OCP)	246
30.1.1.2 Cycle-by-cycle Current Limiting	247
30.1.2 Comparator CMP4	248
30.1.3 Comparator CMP0	249
30.1.4 Comparator Sampling	254
30.1.5 Comparator Output	255
30.2 Comparator Registers	256
30.2.1 CMP_CR0 (0xD5)	256
30.2.2 CMP_CR1 (0xD6)	256
30.2.3 CMP_CR2 (0xDA)	257
30.2.4 CMP_CR3 (0xDC)	259
30.2.5 CMP_SAMR(0x40AD)	259
30.2.6 CMP_SR (0xD7)	260
30.2.7 EVT_FILT (0xD9)	261
31 Power Supply	262
31.1 LDO Introduction	262
31.1.1 LDO Operations	262
31.2 Low Voltage Detector (LVD)	264
31.2.1 LVD Introduction	264
31.2.2 LVD Registers	264
31.2.2.1 CCFG2:RST_MOD (0x401D)	264
31.2.2.2 CCFG1:CK_RST_CFG (0x401E)	265
31.2.2.3 LVSR(0xDB)	265
32 Flash	266
32.1 Flash Introduction	266
32.2 FLA_CR (0x85)	266
32.3 FLA_KEY (0x84)	266
32.4 Flash Operations	267
33 CRC	268
33.1 CRC Functional Block Diagram	268
33.2 CRC16 Polynomial	268
33.3 CRC16 Logic Diagram	268
33.4 CRC Operations	269

33.4.1 CRC Calculation of Single Byte	269
33.4.2 CRC Calculation of ROM Sector	269
33.5 CRC Registers	270
33.5.1 CRC_CR (0x4022)	270
33.5.2 CRC_DIN (0x4021)	271
33.5.3 CRC_DR (0x4023)	271
33.5.4 CRC_BEG (0x4024)	272
33.5.5 CRC_CNT (0x4025)	272
34 Sleep Mode	273
34.1 Introduction	273
34.2 Sleep Mode Register	274
34.2.1 PCON (0x87)	274
35 Code Protection	275
36 Magnetoresistance (MR) Sensor	277
36.1 Introduction	277
36.2 Instructions	277
36.3 Functional Block Diagram	278
36.4 Application Recommendation	279
37 Configuration Registers	280
37.1 CCFG	280
37.1.1 CCFG1: CK_RST_CFG	280
37.1.2 CCFG2: RST_MOD	281
38 Revision History	282

Explanation of Symbols

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

Abbreviations

ADC	Analog to Digital Convertor
BEMF	Back Electromotive Force
BLDC	Brushless Direct Current
CRC	Cyclic Redundancy Check
DAC	Digital to Analog Convertor
DMA	Direct Memory Access
EEPROM	Electrically Erasable Programmable Read Only Memory
FG	Frequency Generator
FICE	Fortior Interactive Connectivity Establishment
FOC	Field Oriented Control
FOSC	Fast Oscillator
GPIO	General Purpose Input Output
IC	Integrated Circuit
I ² C	Internal Integrated Circuit
IRAM	Internal RAM
IDE	Integrated Development Environment
LDO	Low Dropout Regulator
LIN	Local Interconnect Network
LPF	Low Pass Filter
LVD	Low Voltage Detection
MDU	Multiplication Division Unit
ME	Motor Engine
MSB	Most Significant Bit
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NC	Not Connected
PFC	Power Factor Correction
PGA	Programmable Gain Amplifier
PI/PID	Proportional Integral/Proportional Integral Derivative
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
QEP	Quadrature Encoder Pulse
RAM	Random Access Memory
RMW	Read Modified Write

ROM	Read Only Memory
RSD	Rotating State Detection
RTC	Real Time Clock
SCL	Serial Clock Line
SDA	Serial Data Line
SFR	Special Function Register
SMO	Sliding Mode Observer
SOSC	Slow Oscillator
SPI	Serial Peripheral Interface
SVPWM	Space Vector PWM
TSD	Temperature Sensor Detect
UART	Universal Asynchronous Receiver/Transmitter
WDT	Watch Dog Timer
XRAM	External RAM
XSFR	External SFR

1 System Introduction

1.1 Features

- Power supply:
 - Single-power supply mode: When VCC_MODE = 0, external power supply 5V~24V is connected to VCC pin, and internal LDO supplies VDD5 voltage.
 - Dual-power supply mode: When VCC_MODE = 1, external power supply 1 (5V~36V) is connected to VCC pin, and external power supply 2 (5V) is connected to VDD5 pin. $VCC \geq VDD5$.
- Dual core: 8051 core and ME core
- An instruction cycle mostly takes 1 or 2 system clock cycle(s)
- 16kB Flash ROM with CRC, self-program and code protection
- 256 bytes IIRAM and 768 bytes XRAM
- ME: Core integrating LPF module, PI regulator, BLDC module, FOC module, MDU auxiliary computing module
- 1T 16x16 multiplier, 16T 32/16 divider
- 16 interrupt sources with 4 configurable priority levels
- 30*GPIOs
- Timers:
 - 2*Programmable timers with capture feature
 - 1*QEP decoding programmable timer
 - 1*BLDC motor dedicated timer
 - 1*General-purpose timer
 - 1*RTC
- Communication interfaces:
 - 1*SPI
 - 1*I²C
 - 2*UARTs, supporting single-wire mode
 - Dual-channel DMA: supporting data transmission via I²C/SPI/UART
- Analog Peripherals:
 - 12-bit ADC, operating with 1μs conversion time and internal VREF or external VREF selectable as reference voltage
 - Number of ADC channels: 12
 - Internal VREF. 3V, 4V, 4.5V and VDD5 can be selected as the internal reference.
 - Internal VHALF, with 1/2 VREF as the internal reference
 - Three standalone operational amplifiers

- 3-channel analog comparator
- Single-channel 9-bit DAC (DAC0)
- Driver Type:
 - 6N Pre-driver
- FOC module supports single/dual/triple-shunt current sampling
- FOC module supports overmodulation
- Automatic commutation, cycle-by-cycle current limiting and Hall/BEMF-based position sensing for BLDC motor control
- System clock
 - Built-in 24MHz high-speed oscillator
 - Built-in 32.8kHz low-speed oscillator
- WDT
- LVD
- TSD
- Two-wire FICE protocol based in-circuit emulation
- Magnetoresistance (MR) angle sensor

1.2 Applications

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

Applications: robot joints, power tools, low-voltage servos, tripod heads, etc.

1.3 Overview

The high-performance motor drive chip incorporates ME core and 8051 core. ME core implements motor control in real time and 8051 core is used for parameter configuration and routine processing. The dual cores work in parallel to achieve high-performance motor control. The chip integrates high-speed operational amplifiers, comparators, high-speed ADC, multiplier/divider, CRC, SPI, I²C, UART, Timers, PWM modules, built-in high-voltage LDO, which are suitable for FOC or square-wave based BLDC/PMSM motors.

Package type of the chip: QFN56

1.4 Functional Block Diagram

1.4.1 FU6361Q

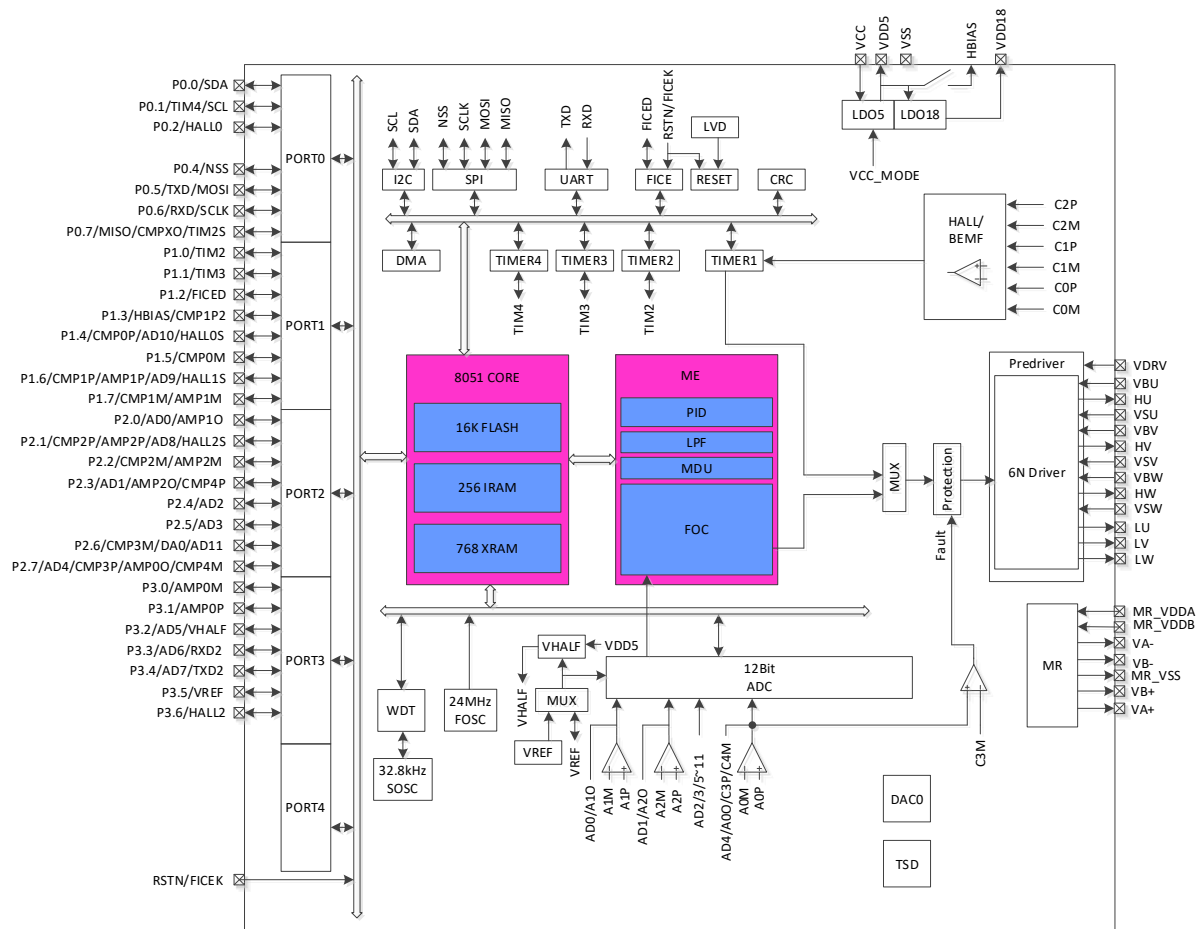


Figure 1-1 FU6361Q Functional Block Diagram

1.5 Memory Organization

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

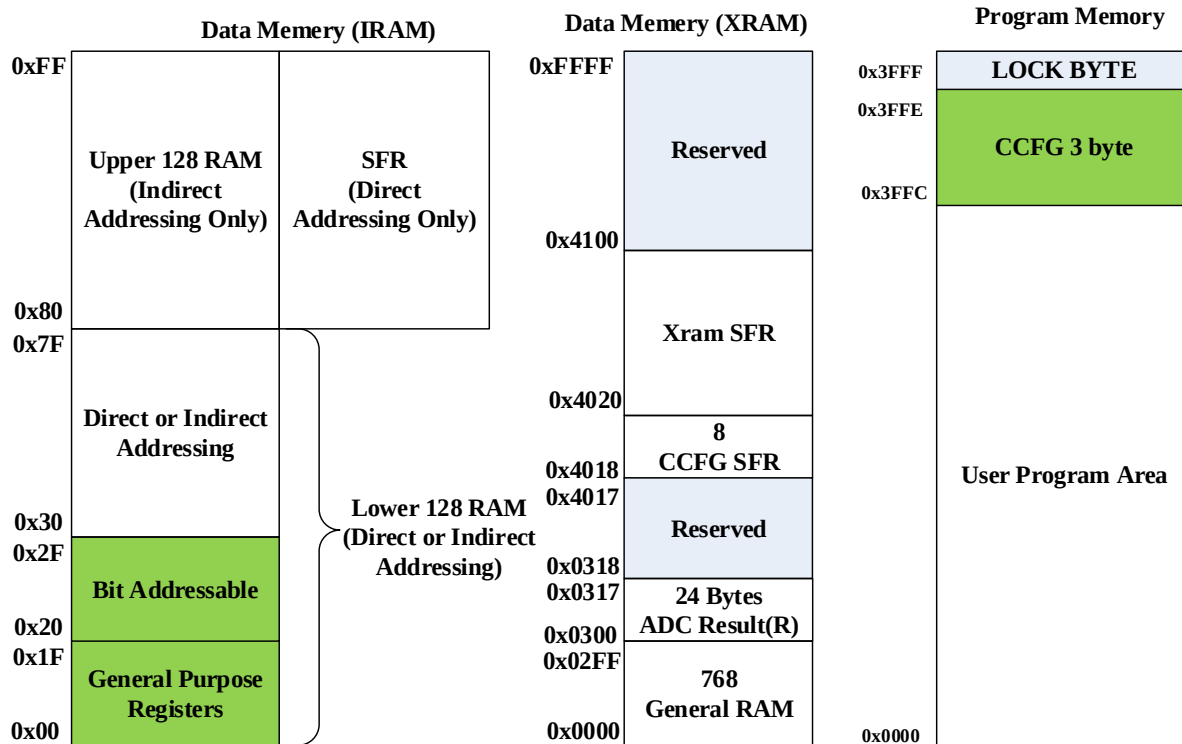


Figure 1-2 Memory Organization

1.5.1 Program Memory

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

1.5.2 Data Memory

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

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

The Internal Data Memory is shown in Figure 1-2. Locations 0x00~0x1F are addressable as 4 banks of general-purpose registers, each bank consisting of 8 registers. Locations 0x20~0x2F are 16-bit addressable, and locations 0x30~0x7F support direct and indirect addressing. When locations 0x80~0xFF are accessed by indirect addressing, it points to RAM. When locations 0x80~0xFF are accessed by direct addressing, it points to SFRs. Stack space is located in the Internal Data Memory.

1.5.3 SFR

Table 1-1 SFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xF8	DRV_OUT	PI_LPF_CR			P0_OE	P1_OE	P2_OE	P3_OE
0xF0	B		PI_KIL	PI_KIH	PI_UKMAXL	PI_UKMAXH	PI_UKMINL	PI_UKMINH
0xE8	P4	P4_OE	PI_EKL	PI_EKH	PI_UKL	PI_UKH	PI_KPL	PI_KPH
0xE0	ACC						LPF_YL	LPF_YH
0xD8	IP3	EVT_FILT	CMP_CR2	LVSR	CMP_CR3	LPF_K	LPF_XL	LPF_XH
0xD0	PSW	P1_IE	P1_IF	P2_IE	P2_IF	CMP_CR0	CMP_CR1	CMP_SR
0xC8	IP2	RST_SR	MDU_MBL	MDU_MBH	MDU_DB0	MDU_DB1		
0xC0	IP1	MDU_CR	MDU_MAL	MDU_MAH	MDU_DA0	MDU_DA1	MDU_DA2	MDU_DA3
0xB8	IP0							
0xB0	P3							
0xA8	IE	TIM2_CR1	TIM2_CNTRL	TIM2_CNTRH	TIM2_DRL	TIM2_DRH	TIM2_ARRL	TIM2_ARRH
0xA0	P2	TIM2_CR0	TIM3_CNTRL	TIM3_CNTRH	TIM3_DRL	TIM3_DRH	TIM3_ARRL	TIM3_ARRH
0x98	UT_CR	UT_DR	UT_BAUDL	UT_BAUDH	TIM3_CR0	TIM3_CR1	TIM4_CR0	TIM4_CR1
0x90	P1		TIM4_CNTRL	TIM4_CNTRH	TIM4_DRL	TIM4_DRH	TIM4_ARRL	TIM4_ARRH
0x88	TCON							
0x80	P0	SP	DPL	DPH	FLA_KEY	FLA_CR		PCON

Notes:

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

1.5.4 XSFR

Table 1-2 XSFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x40f0	EXT0	TIM234_CTRL	CMP_AMP	TSD_ADJ				
0x40e8	FOC_ID_LPFK	FOC_IQ_LPFK	FOC_KFGH	FOC_KFGL			FOC_CR3	
0x40e0	FOC__EMFH	FOC__EMFL	FOC_UDCPSH	FOC_UDCPSL	FOC_UQCPSH	FOC_UQCPSL	FOC_UQEXH	FOC_UQEXL
0x40d8	FOC__POWH	FOC__POWL	FOC_IAMAXH	FOC_IAMAXL	FOC_IBMAXH	FOC_IBMAXL	FOC_ICMAXH	FOC_ICMAXL
0x40d0	FOC__EALPH	FOC__EALPL	FOC_EBETH	FOC_EBETL	FOC_EOMEH	FOC_EOMEL	FOC_ESQUH	FOC_ESQUL
0x40c8	FOC__IBH	FOC__IBL	FOC__IAH	FOC__IAL	FOC__THETAH	FOC__THETAL	FOC__ETHETAH	FOC__ETHETAL
0x40c0	FOC__IBETH	FOC__IBETL	FOC__VBETH	FOC__VBETL	FOC__VALPH	FOC__VALPL	FOC__ICH	FOC__ICL
0x40b8	FOC__UDH	FOC__UDL	FOC__UQH	FOC__UQL	FOC__IDH	FOC__IDL	FOC__IQH	FOC__IQL
0x40b0	FOC_DMAXH	FOC_DMAXL	FOC_DMINH	FOC_DMINL	FOC_QMAXH	FOC_QMAXL	FOC_QMINH	FOC_QMINL
0x40a8	FOC_RTHESTEPH	FOC_RTHESTEPL	FOC_RTHEACCH	FOC_RTHEACCL	FOC_RTHECNT	FOC_THECOR/ CMP SAMR	FOC_THECOMPH	FOC_THECOMPL
0x40a0	FOC_CR1	FOC_CR2	FOC_TSMIN	FOC_TGLI	FOC_TBLO	FOC_TRGDLY	FOC_CSOH	FOC_CSOL
0x4098	FOC_UDCFLTH/ TIM1_ITRIPH	FOC_UDCFLTL/ TIM1_ITRIPL						
0x4090	FOC_IDREFH	FOC_IDREFL	FOC_IQREFH	FOC_IQREFL	FOC_DQKPH	FOC_DQKPL	FOC_DQKIH	FOC_DQKIL
0x4088	FOC_EK3H/ TIM1_RARRH	FOC_EK3L/ TIM1_RARRL	FOC_EK4H/ TIM1_RCNRH	FOC_EK4L/ TIM1_RCNTRL	FOC_EK1H	FOC_EK1L	FOC_EK2H	FOC_EK2L
0x4080	FOC_FBASEH/ TIM1_DBR7H	FOC_FBASEL/ TIM1_DBR7L	FOC_EFREQACCH/ TIM1_BCNRH	FOC_EFREQACCL/ TIM1_BCNTRL	FOC_EFREQMINH/ TIM1_BCCRH	FOC_EFREQMINL/ TIM1_BCCRL	FOC_EFREQHOLDH/ TIM1_BARRH	FOC_EFREQHOLDL/ TIM1_BARRL
0x4078	FOC_KSLIDEH/ TIM1_DBR3H	FOC_KSLIDEL/ TIM1_DBR3L	FOC_EKLPMFMINH/ TIM1_DBR4H	FOC_EKLPMFMINL/ TIM1_DBR4L	FOC_EBMFKH/ TIM1_DBR5H	FOC_EBMFKL/ TIM1_DBR5L	FOC_OMEKLPFH/ TIM1_DBR6H	FOC_OMEKLPLF/ TIM1_DBR6L
0x4070	TIM1_BCORH	TIM1_BCORL			FOC_EKPH/ TIM1_DBR1H	FOC_EKPL/ TIM1_DBR1L	FOC_EKIH/ TIM1_DBR2H	FOC_EKIL/ TIM1_DBR2L
0x4068	TIM1_CR0	TIM1_CR1	TIM1_CR2	TIM1_CR3	TIM1_CR4	TIM1_IER	TIM1_SR	
0x4060	DRV_DTR	DRV_SR	DRV_CR		SYST_ARRH	SYST_ARRL		
0x4058	DRV_DRH	DRV_DRL	DRV_COMRH	DRV_COMRL	DRV_CMRH	DRV_CMRL	DRV_ARRH	DRV_ARRL
0x4050	P1_AN	P2_AN	P3_AN	P0_PU	P1_PU	P2_PU	P3_PU	P4_PU
0x4048				DAC_DR	PH_SEL		AMP_CR	VREF_VHALF_CR
0x4040	CAL_CR0	CAL_CR1						
0x4038	ADC_SCYC	ADC_CR	DMA0_CR0	DMA1_CR0	DMA0_CR1H	DMA0_CR1L	DMA1_CR1H	DMA1_CR1L

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x4030	SPI_CR0	SPI_CR1	SPI_CLK	SPI_DR		DAC_CR	ADC_MASK_SYSCH	ADC_MASK_SYSCL
0x4028	I2C_CR	I2C_ID	I2C_DR	I2C_SR	RTC_TMH	RTC_TML	RTC_STA	
0x4020		CRC_DIN	CRC_CR	CRC_DR	CRC_BEG	CRC_CNT	WDT_CR	WDT_REL
0x4018	CCFG7	CCFG6	CCFG5	CCFG4	CCFG3	CCFG2	CCFG1	CCFG0
0x0310	AD8_DRH	AD8_DRL	AD9_DRH	AD9_DRL	AD10_DRH	AD10_DRL	AD11_DRH	AD11_DRL
0x0308	AD4_DRH	AD4_DRL	AD5_DRH	AD5_DRL	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
0x03f8	LPF0_K		LPF0_X		LPF0_YH		LPF0_YL	
0x03f0	LPF1_K		LPF1_X		LPF1_YH		LPF1_YL	
0x03e8	PI0_UKH		PI0_UKL		PI0_UKMAX		PI0_UKMIN	
0x03e0	PI0_KP		PI0_EK1		PI0_EK		PI0_KI	
0x03d8	PI1_UKH		PI1_UKL		PI1_UKMAX		PI1_UKMIN	
0x03d0	PI1_KP		PI1_EK1		PI1_EK		PI1_KI	
0x03C8	MUL0_MA		MUL0_MB		MUL0_MCH		MUL0_MCL	
0x03C0	MUL1_MA		MUL1_MB		MUL1_MCH		MUL1_MCL	
0x03B8	DIV0_DB		DIV0_DQH		DIV0_DQL		DIV0_DR	
0x03B0	DIV1_DQL		DIV1_DR		DIV0_DAH		DIV0_DAL	
0x03A8	DIV1_DAH		DIV1_DAL		DIV1_DB		DIV1_DQH	
0x03A0	LPF2_K		LPF2_X		LPF2_YH		LPF2_YL	
0x0398	LPF3_K		LPF3_X		LPF3_YH		LPF3_YL	
0x0390	PI2_UKH		PI2_UKL		PI2_UKMAX		PI2_UKMIN	
0x0388	PI2_KP		PI2_EK1		PI2_EK		PI2_KI	
0x0380	PI3_UKH		PI3_UKL		PI3_UKMAX		PI3_UKMIN	
0x0378	PI3_KP		PI3_EK1		PI3_EK		PI3_KI	
0x0370	MUL2_MA		MUL2_MB		MUL2_MCH		MUL2_MCL	
0x0368	MUL3_MA		MUL3_MB		MUL3_MCH		MUL3_MCL	
0x0360	DIV2_DB		DIV2_DQH		DIV2_DQL		DIV2_DR	
0x0358	DIV3_DQL		DIV3_DR		DIV2_DAH		DIV2_DAL	

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

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

2 Pin Definitions

The IO types are defined as follows:

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

2.1 FU6361Q QFN56 Pins

Table 2-1 FU6361Q QFN56 Pin Descriptions

Pin	FU6361Q QFN56	IO Type	Description
VSU	1	P	6N pre-driver phase-U input, as GND reference for phase-U high-side bootstrap
HU	2	DO	6N pre-driver high-side phase-U PWM output
VBU	3	P	6N pre-driver high-side phase-U bootstrap power supply
VSV	4	P	6N pre-driver phase-V input, as GND reference for phase-V high-side bootstrap
HV	5	DO	6N pre-driver high-side phase-U PWM output
VBV	6	P	6N pre-driver high-side phase-V bootstrap power supply
VSW	7	P	6N pre-driver phase-W input, as GND reference for phase-W high-side bootstrap
HW	8	DO	6N pre-driver high-side phase-W PWM output
VBW	9	P	6N pre-driver high-side phase-W bootstrap power supply
MR_VDDA	10	P	Power supply of angle sensor
MR_Vddb	11	P	Power supply of angle sensor
VA-	12	AO	Negative differential sine output of angle sensor
VB-	13	AO	Negative differential cosine output of angle sensor
MR_VSS	14	P	Ground for angle sensor
VB+	15	AO	Positive differential cosine output of angle sensor
VA+	16	AO	Positive differential sine output of angle sensor
VCC	17	P	Power input. The voltage range is determined by VCC_MODE, with an external filter capacitor of 10μF or above.
VSS	18	P	Ground
VDD5	19	P	Mid-voltage power input or 5V LDO power output is determined by VCC_MODE. See descriptions on VCC pin for power connection. It is connected with a 1μF~4.7μF external capacitor.
RSTN/ FICEK	20	DI/ DI	Input of external reset; Built-in pull-up resistor; Schmit input FICE SCL
VDD18	21	P	1.8V LDO output with a 1μF ~ 4.7μF external capacitor
P1.2/ FICED	22	DB/ DB	GPIO, configurable as INT1 input FICE SDA

Pin	FU6361Q QFN56	IO Type	Description
P1.3/ HBIAS/ CMP1P2	23	DB/ DO/ AI	GPIO Hall bias power supply, internally connected to VDD5 via a switch CMP1 positive input
P1.4/ CMP0P/ AD10/ HALL0S	24	DB/ AI/ AI/ DI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP0 positive input Input of ADC channel 10 Hall0 logic level input after function switching
P1.5/ CMP0M	25	DB/ AI/	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP0 negative input
P1.6/ CMP1P/ AMP1P/ AD9/ HALL1S	26	DB/ AI/ AI/ AI/ DI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP1 positive input AMP1 positive input for voltage signals generated by phase current 1 Input of ADC channel 9 Hall1 logic level input after function switching
P1.7/ CMP1M/ AMP1M	27	DB/ AI/ AI	GPIO, configurable with a 4.7k pull-up resistor CMP1 negative input AMP1 negative input
P2.0/ AD0/ AMP1O	28	DB/ AI/ AO	GPIO, configurable as INT1 input Input of ADC channel 0 for collecting amplified signals from phase current 1 AMP1 output
P2.1/ CMP2P/ AMP2P/ AD8/ HALL2S	29	DB/ AI/ AI/ AI/ DI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP2 positive input AMP2 positive input for voltage signals generated by phase current 2 Input of ADC channel 8 Hall2 logic level input after function switching
P2.2/ CMP2M/ AMP2M	30	DB/ AI/ AI	GPIO, configurable as ITN1 input with a 4.7k pull-up resistor CMP2 negative input AMP2 negative input
P2.3/ AD1/ AMP2O/ CMP4P	31	DB/ AI/ AO/ AI	GPIO, configurable as INT1 input Input of ADC channel 1 for collecting amplified signals from phase current 2 AMP2 output CMP4 positive input for bus current sampling to detect overcurrent
P2.4/ AD2	32	DB/ AI	GPIO, configurable as INT1 input Input of ADC channel 2 or bus voltage signal
P2.5/ AD3	33	DB/ AI	GPIO, configurable as INT1 input Input of ADC channel 3
P2.6/ CMP3M/ DA0/ AD11	34	DB/ AI/ AO/ AI	GPIO, configurable as INT1 input Over-current reference signal input; CMP3 negative input Internal DAC voltage, without Buffer output Input of ADC channel 11
P2.7/ AD4/ CMP3P/ AMP0O/ CMP4M	35	DB/ AI/ AI/ AO/ AI	GPIO, configurable as INT1 input Input of ADC channel 4 for collecting the amplified bus current signal CMP3 positive input for bus current sampling to detect overcurrent AMP0 output, the voltage output after the bus current is amplified Over-current reference signal input; CMP4 negative input
P3.0/ AMP0M	36	DB/ AI	GPIO AMP0 negative input for amplifying the bus current signal
P3.1/ AMP0P	37	DB/ AI	GPIO AMP0 positive input for amplifying the bus current signal
P3.2/ AD5/ VHALF	38	DB/ AI/ AO	GPIO Input of over-temperature signal or ADC channel 5 1/2 VDD5 or 1/2 VREF output with an external 1μF capacitor
P3.3/ AD6/ RXD2	39	DB/ AI/ DO	GPIO Input of ADC channel 6 UART2 RXD after function switching

Pin	FU6361Q QFN56	IO Type	Description
P3.4/ AD7/ TXD2	40	DB/ AI/ DO	GPIO Analog voltage input for speed control or input of ADC channel7 UART2 TXD after function switching
P3.5/ VREF	41	DB/ AB	GPIO ADC external VREF input or internal VREF output, with a 1 μ F~4.7 μ F external capacitor
P3.6/ HALL2	42	DB/ DI	GPIO Hall2 logic level input
P0.0/ SDA	43	DB/ DB	GPIO, configurable as INT0 input I ² C SDA, configured as collector open-drain output with a 4.7k pull-up resistor
P0.1/ TIM4/ SCL	44	DB/ DB/ DB	GPIO Timer4 input capture mode I ² C SCL, configured as collector open-drain output with a 4.7k pull-up resistor
P0.2/ HALL0	45	DB/ AI/ DO	GPIO Hall0 logic level input
P0.4/ NSS	46	DB/ DB	GPIO SPI NSS
P0.5/ TXD/ MOSI	47	DB/ DO/ DB	GPIO UART1 TXD before function switching SPI MOSI, master output or slave input
P0.6/ RXD/ MISO	48	DB/ DI/ DB	GPIO UART1 RXD before function switching SPI MISO, master input or slave output
P0.7/ SCLK/ CMPXO/ TIM2S	49	DB/ DB/ DO/ DB	GPIO SPI SCLK Test pin for comparator output Timer2 input capture mode or PWM output after function switching
P1.0/ TIM2	50	DB/ DB	GPIO, configurable as INT1 input Timer2 input capture mode or PWM output before function switching
P1.1/ TIM3	51	DB/ DB	GPIO, configurable as INT1 input Timer3 input capture mode
VDRV	52	P	6N pre-driver power supply, with a 1 μ F~10 μ F external capacitor
COM	53	P	DRV ground
LU	54	DO	6N pre-driver low-side phase-U PWM output
LV	55	DO	6N pre-driver low-side phase-V PWM output
LW	56	DO	6N pre-driver low-side phase-W PWM output

2.2 FU6361Q QFN56 Pinout Diagram

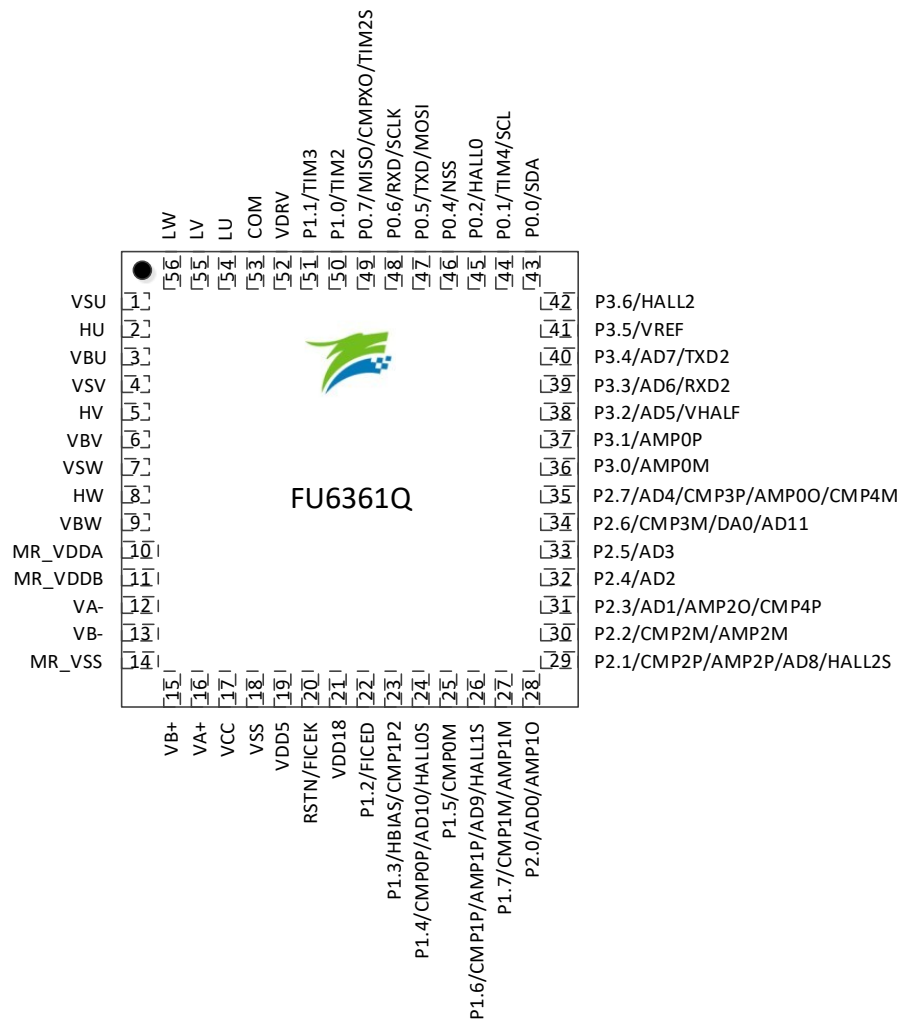


Figure 2-1 FU6361Q QFN56 Pinout Diagram

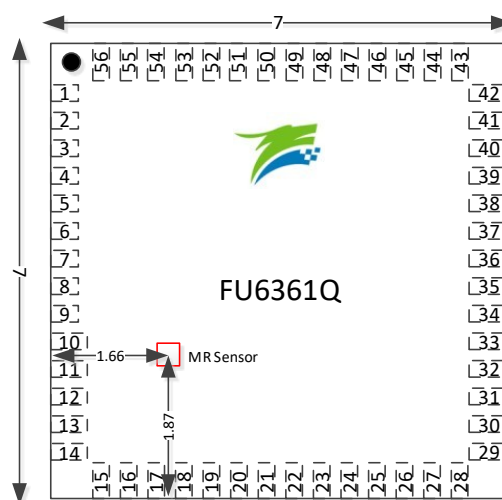
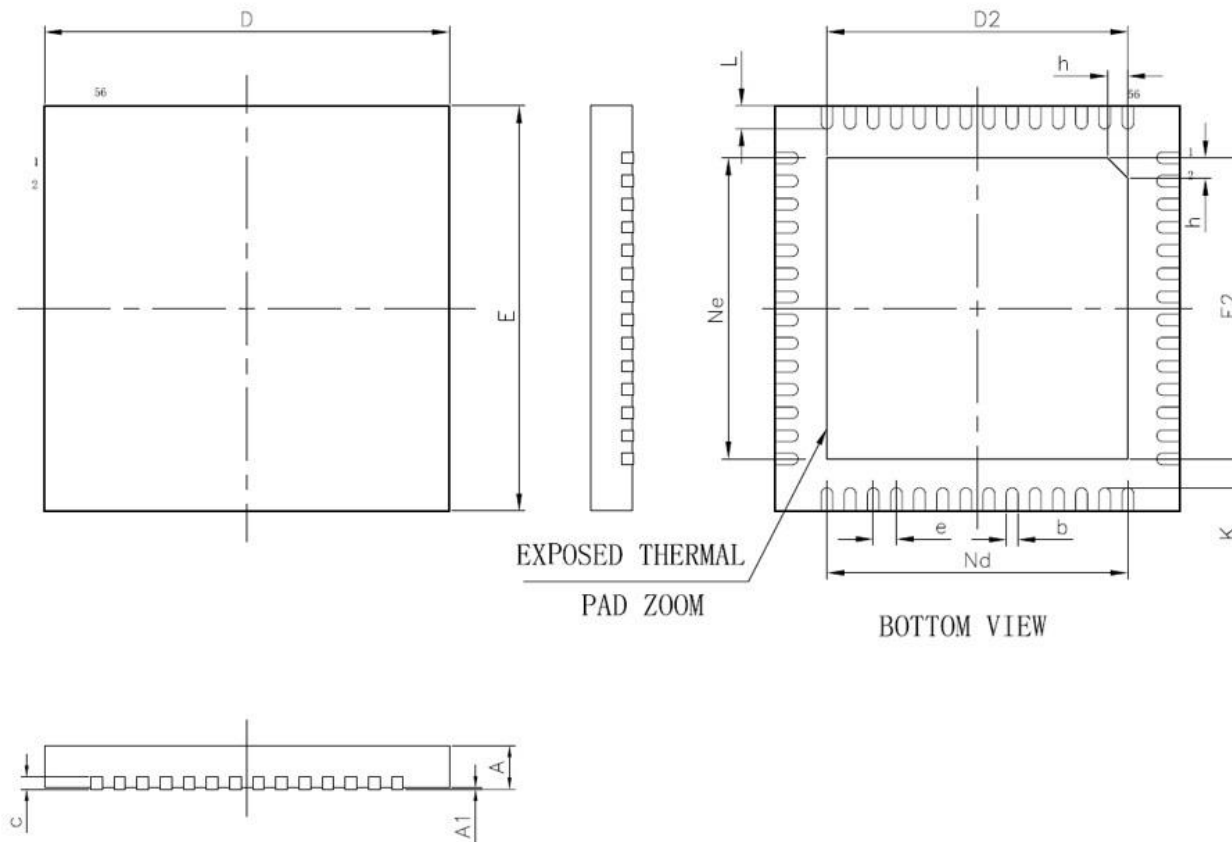


Figure 2-2 Location of the Sensor (unit: mm)

3 Package Information

3.1 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
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
Lead Frame (L/F) (mil)	217*217		

Figure 3-1 QFN56_7X7 Package Drawings and Dimensions

4 Ordering Information

Table 4-1 Model Selections

Model	MIPS (Peak)	FLASH (kByte)	XRAM (kByte)	Clock Circuit		Driver Interface		Driver Type		I²C/UART/SPI	MR Angle Sensor	DMA	GPIO	Timer	Analog Peripherals								Lead-free	Package
				Internal Fast Clock	Internal Slow Clock	6N Pre-driver	PWM	Square Wave	FOC						ADC			DAC		VREF	Operational Amplifier	Comparator		
															Number	Channel	Bits	Number	Bits					
FU6361Q	24	16	0.75	√	√	√	√	√	√	√	√	30	6	1	12	12	1	9	√	3	3	√	QFN56 (7x7 mm)	

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

Table 5-1 Absolute Maximum Ratings

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Operating Ambient Temperature T_A		-40	-	85	°C
Operating Ambient Temperature T_A	$V_{CC} \leq 12V$, $I_{VCC} \leq 30mA$	-40	-	105	°C
Operating Junction Temperature T_J		-40	-	150	°C
Storage Temperature		-55	-	150	°C
VCC to VSS Voltage	Depend on voltage applications.	-0.3	-	36	V
VDD5 to VSS Voltage		-0.3	5	6.5	V
VDRV to VSS Voltage		-0.3	-	22	V
Low-side Output Voltage $V_{LU,LV,LW}$		-0.3	-	$V_{DRV} + 0.3$	V
High-side Floating Voltage $V_{BU,BV,BW}$		-0.3	-	165	V
High-side Floating Offset Voltage $V_{SU,SV,SW}$		$V_{BU,BV,BW} - 22$	-	$V_{BU,BV,BW} + 0.3$	V
High-side Output Voltage $V_{HU,HV,HW}$		$V_{SU,SV,SW} - 0.3$	-	$V_{SU,SV,SW} + 0.3$	V
VDD18		-0.3	1.85	2	V
RSTN/VCC_MODE/GPIO to VSS Voltage		-0.3	-	$V_{DD5} + 0.3$	V

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

5.2 MR Sensor Electrical Characteristics

Table 5-2 MR Sensor Electrical Characteristics

($T_A = 25^\circ\text{C}$ unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Power Supply VDDA/VDDDB		1.0	5.0	25.0	V
Resistance (R)	Bridge Current = 1mA The two bridges are connected in parallel.	800	1000	1200	Ω
Angle Range	Larger than saturation field	0	-	180	Deg
Sensitivity	VDDA=5V, Magnetic Field = 80G Near the zero output	-	2.1	-	mV/°
Output Voltage Amplitude Vamp	VDDA=5V, Magnetic Field = 80G Peak-to-peak Value	100	120	140	mV
Zero Offset of Bridge	Magnetic Field =80G	-3	0	+3	mV/V

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Temperature Coefficient of Bridge Resistance (TCR)	VDDA = 5V T _A = -40 ~ +85°C	-	2800	-	ppm/°C
Temperature Coefficient of Sensitivity (TCS)	VDDA = 5V T _A = -40 ~ +85°C	-	-3200	-	ppm/°C
Temperature Coefficient of Bridge Offset (TCO)	T _A = -40 ~ +85°C	-	±100	-	ppm/°C
Matching Rate of Bridge	(Vamp-A/Vamp-B)*100	97	100	103	%

5.3 Global Electrical Characteristics

Table 5-3 Global Electrical Characteristics

(T_A = 25°C, VCC_MODE = 0 and VCC = VDRV=15V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VCC Operating Voltage	High-voltage single-power supply mode: VCC_MODE=0	5	-	24	V
	High-voltage dual-power supply mode: VCC_MODE = 1, VCC ≥ VDD5 ^[1]	5	-	36	V
VDD5 Operating Voltage	VCC_MODE = 1 and VCC pin is connected with VDD5 pin ^[1]	3	-	5.5	V
V _{BU} to V _{SU} Voltage V _{BV} to V _{SV} Voltage V _{BW} to V _{SW} Voltage		-	-	20	V
I _{VCC} Operating Current ^[2]		-	24	-	mA
I _{VCC} Standby Current ^[2]		-	6	-	mA
I _{VCC} Sleep-mode Current ^[3]	VCC_MODE = 0	-	350	650	μA
	VCC_MODE = 1, VCC = VDD5 = 5V	-	300	500	μA

Notes:

[1] Characteristics may vary with different configurations.

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

[3] Unless otherwise specified, VCC_MODE = 0 means VCC_MODE = GND, and VCC_MODE = 1 means VCC_MODE = VDD5.

5.4 GPIO Electrical Characteristics

Table 5-4 GPIO Electrical Characteristics

(T_A = 25°C, VCC_MODE = 0 and VCC = 15V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Output Rise Time	50pF load, from 10% to 90%, T _A = 25°C	-	15	-	ns
Output Fall Time	50pF load, from 90% to 10%, T _A = 25°C	-	13	-	ns
V _{OH} High-level Output Voltage	I _{OH} = 4mA, VDD5 = 5V	VDD5-0.7	-	-	V
V _{OL} Low-level Output Voltage	I _{OL} = 4mA, VDD5 = 5V	-	-	0.7	V

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _{IH} High-level Input Voltage ^[1]		0.7*VDD5	-	-	V
V _{IL} Low-level Input Voltage	VDD5 = 5V	-	-	0.2*VDD5	V
Pull-up Resistor ^[2]	V _{in} = 0V	-	33	-	kΩ
Pull-up Resistor ^[3]	V _{in} = 0V	-	5.6	-	kΩ

Notes:

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

[2] GPIOs except P0[2:0], P1[6:3], P2[1] and P3[7:6].

[3] P0[2:0], P1[6:3], P2[1] and P3[7:6].

5.5 Pre-driver IO Electrical Characteristics

Table 5-5 Pre-driver IO Electrical Characteristics

(T_A = 25°C, VCC_MODE = 0 and VCC = VDRV=15V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
High-level Peak Output Current		-	0.8	-	A
Low-level Peak Output Current		-	0.8	-	A
VDRV Operating Voltage		6	-	20	V
High-side Floating Voltage V _{BU,BV,BW}		-	-	140	V
High-side Floating Offset Voltage V _{SU,SV,SW}		V _{BU,BV,BW} - 20	-	V _{BU,BV,BW} - 6	V
VDRV UVLO Threshold Voltage		4.3	4.8	5.3	V
VDRV UVLO Release Voltage		4	4.5	5	V
VDRV UVLO Hysteresis Voltage		0.2	0.3	-	V
Output Rise Time	1nF load, from 10% to 90%	-	30	70	ns
Output Fall Time	1nF load, from 90% to 10%	-	30	70	ns
Deadtime	DT	-	100	-	ns

5.6 ADC Electrical Characteristics

Table 5-6 ADC Electrical Characteristics

(T_A = 25°C and VCC = 5V~24V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
INL (Integral Nonlinearity)	12-bit	-	2	-	LSB
DNL (Differential Nonlinearity)	12-bit	-	1.5	-	LSB
OFFSET (Offset Error)	12-bit	-	10	-	LSB
SNR (Signal-to-noise Ratio)	f _{IN} = 350kHz	-	70.8	-	dB
ENOB (Effective Number of Bits)	f _{IN} = 350kHz	-	10.5	-	Bit
SFDR (Spurious-free Dynamic Range)	f _{IN} = 350kHz	-	68.2	-	dB

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
THD (Total Harmonic Distortion)	$f_{IN} = 350\text{kHz}$	-	67	-	dB
R_{IN} Input Resistance		-	500	-	Ω
C_{IN} Input Capacitance		-	30	-	pF
Conversion Time		-	0.6	-	μs
Sampling Time		3	-	63	ADCLK ^[1]

Note:

[1] ADCLK = 12MHz

5.7 VREF and VHALF Electrical Characteristics

Table 5-7 VREF and VHALF Electrical Characteristics

($T_A = -40^\circ\text{C} \sim 85^\circ\text{C}$, VCC_MODE = 0 and VCC = 15V)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VREF	VREF_VHALF_CR[VREFVSEL] = 00	-	4.5	-	V
	VREF_VHALF_CR[VREFVSEL] = 01	-	VDD5	-	V
	VREF_VHALF_CR[VREFVSEL] = 11	-	4	-	V
	VREF_VHALF_CR[VREFVSEL] = 10	-	3	-	V
VHALF		-	VREF/2	-	V

5.8 Operational Amplifier Electrical Characteristics

Table 5-8 Operational Amplifier Electrical Characteristics

($T_A = 25^\circ\text{C}$, VCC_MODE = 0 and VCC = 5V~24V)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{ICMR} Common-mode Input Voltage Range		0	-	VDD5-1.5	V
V_{OS} Operational Amplifier Offset Voltage	$T_A = 25^\circ\text{C}$	-	5	10	mV
A_{OL} Open-loop Gain	$R_L = 100\text{k}\Omega$	-	80	-	dB
Unity-gain Bandwidth (UGBW)	$C_L = 40\text{pF}$	6	10	-	MHz
Slew Rate (SR)	$C_L = 40\text{pF}$	10	15	-	V/ μs
Operational Amplifier Gain	2x	1.88	2	2.12	-
	4x	3.76	4	4.24	-
	8x	7.5	8	8.5	-
	16x	15	16	17	-

5.9 BEMF Electrical Characteristics

Table 5-9 BEMF Electrical Characteristics

($T_A = 25^{\circ}\text{C}$, $VCC_MODE = 0$ and $VCC = 5V \sim 24V$)

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

5.10 OSC Electrical Characteristics

Table 5-10 OSC Electrical Characteristics

($T_A = -40^{\circ}\text{C} \sim 85^{\circ}\text{C}$, $VCC_MODE = 0$ and $VCC = 5V \sim 24V$)

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

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

5.11 Reset Electrical Characteristics

Table 5-11 Reset Electrical Characteristics

($T_A = -40^{\circ}\text{C} \sim 85^{\circ}\text{C}$, $VCC_MODE = 0$ and $VCC = 5V \sim 24V$)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Minimum Time to Reset to Low Level		50	-	-	μs

5.12 LDO Electrical Characteristics

Table 5-12 LDO Electrical Characteristics

($T_A = -40^{\circ}\text{C} \sim 85^{\circ}\text{C}$, $VCC_MODE = 0$ and $VCC = 5V \sim 24V$)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD5 Voltage	$VCC = 7V \sim 24V$ and $VCC_MODE=0$	4.7	5	5.3	V

5.13 Package Thermal Resistance

Table 5-13 QFN56 Package Thermal Resistance

Parameter	Test Conditions	Value	Unit
Junction-to-ambient Temperature Thermal Resistance $\theta_{JA}^{[1]}$	JEDEC standard, 2S2P PCB	33	$^{\circ}\text{C/W}$
	JEDEC standard, 1S0P PCB	55	$^{\circ}\text{C/W}$
Junction-to-case Temperature Thermal Resistance $\theta_{JC}^{[1]}$	JEDEC standard, 2S2P PCB	9.2	$^{\circ}\text{C/W}$

Note:

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

6 Reset Control

6.1 Reset Source (RST_SRC)

The chip includes a reset circuitry with 6 reset sources:

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

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

6.2 Reset Enable

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

6.3 External Reset and Power-on Reset

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

6.4 Low Voltage Detector reset

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

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

6.5 Watchdog Timer Reset

After the watchdog timer (WDT) is enabled, the system is reset if WDT is not initialized before it get overflowed. In this case, when WDT reaches its maximum value, it generates an output pulse to reset the chip, which ensures the software runs normally.

6.6 Flash Error Detector Reset

The Flash memory can be programmed by the software using MOVX instruction for read/write/erase operations (See section Code Protection). A Flash error detector reset (RSTFED) occurs if a Flash erase is attempted targeting the last sector (0x3F80~0x3FFF). RSTFED is always enabled and cannot be disabled.

6.7 Reset Registers

Bit	7	6	5	4	3	2	1	0
Name	RSTPOR	RSTEXT	RSTLVD	RSV	RSTWDT	RSTFED	RSTDBG	RSTCLR
Type	R	R	R	-	R	R	R	W1
Reset	-	-	-	-	-	-	-	0
Bit	Name	Description						
[7]	RSTPOW	Power-On Reset Flag 0: Last reset was not a power-on reset. 1: Last reset was a power-on reset.						
[6]	RSTEXT	External Reset Flag 0: Last reset was not an external reset. 1: Last reset was an external reset.						
[5]	RSTLVD	Low Voltage Detection (LVD) Reset Flag 0: Last reset was not an LVD reset. 1: Last reset was an LVD reset.						
[4]	RSV	Reserved						
[3]	RSTWDT	WDT Reset Flag 0: Last reset was not a WDT reset. 1: Last reset was a WDT reset.						
[2]	RSTFED	Flash Error Detector Reset Flag 0: Last reset was not a Flash error detector reset. 1: Last reset was a Flash error detector reset.						
[1]	RSTDBG	Debug Reset Flag 0: Last reset was not a debug reset. 1: Last reset was a debug reset.						
[0]	RSTCLR	Clear Analog Reset Flags Register The reset flags [7:3] are cleared after “1” is written to this bit. This bit has no effect when it is read.						

7 Interrupt Control

7.1 Interrupt Registers

7.1.1 IE (0xA8)

Bit	7	6	5	4	3	2	1	0
Name	EA	RTCIE	RSV	ES0	SPIIE	EX1	TSDIE	EX0
Type	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Reset	0	0	-	0	0	0	0	0
Bit	Name	Description						
[7]	EA	Enable All Interrupts 0: Disable 1: Enable						
[6]	RTCIE	RTC Interrupt Enable 0: Disable 1: Enable						
[5]	RSV	Reserved						
[4]	ES0	UART Interrupt Enable 0: Disable 1: Enable						
[3]	SPIIE	SPI Interrupt Enable 0: Disable 1: Enable						
[2]	EX1	External Interrupt 1 (INT1) Enable 0: Disable 1: Enable						
[1]	TSDIE	TSD Interrupt Enable 0: Disable 1: Enable						
[0]	EX0	External Interrupt 0 (INT0) Enable 0: Disable 1: Enable						

7.1.2 IP0 (0xB8)

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

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

7.1.3 IP1 (0xC0)

Bit	7	6	5	4	3	2	1	0
Name	PCMP		PADC		PTIM1		PTIM2	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:6]	PCMP	Comparator Interrupt Priority Setting						
[5:4]	PADC	ADC Interrupt Priority Setting						
[3:2]	PTIM1	Timer1 Interrupt Priority Setting						
[1:0]	PTIM2	Timer2 Interrupt Priority Setting						

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

7.1.4 IP2 (0xC8)

Bit	7	6	5	4	3	2	1	0
Name	PTSD		PTIM4		PTIM3		PRTC	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:6]	PTSD	TSD Interrupt Priority Setting						
[5:4]	PTIM4	Timer4 and SysTick Interrupt Priority Setting						
[3:2]	PTIM3	Timer3 Interrupt Priority Setting						
[1:0]	PRTC	RTC Interrupt Priority Setting						

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

7.1.5 IP3 (0xD8)

Bit	7	6	5	4	3	2	1	0
Name	PDMA		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	Description						
[7:6]	PDMA	DMA Interrupt Priority Setting						
[5:4]	PSPI	SPI Interrupt Priority Setting						
[3:2]	PI2C	I ² C Interrupt Priority Setting						
[1:0]	PUART	UART Interrupt Priority Setting						

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

7.1.6 TCON (0x88)

Bit	7	6	5	4	3	2	1	0
Name	RSV		TSDIF	IT1		IF0	IT0	
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0
Bit	Name	Description						
[7:6]	RSV	Reserved						
[5]	TSDIF	TSD Interrupt Flag This flag is often used with the overtemperature status bit LVSR[TSDIF]. 0: No over-temperature event occurs. 1: An over-temperature event occurs. A write of “0” in software clears this bit.						
[4:3]	IT1[1:0]	External Interrupt 1 (INT1) Edge Select 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on edge changes (rise or fall)						
[2]	IF0	External Interrupt 0 (INT0) Interrupt Flag 0: No INT0 event occurs. 1: INT0 event occurs. A write of “0” in software clears this bit.						
[1:0]	IT0[1:0]	External Interrupt 0 (INT0) Edge Select 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on edge changes (rise or fall)						

7.2 Interrupt Summary

Table 7-1 Interrupt Summary

Interrupt Source	Priority Order	Interrupt Vector	Interrupt Flag	Cleared by Software?	Enable Bit	Priority Control
Reset	Highest	0x0000	N/A	N/A	Always enabled	Highest
LVW Interrupt	0	0x0003	LVSR[0]	Y	CCFG1[6]	IP0[1:0]
INT0	1	0x000B	TCON[2]	Y	IE[0]	IP0[3:2]
INT1	2	0x0013	P1IF[7:0] P2IF[7:0]	Y	IE[2]	IP0[5:4]
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]
Comparator Interrupt	7	0x003B	CMP_SR[7]/ CMP_SR[6:4]	Y	CMP_CR0[7:0]	IP1[7:6]
RTC Interrupt	8	0x0043	RTC_STA[6]	Y	IE[6]	IP2[1:0]
TIM3 Interrupt	9	0x004B	TIM3_CR1[7:5]	Y	TIM3_CR1[4:3] TIM3_CR0[3]	IP2[3:2]
TIM4 Interrupt Systick Interrupt	10	0x0053	TIM4_CR1[7:5] DRV_SR[7]	Y	TIM4_CR1[4:3] TIM4_CR0[3] DRV_SR[6]	IP2[5:4]
TSD Interrupt	11	0x005B	TCON[5]	Y	IE[1]	IP2[7:6]
UART Interrupt	12	0x0063	UT_CR[1:0]	Y	IE[4]	IP3[1:0]
I ² C 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	DMA0_CR0[7] DMA1_CR0[7]	Y	DMA0_CR0[2]	IP3[7:6]

The chip includes an interrupt system with a total of 16 interrupt sources. Each interrupt source can be individually programmed in IP0 ~ IP3 registers with one of four priority levels. A low priority interrupt can be interrupted by a high priority interrupt. If two interrupt requests are generated at the same time, the interrupt with the higher priority is serviced first. See Table 7-1 for the interrupt sources and default priority orders, where the lower the mark the higher the priority level. An interrupt will not be preempted by another of the same priority level.

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

7.3 External Interrupt

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

The digital input signals from P0.0 ~ P0.6/P1.1 or output signals from CMP4 can be used to trigger an INT0 after IE[EX0] is set to “1”. The interrupt source is selected through LVSR[EXT0CFG] bit. These trigger sources share one interrupt entry point, one interrupt flag bit TCON[IF0] and one interrupt enable bit IE[EX0]. TCON[IT0] bit selects the interrupt edge.

The digital input signals from P1.0 ~ 1.7/P2.0 ~ 2.7 can be used to trigger an INT1 after IE[EX1] is set to “1” and the corresponding P1IE/P2IE is configured. P1_IF and P2_IF are interrupt flag bits, and P1_IE and P2_IE are interrupt enable bits. These 16 interrupt sources share one interrupt entry and one interrupt enable bit IE[EX1].

Table 7-2 IO Corresponding to INT1

SFR Addr	Bit	Name	Description	R/W	Reset
0xD1	[7:0]	P1IE[7:0]	Interrupt enable bit when Port1 serves as INT1.	R/W	0x00
0xD2	[7:0]	P1IF[7:0]	Interrupt flag bit when Port1 serves as INT1. Note: When “0” is written to MCU to clear an interrupt flag bit, other flag bits must be written with “1”. Otherwise, the irrelevant flag bits may be cleared. To clear P1IF[0], the statement <code>mov D2h,#0FEh</code> is recommended.	R/W	0x00
0xD3	[7:0]	P2IE[7:0]	Interrupt enable bit when Port2 serves as INT1.	R/W	0x00
0xD4	[7:0]	P2IF[7:0]	Interrupt flag bit when Port2 serves as INT1. Note: When “0” is written to MCU to clear an interrupt flag bit, other flag bits must be written with “1”. Otherwise, the irrelevant flag bits may be cleared. To clear P2IF[0], the statement <code>mov D4h,#0Feh</code> is recommended.	R/W	0x00

8 I²C

The I²C module provides an industry standard two-wire serial interface and is a simple bi-directional synchronous serial bus for communication between MCU and external I²C devices. The bus consists of two serial lines, SDA (serial data line) and SCL (serial clock line), with open-drain output, which works normally after the pull-up resistor is activated and the I/O pin turns its state to VDD5.

Features:

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

Both SDA and SCL lines are high level when the bus is idle, which is the only basis for detecting whether the bus is idle or not. Only one master device and at least one slave device are active on the bus during the transmission. When the bus is occupied, other devices must wait for the bus idle to start an I²C communication. The master starts the bus to transfer data. Clock signal is sent to all devices via SCL and the slave address and read/write mode are sent via SDA. When a device on the bus matches the address, it acts as a slave. The relationships between masters and slaves or data transfer direction on the bus are not constant. The process for the master to send data to the slave is shown in Figure 8-1. The master first addresses the slave device and waits for the slave response. And then, it sends data to the slave. Finally, the master terminates the data transmission. The process for the master to receive data from the slave is shown in Figure 8-2. The master first addresses the slave and waits for the slave response. And then, it receives the data from the slave. Finally, the master terminates the data transmission. In this case, the master generates the timing clock and stops the data transmission.

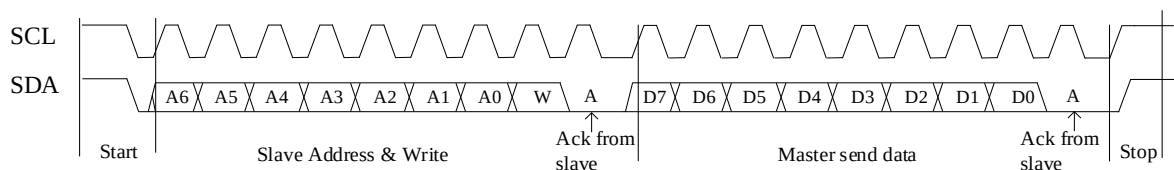


Figure 8-1 Master Sends Data to Slave

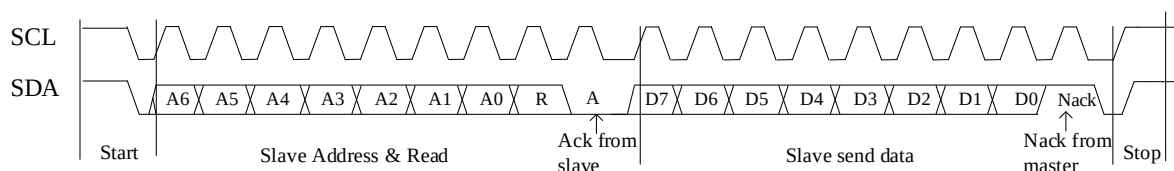


Figure 8-2 Master Receives Data from Slave

The I²C module supports master mode and slave mode with the fastest speed up to 1MHz. I²C feature can be used after I²C parameters and correct I²C address are configured. I²C communication is controlled by START signal (STA), read/write signal (DMOD), bus pending signal (STR), acknowledgement signal

(NACK) and STOP signal (STP).

8.1 I²C Operations

8.1.1 Master Mode

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

8.1.2 Slave Mode

1. Set I2C_CR[I2CMS] to “0” to select slave mode;
2. Configure I2C_ID[I2CADD] to set the slave address or set I2C_ID[GC] to “1” to enable general call mode;
3. Set I2C_CR[I2CEN] to “1” to enable I²C;
4. After START signal and the correct address are received, I2C_SR[I2CSTA] and I2C_SR[STR] are set to “1” by hardware and SCL is pulled LOW by the slave. ACK/NACK is configured via I2C_SR[NACK] and the slave determines whether to receive or send the data via I2C_SR[DMOD];
5. Sending Data: Write the data to I2C_DR register, and clear I2C_SR[STR] to release SCL. The data is sent after ACK/NACK is transmitted. After the data is sent and ACK/NACK is received from the master, I2C_SR[STR] is set to “1” by hardware and SCL is pulled LOW by the slave;
6. Receiving Data: Clear I2C_SR[STR] to release SCL to receive data. After the data is received, I2C_SR[STR] is set to “1” by hardware and SCL is pulled LOW by the slave. ACK/NACK is

configured via I2C_SR[NACK] to reset I2C_SR[STR] to release SCL for ACK/NACK transmission. If new data is received, I2C_SR[STR] is set to “1” by hardware and SCL is pulled LOW by the slave;

7. RESTART: If the slave is processing a service when receiving START signal, it stops the current routine and waits for receiving address.

8.1.3 I²C Interrupt Sources

The interrupt sources of I²C include:

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

An I²C interrupt request is generated when the interrupt enable bit I2C_CR[I2CIE] is set to “1”.

8.2 I²C Registers

8.2.1 I2C_CR (0x4028)

Bit	7	6	5	4	3	2	1	0
Name	I2CEN	I2CMS	RSV			I2CSPD		I2CIE
Type	R/W	R/W	-	-	-	R/W	R/W	R/W
Reset	0	0	-	-	-	0	0	0
Bit	Name	Description						
[7]	I2CEN	I ² C Enable Enable the associated GPIO and switch to I ² C mode, serving as collector open-drain output. The pull-up setting decides whether to pull I ² C HIGH. 0: Disable 1: Enable						
[6]	I2CMS	Master/Slave Mode Selection 0: Slave 1: Master						
[5:3]	RSV	Reserved						
[2:1]	I2CSPD	I ² C transfer rate setting, valid only in Master Mode 00: 100kHz 01: 400kHz 10: 1MHz 11: Reserved						
[0]	I2CIE	I ² C Interrupt Enable 0: Disable 1: Enable. An I ² C interrupt request is generated when the interrupt enable bit I2C_CR[I2CIE] is set to “1”.						

8.2.2 I2C_ID (0x4029)

Bit	7	6	5	4	3	2	1	0
Name	I2CADD							GC
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	1	0	1	0
Bit	Name	Description						
[7:1]	I2CADD	I ² C address						
[0]	GC	General call, valid only in Slave Mode 0: General call is disabled 1: General call is enabled, i.e., receiving device responses at address 0x00 as well.						

8.2.3 I2C_DR (0x402A)

Bit	7	6	5	4	3	2	1	0
Name	I2C_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	I2C_DR	I ² C Data Register						

8.2.4 I2C_SR (0x402B)

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

Bit	Name	Description															
7	I2CBSY	I²C Busy Flag When I2C_CR[I2CEN] = 0, this bit is cleared to “0” by hardware. Master Mode: This bit is set to “1” by hardware after START is sent, and cleared to “0” by hardware after STOP is sent. Slave Mode: This bit is set to “1” by hardware after START is received and address matches, and cleared to “0” by hardware after STOP is received.															
6	DMOD	I²C R/W Flag Master Mode: 0: Write (master sends the data, and slave receives the data) 1: Read (master receives the data, and slave sends the data) In this mode: ■ This bit can be configured only after I2C_SR[I2CSTA] is set to “1”. ■ A write of “1” to I2C_SR[I2CSTA] changes this bit as well. Slave Mode: 0: Write (master sends the data, and slave receives the data) 1: Read (master receives the data, and slave sends the data)															
5	RSV	Reserved															
4	I2CSTA	Master Mode: When this bit is set to “1” in software, START and address bytes are sent after both SCL and SDA are HIGH. This bit is cleared to “0” by hardware automatically when the transmission is completed, and I2C_SR[I2CSTA] writing is forbidden during data transmission. After the data is sent or received, I2C_SR[I2CSTA] is set to “1” to send RESTART. 0: Not START and address bytes 1: Send START or RESTART and address bytes Slave Mode: This bit is set to “1” after hardware receives START and address matches, and cleared to “0” by software. Table 8-1 Mapping of I2C_SR[I2CSTA] and I2C_SR[I2CSTP] in Slave Mode <table border="1"> <thead> <tr> <th>I2CSTA</th><th>I2CSTP</th><th>I²C Data Type</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Data byte</td></tr> <tr> <td>0</td><td>1</td><td>STOP</td></tr> <tr> <td>1</td><td>0</td><td>START + address bytes</td></tr> <tr> <td>1</td><td>1</td><td>STOP received first, then START + address bytes</td></tr> </tbody> </table> Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTA] is automatically cleared to “0”.	I2CSTA	I2CSTP	I ² C Data Type	0	0	Data byte	0	1	STOP	1	0	START + address bytes	1	1	STOP received first, then START + address bytes
I2CSTA	I2CSTP	I ² C Data Type															
0	0	Data byte															
0	1	STOP															
1	0	START + address bytes															
1	1	STOP received first, then START + address bytes															
3	I2CSTP	Master Mode: This bit cannot be written to “1” by software unless I2C_SR[I2CBSY] = 1. STOP is transmitted after I2C_SR[STR] is cleared. After the transmission, this bit is cleared to “0” automatically by hardware. If I2C_SR[I2CSTA] and I2C_SR[I2CSTP] are written to “1” at the same time and I2C_SR[I2CBSY] is “1”, I²C first sends STOP, then START and address bytes. After START and address bytes are transmitted,															

		I2C_SR[STR] is set to “1” by hardware. I2C_SR[I2CSTP] writing is forbidden during data transmission. 0: STOP is not transmitted. 1: STOP is transmitted. Slave Mode: This bit is set to “1” by hardware after STOP is received, and cleared to “0” by software. See Table 8-1 for status flags. Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTP] is automatically cleared to “0” by hardware.
2	STR	I²C Bus Pending Flag When this bit is set to “1”, SCL is pulled LOW for data transmission on bus. This bit is set to “1” by hardware and cleared to “0” by software. Master Mode: 1. After the hardware sends START + address byte and receives ACK/NACK; 2. After the hardware sends STOP and START + address bytes in sequence and receives ACK/NACK; 3. After the hardware sends data and receives ACK/NACK; 4. After the hardware receives the data; Slave Mode: 1. After the hardware receives START + address bytes; 2. After the hardware receives the data; 3. After the hardware sends ACK/NACK and data. Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTP] is automatically cleared to “0” by hardware.
1	NACK	This bit refers to the feedback from a receiver to a sender after a byte is transferred via I²C, i.e., the 9th bit of data. In sending mode, this bit is read-only and holds ACK/NACK from receiving device. In receiving mode, this bit can be read or written and is used to send ACK/NACK. Reading this bit obtains the written values. 0: ACK 1: NACK Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTP] is automatically cleared to “0” by hardware.
0	I2CIF	I²C Interrupt Flag After this bit is cleared, the data continues to be transferred via I²C. It is controlled by the hardware. 0: No Interrupt Pending 1: Interrupt Pending Master Mode: When I2C_SR[STR] = 1, this bit is set to “1”, otherwise, it is set to “0”. Slave Mode: When I2C_SR[I2CSTP] = 1 or I2C_SR[STR] = 1, this bit is set to “1”, otherwise, it is set to “0”. Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTP] is automatically cleared to “0” by hardware.

9 SPI

9.1 SPI Introduction

SPI provides access to a high-speed and full-duplex synchronous serial bus. SPI can operate as a master or slave device in 3-wire or 4-wire mode, and supports multiple masters and slaves on a single SPI bus. The four signals for SPI are MOSI, MISO, SCLK and NSS.

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave.

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. The MISO pin is placed in a high-impedance state when the SPI module is disabled or when the SPI operates in 4-wire mode as a slave that is not selected.

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

The slave-select (NSS) is dependent on the setting of SPI_CR1[NSSMOD] bit. When SPI operates in 3-Wire Mode, NSS pin is disabled and serves as the general-purpose I/O pin. When SPI operates as a slave, NSS pin is enabled as an input to detect NSS signals from the master. When the SPI works in single master/slave mode, NSS pin of the master is configured as an output to activate SPI of the slave. When SPI operates in multi-master mode, NSS pin is enabled as an input. In this mode, a particular SPI master function is disabled to prevent SPI bus collision where two or more masters simultaneously initiate data transfer. When SPI works in single-master multi-slave mode, multiple addressed slave devices can be selected by the configured I/O pins.

9.2 SPI Operations

9.2.1 SPI Master Mode

When SPI_CR0[SPIMS] = 1, SPI operates in master mode. When the data is written to SPI_DR, it is firstly written to the transmit buffer and SPI_CR1[TXBMT] is cleared to “0”. If the shift register is empty, the data in the transmit buffer will be transferred to the shift register for the transmission. The master SPI begins its transmission by driving the MSB of shift register on its MOSI pin. After the transmission is completed, SPI_CR1[SPIIF] and SPI_CR1[TXBMT] are set to “1”. The data in the shift register is that received by MISO, which is transferred to the receive buffer. The data from SPI_DR is that of the receive buffer. If the data is written to SPI_DR when SPI_CR1[TXBMT] is “0”, the write conflict flag bit SPI_CR1[WCOL] will be set to “1” and the data in the transmit buffer keeps unchanged.

9.2.1.1 Master Mode Configuration

1. Configure SPI_CR1[NSSMOD] to set the SPI operating mode;

2. Configure SPI_CR0[CPOL] to set the clock polarity;
3. Configure SPI_CR0[CPHA] to set the clock phase;
4. Set SPI_CR0[SPIMS] to “1” to select master mode;
5. Configure SPI_CLK to set the SCK rate;
6. Set SPI_CR1[SPIEN] to “1” to enable SPI;
7. Write the data to SPI_DR. SPI transmits data for each write;

9.2.2 SPI Slave Mode

When SPI_CR0[SPIMS] = 0, SPI operates in slave mode. In this mode, SCK signal is sent by the master SPI. If no SCK signal is input, shift register of the slave is in the stop state. If the signal of SCK is input, the shift register of slave starts to receive and transmit data through MOSI and MISO pins. After the transmission is completed, SPI_CR1[SPIIF] and SPI_CR1[TXBMT] are set to “1”. The receive buffer empty flag bit SPI_CR0[RXBMT] is cleared, indicating the new data has not been read. If SPI_CR0[RXBMT] is “0” and there is new data ready to be sent to the receive buffer, SPI_CR1[RXOVR] is set to “1” and the data in the receive buffer remains unaffected. When data is written to SPI_DR, SPI_CR1[TXBMT] is cleared. If data is written in this case, the write conflict flag bit SPI_CR1[WCOL] is set to “1” and the data in the transmit buffer keeps unchanged. If SPI operates in 4-wire mode, NSS pin is enabled as an input and the counter is reset when falling edge of the NSS signal is detected.

9.2.2.1 Slave Mode Configurations

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

9.2.3 SPI Interrupt Sources

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

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

- SPI interrupt flag SPI_CR1[SPIIF] is set to “1” each time after the byte is transferred. It applies to all SPI modes.
- If SPI_DR is written when the data in transmit buffer has not been transferred to the shift register, the write conflict flag SPI_CR1[WCOL] is set to “1” and the write operation will not be implemented. It applies to all SPI modes.

- When SPI works as a master in a multi-master system and NSS pin is pulled LOW, the mode error flag SPI_CR1[MODF] is set to “1”. When a mode error occurs, SPI_CR0[SPIMS] and SPI_CR1[SPIEN] are cleared. SPI is forbidden to allow another master to control the bus.
- The receive overflow flag SPI_CR1[RXOVRN] is set to “1” when SPI operates in slave mode and a transmission is completed while the receive buffer still holds unread data from a previous transfer. And the received data will not be transferred to the receive buffer.

9.2.4 SPI Operating Modes

SPI operates in 3-wire mode, 4-wire slave/multi-master mode or 4-wire single master mode, which is selected by SPI_CR1[NSSMOD].

When SPI_CR1[NSSMOD] = 00, SPI operates in 3-wire Mode. NSS pin is not necessary in this mode and there is only one master and one slave on the SPI bus. The connection diagram is shown in Figure 9-1.

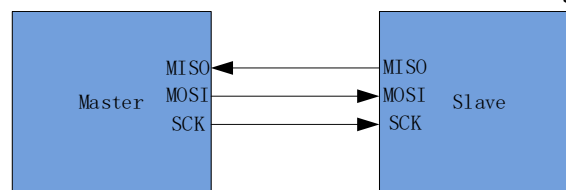


Figure 9-1 Connection Diagram of 3-Wire Single Master/Slave Mode

When SPI_CR1[NSSMOD] = 01, SPI operates in 4-wire mode. In this mode, NSS pin is enabled as an input, waiting to be addressed by the master. When SPI_CR0[SPIMS] = 0, SPI operates in 4-wire slave Mode. When SPI_CR0[SPIMS] = 1, SPI operates in 4-wire master mode. For multi-master mode, after NSS pin of a master on the SPI bus is pulled low, SPI_CR1[MODF] is set to “1” to disable the SPI and the master works as a slave. For slave mode, SPI is re-enabled after NSS pin of the slave is pulled LOW by two system cycles. The connection diagram of multi-master mode is shown in Figure 9-2.

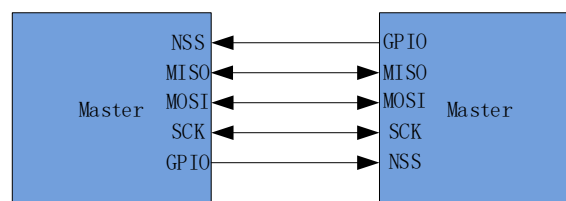


Figure 9-2 Connection Diagram of 4-Wire Multi-Master Mode

When SPI_CR1[NSSMOD] = 1X, SPI operates in 4-wire mode that is valid in master mode only. In this case, SPI_CR1[NSSMOD0] setting decides the output level of NSS pin. When SPI_CR1[NSSMOD0] = 1, NSS pin outputs high level, and when SPI_CR1[NSSMOD0] = 0, the pin outputs low level. The connection diagram is shown in Figure 9-3.

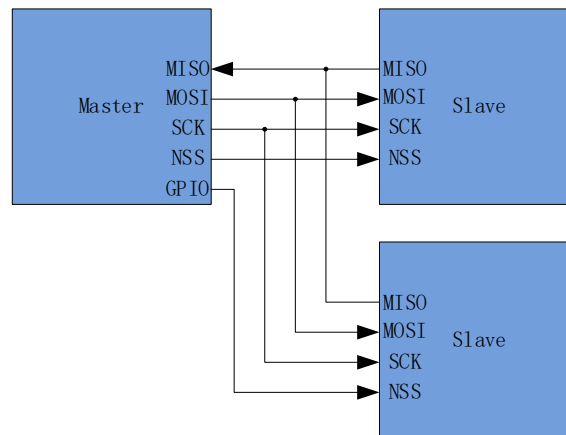


Figure 9-3 Connection Diagram of 4-Wire Single Master Mode

9.2.5 Serial Clock Timing

Four combinations of serial clock phase and idle polarity can be selected using the CPHA and CPOL bits in the SPI_CR0 Register. SPI_CR0[CPHA] selects the clock phase (the edge of the SCLK signal used to latch the data in shift register). SPI_CR0[CPOL] selects the polarity. Both master and slave devices must be configured with the same clock phase and polarity. When the clock phase and polarity is configured, SPI shall be disabled (SPI_CR1[SPIEN] = 0). The timing relationship between SCL and SDA in master mode is shown in Figure 9-4, and that in slave mode is shown in Figure 9-5 and Figure 9-6.

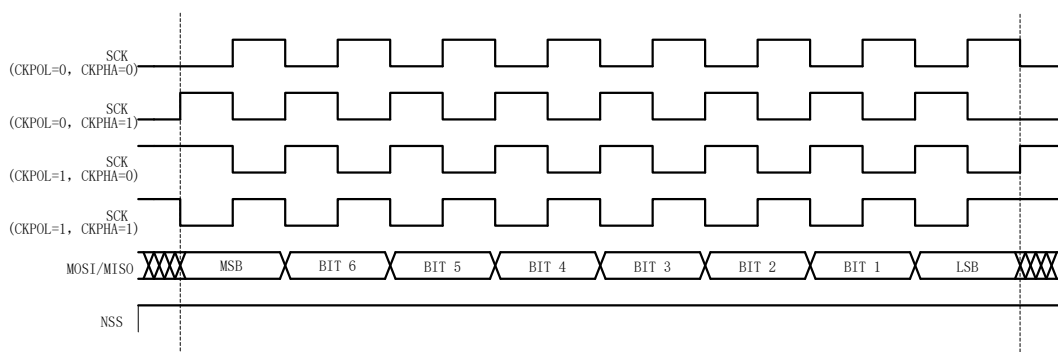


Figure 9-4 SDA/SCL Line Timing Diagram in Master Mode

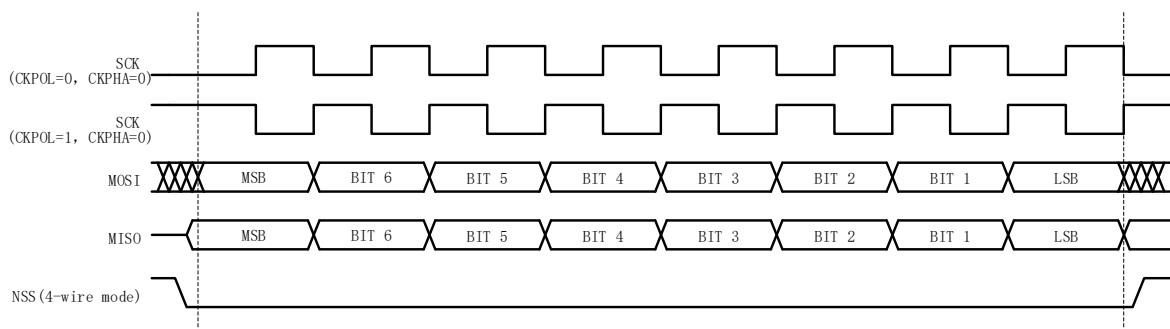


Figure 9-5 SDA/SCL Line Timing Diagram in Slave Mode (SPI_CR0[CPHA] = 0)

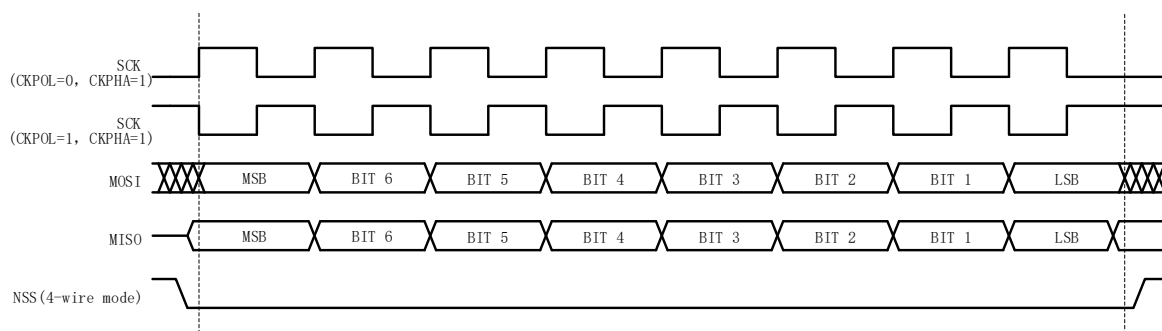


Figure 9-6 SDA/SCL Line Timing Diagram in Slave Mode ($SPI_CR0[CPHA] = 1$)

9.3 SPI Registers

9.3.1 SPI_CR0 (0x4030)

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

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

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

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

9.3.2 SPI_CR1 (0x4031)

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

Bit	Name	Description
[7]	SPIIF	SPI Interrupt Flag This bit is set to “1” by hardware each time after a data frame (8-bit) is transferred. It can be cleared to “0” by software only.
[6]	WCOL	Write Conflict Interrupt Flag When SPI_CR1[TXBMT] is “0”, a write to SPI_DR sets this bit to “1”. This bit can be cleared to “0” by software only.

[5]	MODF	Master Mode Fault Interrupt Flag This bit is set to “1” when a master mode conflict is detected (SPI_CR0[NSSIN] = 0, SPI_CR1[SPIMS] = 1 and SPI_CR1[NSSMOD] = 01). This bit can be cleared to “0” by software only.
[4]	RXOVR	Receive Overflow Interrupt Flag (Slave Mode only) This bit is set to “1” by hardware (and generates a SPI interrupt) when the Receive Buffer still holds unread data from a previous transfer and the last bit of the current transfer has been shifted into the SPI shift register. This bit cannot be clear to “0” automatically by hardware, and can be cleared to “0” by software only.
[3:2]	NSSMOD	NSS Operating Mode Selection 00: 3-wire slave or 3-wire master mode. NSS signal is not routed to a port pin. 01: 4-wire slave or multi-master mode (default). NSS pin is enabled as an input. 1X: 4-wire single-master mode. NSS pin is enabled as an output and outputs NSSMOD0 value.
[1]	TXBMT	Transmit Buffer Empty Flag This bit is cleared to “0” when new data is written to the Transit Buffer. It is set to “1” when the data in the Transit Buffer is transferred to the SPI shift register, indicating that it is safe to write a new byte to the transmit buffer.
[0]	SPIEN	SPI Enable 0: Disable 1: Enable

9.3.3 SPI_CLK (0x4032)

Bit	7	6	5	4	3	2	1	0
Name	SPI_CLK							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	SPI_CLK		SPI Clock Rate Setting This bit is valid in master mode only, and can be written only when SPI_CR1[SPIEN] = 0. $fsck = sysclk / (2 \times (SPI_CLK[7:0] + 1))$ For example, if sysclk = 24MHz and SPI_CLK = 0x04, Then $fsck = 24000000 / (2 \times (4 + 1)) = 2400kHz$					

9.3.4 SPI_DR (0x4033)

Bit	7	6	5	4	3	2	1	0
Name	SPI_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	SPI_DR		SPI Data Register SPI_DR Register is used to transmit and receive SPI data. In master mode, when the date is written to SPI_DR, it's transferred to Transit Buffer for the transmission. When SPI_DR is read, the data held in Receive Buffer is obtained.					

10 UART

10.1 UART Introductions

10.1.1 UART Mode0

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

Sending Data: Write the data to `UT_DR` and reset `UT_CR[TI]`. TXD pin outputs shift pulse, and RXD pin sends the data held by `UT_DR`. The output clock frequency is $SYS_CLK/12$. `UT_CR[TI]` is set to "1" after the transmission is completed.

Receiving Data: Set `UT_CR[REN]` to "1" to receive the data and clear `UT_CR[RI]`. TXD pin outputs shift pulse, and RXD pin receives the data. The shift pulse frequency is $SYS_CLK/12$. After the data is received, `UT_CR[RI]` is set to "1" and `UT_DR` is read to obtain the data.

10.1.2 UART Model

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

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

Receiving Data: Set `UT_CR[REN]` to "1" to receive the data and clear `UT_CR[RI]`. The data is received via RXD pin. After the data is received, `UT_CR[RB8]` and `UT_CR[RI]` are set to "1" and `UT_DR` is read to obtain the data.

10.1.3 UART Mode2

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

Sending Data: Write the data to `UT_DR`, set `UT_CR[TB8]` and reset `UT_CR[TI]`. TXD outputs 11-bit data. `UT_CR[TI]` is set to "1" after the transmission is completed.

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

10.1.4 UART Mode3

The operations are the same as those for UART mode 2, but baud rate is the same as that for UART model.

10.1.5 UART Interrupt Sources

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

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

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

10.2 UART Registers

10.2.1 UT_CR (0x98)

Bit	7	6	5	4	3	2	1	0
Name	MOD		SM2	REN	TB8	RB8	TI	RI
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:6]	MOD	Mode Selection 00: UART Mode0——Shift Register Baud rate: SYS_CLK/12 01: UART Mode1——8-bit UART Baud rate: SYS_CLK / (16 / (1 + UT_BAUD[BAUD_SEL])) / (UT_BAUD + 1) 10: UART Mode2——9-bit UART Baud rate: SYS_CLK/(32 – 16*UT_BAUD[BAUD_SEL]) 11: UART Mode3——9-bit UART Baud rate: SYS_CLK/(16 / (1 + UT_BAUD[BAUD_SEL])) / (UT_BAUD + 1)						
[5]	SM2	0: Single-device Communication 1: Multi-device Communication						
[4]	REN	UART Receive Enable 0: Disable 1: Enable. This bit can be cleared to “0” by software only.						
[3]	TB8	Bit9 of the sent data in UART Mode2 and UART Mode3. This bit can be cleared by hardware as required.						
[2]	RB8	Bit9 of the received data in UART Mode2 and UART Mode3. If the bit SM2 is set as “0”, it serves as the STOP bit. It does not work in UART Mode0.						
[1]	TI	Data Sending Completed Interrupt Flag. This bit is set to “1” after the data is sent, and can be cleared to “0” by software only.						
[0]	RI	Data Receiving Completed Interrupt Flag. This bit is set to “1” after the data is received, and can be cleared to “0” by software only.						

10.2.2 UT_DR (0x99)

Bit	7	6	5	4	3	2	1	0
Name	UT_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	UT_DR	Send/Receive Data						

10.2.3 UT_BAUD (0x9A,0x9B)

UT_BAUDH(0x9B)								
Bit	15	14	13	12	11	10	9	8
Name	BAUD_SEL	RSV			UT_BAUDH[11:8]			
Type	R/W	-	-	-	R/W	R/W	R/W	R/W
Reset	0	-	-	-	0	0	0	0
UT_BAUDL(0x9A)								
Bit	7	6	5	4	3	2	1	0
Name	UT_BAUDL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	1	0	1	1
Bit	Name	Description						
[15]	BAUD_SEL	Frequency Multiplier Enable						
[14:12]	RSV	Reserved						
[11:0]	UT_BAUD	Baud rate setting in UART Mode1 and UART Mode3						

11 MDU

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

Multiply Mode: $MD_MC3\sim0 = \{MD_MAH \text{ and } MD_MAL\} * \{MD_MBH \text{ and } MD_MBL\}$

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

Division Mode: $MD_DC3\sim0, MD_DD1\sim0 = MD_DA3\sim0 / MD_DB1\sim0$

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

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

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

11.1 Multiplication Configurations

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

11.2 Division Configurations

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

11.3 Important Matters

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

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

11.4 MDU Registers

11.4.1 MDU_CR (0xC1)

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

11.4.2 MD_MBL (0xCA)

Bit	7	6	5	4	3	2	1	0
Name	MD_MBL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	MD_MBL	Multiplier bit [7:0] (write only) or product bit [7:0] (read only)						

11.4.3 MD_MBH (0xCB)

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

11.4.4 MD_MAL (0xC2)

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

11.4.5 MD_MAH (0xC3)

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

11.4.6 MD_DA0 (0xC4)

Bit	7	6	5	4	3	2	1	0
Name	MD_DA0							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	MD_DA0	Dividend bit [7:0] (write only) or quotient bit [7:0] (read only)						

11.4.7 MD_DA1 (0xC5)

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

11.4.8 MD_DA2 (0xC6)

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

11.4.9 MD_DA3 (0xC7)

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

11.4.10 MD_DB0 (0xCC)

Bit	7	6	5	4	3	2	1	0
Name	MD_DB0							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	MD_DB0	Divisor bit [7:0] (write only) or remainder bit [7:0] (read only)						

11.4.11 MD_DB1 (0xCD)

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

12 PI

12.1 PI Operations

1. PI algorithm:
$$U(k) = U(k-1) + K_p * (E(k) - E(k-1)) + K_i * E(k) \text{ ----- } (U_{k_min} < U(k) < U_{k_max})$$
2. Configuring PI_LPF_CR[PISTA] = 1 to enable PI. After four clock cycles, the calculation is completed and PI_UK is updated.
3. Q12 (default) or Q15 can be selected for PI parameters. It means data format of PI_KP and PI_KI is Q12 and that of other registers are Q15.
4. U(k-1) and E(k-1) default to the previous U(k) and E(k). You can write the data to PI_EK register and start PI to change E(k-1), or write the data to PI_UK register to change U(k-1).

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

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


12.2 PI Registers

12.2.1 PI_LPF_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2SS	RSV				PIRANGE	PISTA	LPFSTA
Type	R/W	-	-	-	-	R/W	R/W	R/W
Reset	0	-	-	-	-	0	0	0
Bit	Name	Description						
[7]	T2SS	Input Mode Selection of TIM2 Step Motor 0: P10 for direction input, and P07 for pulse input 1: P10 for backward pulse input, and P07 for forward pulse input						
[6:3]	RSV	Reserved						
[2]	PIRANGE	Data Format of PI Parameters 0: Q12. The value range of KP and KI (-32768,32767) corresponds to range (-8,8). 1: Q15. The value range of KP and KI (-32768,32767) corresponds to range (-1,1).						
[1]	PISTA	PI Regulator Enable Bit. It is set to "1" by software, and can be automatically cleared to "0" by hardware. 0: Disable 1: Enable						
[0]	LPFSTA	LPF Enable Bit. It is set to "1" by software, and can be automatically cleared to "0" by hardware. 0: Disable 1: Enable						

12.2.2 PI_EK (0xEA, 0xEB)

PI_EKH(0xEB)								
Bit	15	14	13	12	11	10	9	8
Name	PI_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI_EKL(0xEA)								
Bit	7	6	5	4	3	2	1	0
Name	PI_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI_EK	Input Deviation; Range (-32768, 32767)						

12.2.3 PI_UK (0xEC, 0xED)

PI_UKH(0xED)								
Bit	15	14	13	12	11	10	9	8
Name	PI_UK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI_UKL(0xEC)								
Bit	7	6	5	4	3	2	1	0
Name	PI_UK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	PI_UK	Calculation result of PI regulator Range (-32768, 32767)

12.2.4 PI_KP (0xEE, 0xEF)

PI_KPH(0xEF)								
Bit	15	14	13	12	11	10	9	8
Name	PI_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI_KPL(0xEE)								
Bit	7	6	5	4	3	2	1	0
Name	PI_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI_KP	KP Coefficient Range (-32768, 32767)						

12.2.5 PI_KI (0xF2, 0xF3)

PI_KIH(0xF3)								
Bit	15	14	13	12	11	10	9	8
Name	PI_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI_KIL(0xF2)								
Bit	7	6	5	4	3	2	1	0
Name	PI_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI_KI	KI Coefficient Range (-32768,32767)						

12.2.6 PI_UKMAX (0xF4, 0xF5)

PI_UKMAXH(0xF5)								
Bit	15	14	13	12	11	10	9	8
Name	PI_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI_UKMAXL(0xF4)								
Bit	7	6	5	4	3	2	1	0
Name	PI_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI_UKMAX	UK Maximum Value; Range (-32768,32767)						

12.2.7 PI_UKMIN (0xF6, 0xF7)

PI_UKMINH(0xF7)								
Bit	15	14	13	12	11	10	9	8
Name	PI_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI_UKMINL(0xF6)								
Bit	7	6	5	4	3	2	1	0
Name	PI_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI_UKMIN	UK Minimum Value Range (-32768, 32767)						

13 LPF

13.1 LPF Operations

1. LPF algorithm:

$$Y(k) = Y(k-1) + LPF_K * (X(k) - Y(k-1))$$
2. Configuring PI_LP_F_CR[LPFSTA] = 1 to enable LPF. After four clock cycles, LPF_Y is updated.
3. Y(k-1) default to the previous Y(k). You can write the data to LPF_Y register to change Y(k-1).

13.2 LPF Registers

13.2.1 PI_LP_F_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2SS	RSV				PIRANGE	PISTA	LPFSTA
Type	R/W	-	-	-	-	R/W	R/W	R/W
Reset	0	-	-	-	-	0	0	0
Bit	Name	Description						
[7:1]		See PI_LP_F_CR (0xF9) for details.						
[0]	LPFSTA	LPF Enable Bit. It is set to "1" by software, and can be automatically cleared to "0" by hardware. 0: Disable 1: Enable						

13.2.2 LPF_K (0xDD)

Bit	7	6	5	4	3	2	1	0
Name	LPF_K							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	LPF_K	LPF Coefficient The value range (-128,127) corresponds to range (-1,1).						

13.2.3 LPF_X (0xDE, 0xDF)

LPF_XH(0xDF)								
Bit	7	6	5	4	3	2	1	0
Name	LPF_X[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF_XL(0xDE)								
Bit	7	6	5	4	3	2	1	0
Name	LPF_X[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	LPF_X	Input Value Range (-32768, 32767)						

13.2.4 LPF_Y (0xE6, 0xE7)

LPF_YH(0xE7)								
Bit	7	6	5	4	3	2	1	0
Name	LPF_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF_YL(0xE6)								
Bit	7	6	5	4	3	2	1	0
Name	LPF_Y[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	LPF_Y	Output Value Range (-32768, 32767)						

14 SMDU

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

14.1 SMDU Features

The SMDU module has the following features:

- Support invocation with nested interrupt
- Hardware acceleration to reduce CPU load
- Support the following modes:
 - 16-bit signed multiplication
 - 16-bit signed multiplication (result shifted left by one-bit)
 - 16-bit unsigned multiplication
 - 32-bit/16-bit unsigned division
 - LPF
 - Coordinate transformation (SIN/COS)
 - Arctangent (ATAN)
 - PI

14.2 SMDU Instructions

14.2.1 SMDU Operations

SMDU is operated as follows.

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

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

14.2.2 16-bit Signed Multiplication with the Result Shifted Left by One-bit

When MDU_CR[MDUMOD] = 000, SMDU module works in the 16-bit signed multiplication mode with the result shifted left by one-bit. As shown in Table 14-1, after 16-bit signed data is written to MULx_MA and MULx_MB as multiplicand and multiplier respectively, 32-bit signed data is obtained by

the product shifting left by one bit. The result is accessed by reading MULx_MC register.

Table 14-1 Register Definitions in 16-bit Signed Multiplication Mode with Result Shifted Left by One-bit

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

14.2.3 16-bit Signed Multiplication

When MDU_CR[MDUMOD] = 001, SMDU module works in the 16-bit signed multiplication mode. As shown in Table 14-2, 32-bit signed data is obtained after 16-bit signed data is written to MULx_MA and MULx_MB as multiplicand and multiplier respectively. The result is accessed by reading MULx_MC register.

Table 14-2 Register Definitions in 16-bit Signed Multiplication Mode

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

14.2.4 16-bit Unsigned Multiplication

When MDU_CR[MDUMOD] = 010, SMDU module works in the 16-bit unsigned multiplication mode. As shown in Table 14-3, 32-bit unsigned data is obtained after 16-bit unsigned data is written to MULx_MA and MULx_MB as multiplicand and multiplier respectively. The result is accessed by reading MULx_MC register.

Table 14-3 Register Definitions in 16-bit Unsigned Multiplication Mode

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

14.2.5 32-bit/16-bit Unsigned Division

When MDU_CR[MDUMOD] = 011, SMDU module works in the 32-bit/16-bit unsigned division mode. As shown in Table 14-4, 32-bit unsigned quotient and 16-bit unsigned remainder is obtained after 32-bit unsigned dividend and a 16-bit unsigned divisor are written to DIVx_DA and DIVx_DB registers respectively. The quotient and remainder are accessed by reading DIVx_DQ and DIVx_DR registers respectively.

Table 14-4 Register Definitions in the Unsigned Division Mode

Data Register	Input	Output
DIVx_DA	Dividend	-
DIVx_DB	Divisor	-
DIVx_DQ	-	Quotient
DIVx_DR	-	Remainder

14.2.6 LPF

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

The calculation formula of LPF:

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

Where,

Y_k : Filtered value

Y_{k-1} : Previous filtered output

K : Filter coefficient

X_k : Value to be filtered

As shown in Table 14-5, Y_k and Y_{k-1} are 32-bit signed data, and X_k and K are 16-bit signed data. Y_k is obtained after Y_{k-1} is written to LPFx_Y, K to LPFx_K and X_k to LPFx_X, and is accessed by reading LPFx_Y.

Table 14-5 Register Definitions in LPF Mode

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

14.2.7 Coordinate Transformation (SIN/COS)

When MDU_CR[MDUMOD] = 100, SMDU module works in Coordinate Transformation mode. As shown in Figure 14-1, the coordinate transformation converts the components $\cos_i$ and $\sin_i$ of vector A under the x-y axis to the components $\cos_o$ and $\sin_o$ under the x'-y' axis, with the x'-y' axis lagging the x-y axis by θ .

The formula for coordinate transformation is:

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

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

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

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

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

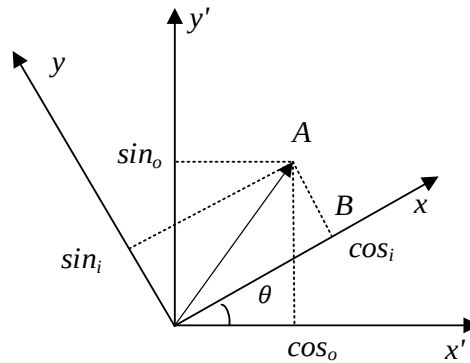


Figure 14-1 Coordinate Transformation

As shown in Table 14-6, $\cos_i$, $\sin_i$, θ , $\cos_o$ and $\sin_o$ are all 16-bit signed data. $\cos_i$ is written to SCATx_COS, $\sin_i$ to SCATx_SIN and θ to SCATx_THE to calculate $\cos_o$ and $\sin_o$. The results $\cos_o$ and $\sin_o$ are accessed by reading SCATx_RES1 and SCATx_RES2 respectively.

Table 14-6 Register Definitions in the Coordinate Transformation Mode

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

14.2.8 Arctangent

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

ATAN calculates the amplitude and angle of a vector based on sine and cosine inputs. The calculation formula is:

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

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

Where,

$U \sin \theta$: Sin component of the vector

$U \cos \theta$: Cosine component of the vector

θ : Calculated vector angle

U : Calculated vector amplitude

As shown in Table 14-7, $U \cos \theta$ and $U \sin \theta$, U and θ are 16-bit signed data. $U \cos \theta$ is written to SCATx_COS and $U \sin \theta$ to SCATx_SIN to calculate U and θ . U and θ are accessed by reading SCATx_RES1 and SCATx_RES2 respectively.

Table 14-7 Register Definitions in ATAN Mode

Data Register	Input	Output
SCATx COS	$U \cos \theta$	-
SCATx SIN	$U \sin \theta$	-
SCATx RES1	-	U
SCATx RES2	-	θ

14.2.9 PI

14.2.9.1 PI Introduction

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

PI algorithm:

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

Where,

U_k : Output for round k of calculation

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

E_k : Deviation for round k of input

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

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

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

14.2.9.2 PI Features

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

14.2.9.3 PI Operations

1. Initialize SMDU before operations, and configure K_p , K_i and the maximum and minimum values of U_k ;
2. Set MDU_CR[MDUMOD] to 111, and then configure MDU_CR[MDUSTA] bit to select the desired computing unit and start PI computing. At this time, busy flag MDU_CR[MDUBUSY] is automatically set to “1”;
3. Read MDU_CR[MDUBUSY] bit by software. When this bit is 0, it indicates that the calculation is completed, and calculation result PIx_UK is updated.

4. Read PIx_UK to obtain the output.

Notes:

- The data format of PI_KP is Q12 and that of other registers are Q15.
- PIx_UK and PIx_EK1 values default to the previous calculated U_k and E_k . The related values change after PIx_EK1 and PIx_UK are written.
- When PI controller is invoked repeatedly, relevant parameters shall be saved after each PI operation, and initialized before the next PI operation. Initialization codes are shown as below:

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

14.3 SMDU Registers

14.3.1 MDU_CR (0xC1)

MDU_CR(0xC1)								
Bit	7	6	5	4	3	2	1	0
Name	MDUBSY	MDUSTA3	MDUSTA2	MDUSTA1	MDUSTA0	MDUMOD2	MDUMOD1	MDUMOD0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7]	MDUBSY		After SMDU module is enabled, this bit is used to indicate operating status of the module. 0: Idle 1: Busy					
[6]	MDUSTA3		After SMDU module is enabled, this bit is used to enable or disable computing unit 3. 0: Disable 1: Enable					
[5]	MDUSTA2		After SMDU module is enabled, this bit is used to enable or disable computing unit 2. 0: Disable 1: Enable					
[4]	MDUSTA1		After SMDU module is enabled, this bit is used to enable or disable computing unit 1. 0: Disable 1: Enable					
[3]	MDUSTA0		After SMDU module is enabled, this bit is used to enable or disable computing unit 0. 0: Disable 1: Enable					
[2:0]	MDUMOD [2:0]		After SMDU module is enabled, this bit is used to select operating mode of the module. 000: Signed Multiplication (the result shifted left by one-bit) 001: Signed Multiplication 010: Unsigned Multiplication 011: 32-bit/16-bit Unsigned Division 100: Coordinate Transformation (SIN/COS) 101: ATAN 110: LPF 111: PI					

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

14.3.2 TIM234_CTRL(0x40f1)

TIM234_CTRL(0x40f1)								
Bit	7	6	5	4	3	2	1	0
Name	RSV	MDU_EN_N	TIM2_FAST_DI_R	TIM2_D_R_SEL	TIM4_R_CTRL	TIM3_R_CTRL	TIM2_R_CTRL	TIM3_48M
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
Bit	Name		Description					
[7]	RSV		Reserved					

[6]	MDU_EN_N	When this bit is set to “1”, SMDU module is enabled and MDU module is disabled. 0: SMDU module is disabled. 1: SMDU module is enabled.
[5]	TIM2_FAST_DIR	After this bit is enabled in TIM2 QEP mode, SMDU module can quickly identify the direction. 0: Disable 1: Enable
[4]	TIM2_DR_SEL	After this bit is enabled in TIM2 QEP mode, the timer is reset to 0 or TIM2_DR when it reaches to TIM2_DR or 0. 0: Disable 1: Enable
[3]	TIM4_RCTRL	After this bit is enabled, DR = 0/0xffff when TIM4_CNTR overflows. 0: Disable 1: Enable
[2]	TIM3_RCTRL	After this bit is enabled, DR = 0/0xffff when TIM3_ARR overflows. 0: Disable 1: Enable
[1]	TIM2_RCTRL	After this bit is enabled, DR = 0/0xffff when TIM2_ARR overflows. 0: Disable 1: Enable
[0]	TIM3_48M	TIM3_48M Input Capture Enable 0: Disable 1: Enable

14.3.3 MUL0_MA (0x03C8, 0x03C9)

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

14.3.4 MUL0_MB (0x03CA, 0x03CB)

MUL0_MBH(0x03CA)								
Bit	15	14	13	12	11	10	9	8
Name	MUL0_B[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL0_MBL(0x03CB)								
Bit	7	6	5	4	3	2	1	0
Name	MUL0_B[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL0_MB	Data register B of MUL0; Multiplier of the multiplication

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

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

14.3.6 MUL1_MA (0x03C0, 0x03C1)

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

14.3.7 MUL1_MB (0x03C2, 0x03C3)

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

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

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

14.3.9 MUL2_MA (0x0370, 0x0371)

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

14.3.10 MUL2_MB (0x0372, 0x0373)

MUL2_MBH(0x0372)								
Bit	15	14	13	12	11	10	9	8
Name	MUL2_MB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MBL(0x0373)								
Bit	7	6	5	4	3	2	1	0
Name	MUL2_MB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	MUL2_MB	Data register B of MUL2; Multiplier of the multiplication						

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

MUL2_MCHH(0x0374)								
Bit	31	30	29	28	27	26	25	24
Name	MUL2_MC[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MCHL(0x0375)								
Bit	23	22	21	20	19	18	17	16
Name	MUL2_MC[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MCLH(0x0376)								
Bit	15	14	13	12	11	10	9	8
Name	MUL2_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MCLL(0x0377)								

Bit	7	6	5	4	3	2	1	0
Name	MUL2_MC[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	MUL2_MC	Product of Register MUL2. The 16 high-order bits of the data are held by MUL2_MCH and the 16 low-order bits by MUL2_MCL.						

14.3.12 MUL3_MA (0x0368, 0x0369)

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

14.3.13 MUL3_MB (0x036A, 0x036B)

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

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

MUL3_MCHH(0x036C)								
Bit	31	30	29	28	27	26	25	24
Name	MUL3_MC[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MCHL(0x036D)								
Bit	23	22	21	20	19	18	17	16

Name	MUL3_MC[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MCLH(0x036E)								
Bit	15	14	13	12	11	10	9	8
Name	MUL3_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MCLL(0x036F)								
Bit	7	6	5	4	3	2	1	0
Name	MUL3_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[31:0]	MUL3_MC		Product of MUL3. The 16 high-order bits of the data are held by MUL3_MCH and the 16 low-order bits by MUL3_MCL.					

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

DIV0_DAHH(0x03B4)								
Bit	31	30	29	28	27	26	25	24
Name	DIV0_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_D AHL(0x03B5)								
Bit	23	22	21	20	19	18	17	16
Name	DIV0_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_D ALH(0x03B6)								
Bit	15	14	13	12	11	10	9	8
Name	DIV0_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_D ALL(0x03B7)								
Bit	7	6	5	4	3	2	1	0
Name	DIV0_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[31:0]	DIV0_DA		Dividend of DIV0. The 16 high-order bits of the data are held by DIV0_DAH and the 16 low-order bits by DIV0_DAL.					

14.3.16 DIV0_DB (0x03B8, 0x03B9)

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

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

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

14.3.18 DIV0_DR (0x03BE, 0x03BF)

DIV0_DRH(0x03BE)								
Bit	15	14	13	12	11	10	9	8
Name	DIV0_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

DIV0_DRL(0x03BF)								
Name	DIV0_DR[7:0]							
Bit	7	6	5	4	3	2	1	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	DIV0_DR	Remainder of DIV0						

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

DIV1_DAHH(0x03A8)								
Bit	31	30	29	28	27	26	25	24
Name	DIV1_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_D AHL(0x03A9)								
Bit	23	22	21	20	19	18	17	16
Name	DIV1_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_D ALH(0x03AA)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_D ALL(0x03AB)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	DIV1_DA	Dividend of DIV1. The 16 high-order bits of the data are held by DIV1_DA and the 16 low-order bits by DIV1_DA.						

14.3.20 DIV1_DB (0x03AC, 0x03AD)

DIV1_DBH(0x03AC)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_D BL(0x03AD)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	DIV1_DB	Data register B of DIV1; Divisor of the division

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

DIV1_DQHH(0x03AE)								
Bit	31	30	29	28	27	26	25	24
Name	DIV1_DQ[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DQHL(0x03AF)								
Bit	23	22	21	20	19	18	17	16
Name	DIV1_DQ[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DQLH(0x03B0)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DQLL(0x03B1)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	DIV1_DQ	Quotient of DIV1. The 16 high-order bits of the data are held by DIV1_DQH and the 16 low-order bits by DIV1_DQL.						

14.3.22 DIV1_DR (0x03B2, 0x03B3)

DIV1_DRH(0x03B2)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DRL(0x03B3)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	DIV1_DR	Remainder of DIV1						

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

DIV2_DAHH(0x035C)								
Bit	31	30	29	28	27	26	25	24
Name	DIV2_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

DIV2_DAH(0x035D)								
Bit	23	22	21	20	19	18	17	16
Name	DIV2_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DALH(0x035E)								
Bit	15	14	13	12	11	10	9	8
Name	DIV2_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DALL(0x035F)								
Bit	7	6	5	4	3	2	1	0
Name	DIV2_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	DIV2_DA	Dividend of DIV2. The 16 high-order bits of the data are held by DIV2_DA and the 16 low-order bits by DIV2_DA.						

14.3.24 DIV2_DB (0x0360, 0x0361)

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

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

DIV2_DQHH(0x0362)								
Bit	31	30	29	28	27	26	25	24
Name	DIV2_DQ[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DQHL(0x0363)								
Bit	23	22	21	20	19	18	17	16
Name	DIV2_DQ[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DQLH(0x0364)								

Bit	15	14	13	12	11	10	9	8
Name	DIV2_DQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DQLL(0x0365)								
Bit	7	6	5	4	3	2	1	0
Name	DIV2_DQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	DIV2_DQ	Quotient of DIV2. The 16 high-order bits of the data are held by DIV2_DQH and the 16 low-order bits by DIV2_DQL.						

14.3.26 DIV2_DR (0x0366, 0x0367)

DIV2_DRH(0x0366)								
Bit	15	14	13	12	11	10	9	8
Name	DIV2_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DRL(0x0367)								
Bit	7	6	5	4	3	2	1	0
Name	DIV2_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	DIV2_DR	Remainder of DIV2						

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

DIV3_DAHH(0x0350)								
Bit	31	30	29	28	27	26	25	24
Name	DIV3_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_D AHL(0x0351)								
Bit	23	22	21	20	19	18	17	16
Name	DIV3_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_D ALH(0x0352)								
Bit	15	14	13	12	11	10	9	8
Name	DIV3_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_D ALL(0x0353)								
Bit	7	6	5	4	3	2	1	0

Name	DIV3_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	DIV3_DA	Dividend of DIV3. The 16 high-order bits of the data are held by DIV3_DAH and the 16 low-order bits by DIV3_DAL.						

14.3.28 DIV3_DB (0x0354, 0x0355)

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

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

DIV3_DQHH(0x0356)								
Bit	31	30	29	28	27	26	25	24
Name	DIV3_DQ[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DQHL(0x0357)								
Bit	23	22	21	20	19	18	17	16
Name	DIV3_DQ[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DQLH(0x0358)								
Bit	15	14	13	12	11	10	9	8
Name	DIV3_DQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DQLL(0x0359)								
Bit	7	6	5	4	3	2	1	0
Name	DIV3_DQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	DIV3_DQ	Quotient of DIV3. The 16 high-order bits of the data are held by DIV3_DQH and the 16 low-order bits by DIV3_DQL.						

14.3.30 DIV3_DR (0x035A, 0x035B)

DIV3_DRH(0x035A)								
Bit	15	14	13	12	11	10	9	8
Name	DIV3_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DRL(0x035B)								
Bit	7	6	5	4	3	2	1	0
Name	DIV3_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	DIV3_DR	Remainder of DIV3						

14.3.31 SCAT0_COS (0x0346, 0x0347)

SCAT0_COSH(0x0346)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT0_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT0_COSL(0x0347)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT0_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	SCAT0_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT0						

14.3.32 SCAT0_SIN (0x0348, 0x0349)

SCAT0_SINH(0x0348)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT0_SIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT0_SINL(0x0349)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT0_SIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	SCAT0_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT0						

14.3.33 SCAT0_THE (0x034A, 0x034B)

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

14.3.34 SCAT0_RES1 (0x034C, 0x034D)

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

14.3.35 SCAT0_RES2 (0x034E, 0x034F)

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

14.3.36 SCAT1_COS (0x033C, 0x033D)

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

14.3.37 SCAT1_SIN (0x033E, 0x033F)

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

14.3.38 SCAT1_THE (0x0340, 0x0341)

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

14.3.39 SCAT1_RES1 (0x0342, 0x0343)

SCAT1_RES1H(0x0342)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT1_RES1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT1_RES1L(0x0343)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT1_RES1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	SCAT1_RES1	COS output in SIN/COS mode of computing unit SCAT1; U output in ATAN mode						

14.3.40 SCAT1_RES2 (0x0344, 0x0345)

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

14.3.41 SCAT2_COS (0x0332, 0x0333)

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

14.3.42 SCAT2_SIN (0x0334, 0x0335)

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

14.3.43 SCAT2_THE (0x0336, 0x0337)

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

14.3.44 SCAT2_RES1 (0x0338, 0x0339)

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

14.3.45 SCAT2_RES2 (0x033A, 0x033B)

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

14.3.46 SCAT3_COS (0x0328, 0x0329)

SCAT3_COSH(0x0328)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT3_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT3_COSL(0x0329)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT3_COS[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	SCAT3_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT3						

14.3.47 SCAT3_SIN (0x032A, 0x032B)

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

14.3.48 SCAT3_THE (0x032C, 0x032D)

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

14.3.49 SCAT3_RES1 (0x032E, 0x032F)

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

14.3.50 SCAT3_RES2 (0x0330, 0x0331)

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

14.3.51 LPF0_K (0x03F8, 0x03F9)

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

14.3.52 LPF0_X (0x03FA 0x03FB)

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

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

LPF0_YHH(0x03FC)								
Bit	31	30	29	28	27	26	25	24
Name	LPF0_Y[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_YHL(0x03FD)								
Bit	23	22	21	20	19	18	17	16
Name	LPF0_Y[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_YLH(0x03FE)								
Bit	15	14	13	12	11	10	9	8
Name	LPF0_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_YLL(0x03FF)								
Bit	7	6	5	4	3	2	1	0

Name	LPF0_Y[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	LPF0_Y[31:0]	Input and output of the register in LPF0 Input: LPF0_Y _{k-1} Output: LPF0_Y _k						

14.3.54 LPF1_K (0x03F0, 0x03F1)

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

14.3.55 LPF1_X (0x03F2, 0x03F3)

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

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

LPF1_YHH(0x03F4)								
Bit	31	30	29	28	27	26	25	24
Name	LPF1_Y[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF1_YHL(0x03F5)								
Bit	23	22	21	20	19	18	17	16
Name	LPF1_Y[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF1_YLH(0x03F6)								

Bit	15	14	13	12	11	10	9	8
Name	LPF1_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF1_YLL(0x03F7)								
Bit	7	6	5	4	3	2	1	0
Name	LPF1_Y[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[31:0]	LPF1_Y[31:0]	Input and output of the register in LPF1 Input: LPF1_Y _{k-1} Output: LPF1_Y _k						

14.3.57 LPF2_K (0x03A0, 0x03A1)

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

14.3.58 LPF2_X (0x03A2, 0x03A3)

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

14.3.59 LPF2_Y (0x03A4, 0x03A5, 0x03A6, 0x03A7)

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

14.3.60 LPF3_K (0x0398, 0x0399)

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

14.3.61 LPF3_X (0x039A, 0x039B)

LPF3_XH(0x039A)								
Bit	15	14	13	12	11	10	9	8
Name	LPF3_X[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF3_XL(0x039B)								
Bit	7	6	5	4	3	2	1	0
Name	LPF3_K[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	LPF3_K[15:0]		X input of LPF3					

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

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

14.3.63 PI0_KP (0x03E0, 0x03E1)

PI0_KPH(0x03E0)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_KPL(0x03E1)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI0_KP	Proportional coefficient of PI0						

14.3.64 PI0_EK1 (0x03E2, 0x03E3)

PI0_EK1H(0x03E2)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_EK1L(0x03E3)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI0_EK1	Previous input deviation of PI0						

14.3.65 PI0_EK (0x03E4, 0x03E5)

PI0_EKH(0x03E4)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_EKL(0x03E5)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI0_EK	Present input deviation of PI0						

14.3.66 PI0_KI (0x03E6, 0x03E7)

PI0_KIH(0x03E6)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_KIL(0x03E7)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI0_KI	Integral coefficient of PI0						

14.3.67 PI0_UKH (0x03E8, 0x03E9)

PI0_UKHH(0x03E8)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKHL(0x03E9)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI0_UKH	16 high-order bits of PI0 output						

14.3.68 PI0_UKL (0x03EA, 0x03EB)

PI0_UKLH(0x03EA)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKLL(0x03EB)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI0_UKL	16 low-order bits of PI0 output						

14.3.69 PI0_UKMAX (0x03EC, 0x03ED)

PI0_UKMAXH(0x03EC)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKMAXL(0x03ED)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	PI0_UKMAX		Maximum output of PI0					

14.3.70 PI0_UKMIN (0x03EE, 0x03EF)

PI0_UKMINH(0x03EE)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKMINL(0x03EF)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	PI0_UKMIN		Minimum output of PI0					

14.3.71 PI1_KP (0x03D0, 0x03D1)

PI1_KPH(0x03D0)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_KPL(0x03D1)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	PI1_KP		Proportional coefficient of PI1					

14.3.72 PI1_EK1 (0x03D2, 0x03D3)

PI1_EK1H(0x03D2)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_EK1L(0x03D3)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI1_EK1	Previous input deviation of PI1						

14.3.73 PI1_EK (0x03D4, 0x03D5)

PI1_EKH(0x03D4)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_EKL(0x03D5)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI1_EK	Present input deviation of PI1						

14.3.74 PI1_KI(0x03D6, 0x03D7)

PI1_KIH(0x03D6)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_KIL(0x03D7)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI1_KI	Integral coefficient of PI1						

14.3.75 PI1_UKH (0x03D8, 0x03D9)

PI1_UKHH(0x03D8)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKHL(0x03D9)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI1_UKH	16 high-order bits of PI1 output						

14.3.76 PI1_UKL (0x03DA, 0x03DB)

PI1_UKLH(0x03DA)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKLL(0x03DB)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI1_UKL	16 low-order bits of PI1 output						

14.3.77 PI1_UKMAX (0x03DC, 0x03DD)

PI1_UKMAXH(0x03DC)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKMAXL(0x03DD)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI1_UKMAX	Maximum output of PI1						

14.3.78 PI1_UKMIN (0x03DE, 0x03DF)

PI1_UKMINH(0x03DE)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKMINL(0x03DF)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI1_UKMIN	Minimum output of PI1						

14.3.79 PI2_KP (0x0388, 0x0389)

PI2_KPH(0x0388)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KPL(0x0389)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_KP	Proportional coefficient of PI2						

14.3.80 PI2_EK1 (0x038A, 0x038B)

PI2_EK1H(0x038A)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EK1L(0x038B)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_EK1	Previous input deviation of PI2						

14.3.81 PI2_EK (0x038C, 0x038D)

PI2_EKH(0x038C)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EKL(0x038D)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_EK	Present input deviation of PI2						

14.3.82 PI2_KI (0x038E, 0x038F)

PI2_KIH(0x038E)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KIL(0x038F)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_KI	Integral coefficient of PI2						

14.3.83 PI2_UKH (0x0390, 0x0391)

PI2_UKHH(0x0390)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKHL(0x0391)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_UKH	16 high-order bits of PI2 output						

PI2_UKL (0x0392, 0x0393)

PI2_UKLH(0x0392)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKLL(0x0393)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_UKL	16 low-order bits of PI2 output						

14.3.84 PI2_MAX (0x0394, 0x0395)

PI2_MAXH(0x0394)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_MAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_MAXL(0x0395)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_MAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_MAX	Maximum output of PI2						

14.3.85 PI2_MIN (0x0396, 0x0397)

PI2_MINH(0x0396)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_MIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_MINL(0x0397)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_MIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI2_MIN	Minimum output of PI2						

14.3.86 PI3_KP (0x0378, 0x0379)

PI3_KPH(0x0378)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KPL (0x0379)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_KP	Proportional coefficient of PI3						

14.3.87 PI3_EK1 (0x037A, 0x037B)

PI3_EK1H(0x037A)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EK1L(0x037B)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_EK1	Previous input deviation of PI3						

14.3.88 PI3_EK (0x037C, 0x037D)

PI3_EKH(0x037C)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EKL(0x037D)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_EK	Present input deviation of PI3						

14.3.89 PI3_KI (0x037E, 0x037F)

PI3_KIH(0x037E)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KIL(0x037F)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_KI	Integral coefficient of PI3						

14.3.90 PI3_UKH (0x0380, 0x0381)

PI3_UKHH(0x0380)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKHL(0x0381)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_UKH	16 high-order bits of PI3 output						

14.3.91 PI3_UKL (0x0382, 0x0383)

PI3_UKLH(0x0382)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKLL(0x0383)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_UKL	16 low-order bits of PI3 output						

14.3.92 PI3_UKMAX (0x0384, 0x0385)

PI3_UKMAXH(0x0384)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKMAXL(0x0385)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_UKMAX	Maximum output of PI3						

14.3.93 PI3_UKMIN (0x0386, 0x0387)

PI3_UKMINH(0x0386)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKMINL(0x0387)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	PI3_UKMIN	Minimum output of PI3						

15 FOC/SVPWM

15.1 FOC/SVPWM Overview

15.1.1 FOC Introduction

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

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

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

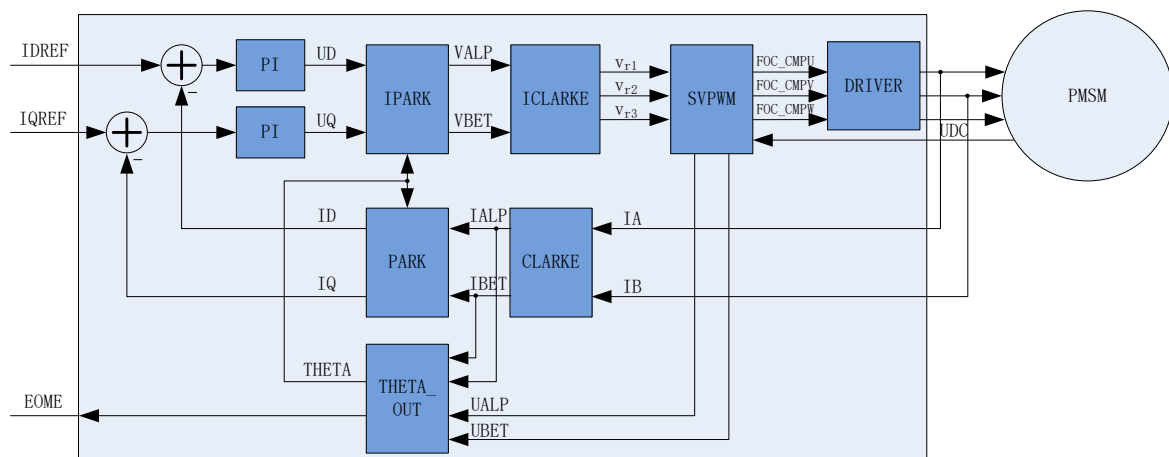


Figure 15-1 FOC Block Diagram

15.1.2 Reference Voltage (VREF) Input

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

15.1.3 PI Controller

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

1. Rotor flux control: PI controller of axis D, reference current FOC_IDREF minus feedback current FOC_ID as deviation input, proportional coefficient FOC_DKP and integral coefficient FOC_DKI to adjust the performance of PI controller, FOC_DMAX and FOC_DMIN to limit the output, and finally output D axis voltage FOC_UD.
2. Torque control: PI controller of q-axis current, with current reference FOC_IQREF minus feedback current FOC_IQ as deviation input, proportional coefficient FOC_DQKP and the integral coefficient FOC_DQKI for adjustment of PI performance, and FOC_QMAX and FOC_QMIN for limiting of the output amplitude. The output is voltage reference of q-axis FOC_UQ.
3. Angle estimation: PI controller of the estimator, proportional coefficient FOC_EKP and integral coefficient FOC_EKI adjust the performance of PI controller, and finally output estimated angle FOC_ETHETA.
4. PLL estimation: PI controller of PLL estimator, proportional coefficient FOC_PLLKP and integral coefficient FOC_PLLKI adjust the performance of PI controller, and finally output estimated BEMF FOC_EALP and FOC_EBET.

15.1.4 Coordinate Transformation

15.1.4.1 Inverse Park Transformation

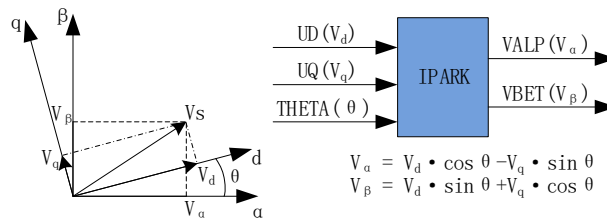


Figure 15-2 Park Transformation

Inverse Park transformation is used to transform two voltage vectors obtained by PI controller, FOC_UD and FOC_UQ, from dq-axis coordinate to $\alpha\beta$ -axis coordinate.

15.1.4.2 Inverse Clarke Transformation

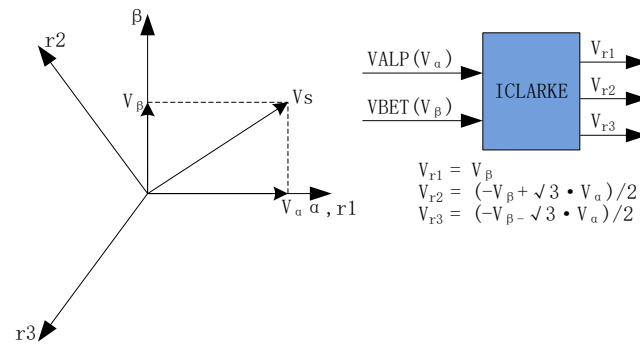


Figure 15-3 Inverse Clarke Transformation

Inverse Clarke transformation is used to transform voltage vector from $\alpha\beta$ -axis coordinate to 3-phase stationary coordinate.

15.1.4.3 Clarke Transformation

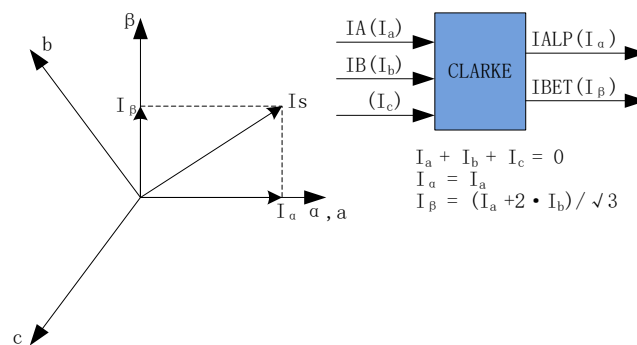


Figure 15-4 Clarke Transform

Clarke transformation is used to transform the sampled current from 3-phase stationary coordinate to $\alpha\beta$ -axis coordinate.

15.1.4.4 Park Transformation

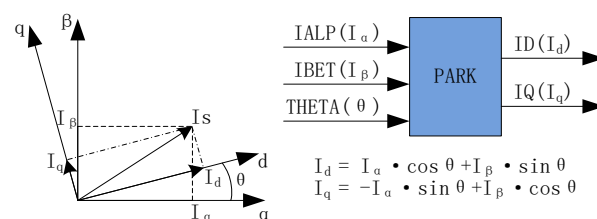


Figure 15-5 Park Transform

Park transformation is used to transform the current vectors, obtained after Clarke transformation, from $\alpha\beta$ -axis coordinate to dq-axis coordinate to get the sampled dq-axis current.

15.1.5 SVPWM

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

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

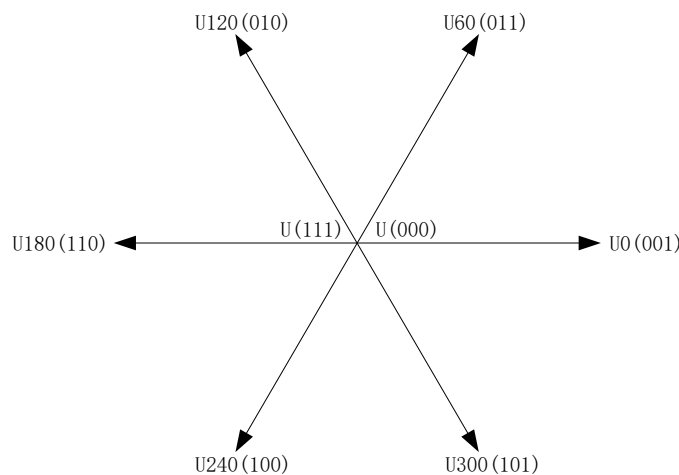


Figure 15-6 SVPWM Voltage Vector

SVPWM uses the sum of two adjacent vectors to generate any voltage vector located in the voltage vector space. U_{OUT} is the desired vector and it is in the sector between U_{60} and U_0 . In defined PWM cycle (T), the effect, U_0 applied $2T_1/T$ time and U_{60} applied $2T_2/T$ time, is equivalent to the U_{OUT} . The rest of time (T_0) is applied by zero voltage vector.

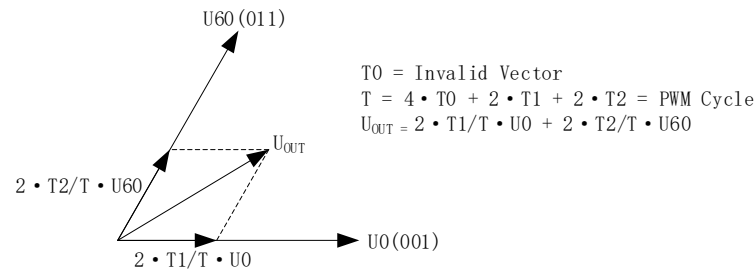


Figure 15-7 SVPWM Voltage Vector Synthesis

Table 15-1 States of SVPWM Inverter

Phase C	Phase B	Phase A	V _{ab}	V _{bc}	V _{ca}	V _{ds}	V _{qs}	Vector
0	0	0	0	0	0	0	0	U(000)
0	0	1	V _{DC}	0	-V _{DC}	2/3V _{DC}	0	U ₀
0	1	1	0	V _{DC}	-V _{DC}	1/3V _{DC}	1/3V _{DC}	U ₆₀
0	1	0	-V _{DC}	V _{DC}	0	-1/3V _{DC}	1/3V _{DC}	U ₁₂₀
1	1	0	-V _{DC}	0	V _{DC}	-2/3V _{DC}	0	U ₁₈₀
1	0	0	0	-V _{DC}	V _{DC}	-1/3V _{DC}	-1/3V _{DC}	U ₂₄₀
1	0	1	V _{DC}	-V _{DC}	0	1/3V _{DC}	-1/3V _{DC}	U ₃₀₀
1	1	1	0	0	0	0	0	U(111)

15.1.5.1 Continuous SVPWM

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

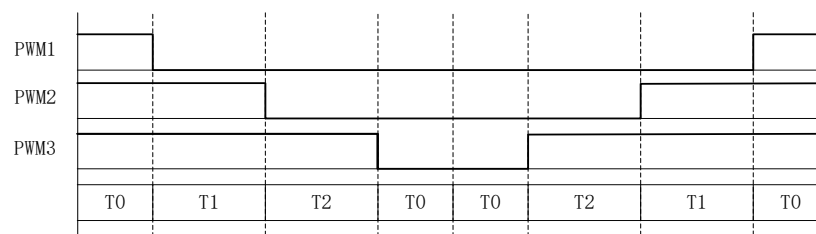


Figure 15-8 Output Level of Continuous SVPWM

15.1.5.2 Discontinuous SVPWM

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

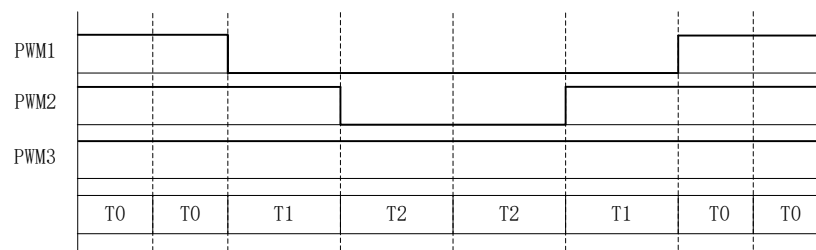


Figure 15-9 Output Level of Discontinuous SVPWM

15.1.6 Overmodulation

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

15.1.7 Deadtime Compensation

Deadtime compensation is available in dual/triple-shunt current sampling mode. The compensation value of deadtime is configured by FOC_TSMIN. This mode improves sinusoidal waveform of the current.

15.1.8 Current and Voltage Sampling

In FOC mode, bus voltage and phase current are sampled by hardware automatically. Before FOC module operates, ADC and operational amplifier shall be enabled and the corresponding control registers be configured. No configuration is required for ADC channel and scan mode. Single/dual/triple-shunt current sampling mode is selected by setting FOC_CR1[CSM]. In single-shunt current sampling mode, ADC channel 4 is the default sampling channel of the bus current (itrip). In dual-shunt current sampling mode, ADC channel 0 and channel 1 are the default sampling channels of A-phase current (ia) and B-phase current (ib) respectively. In triple-shunt current sampling mode, ADC channel 0, channel 1 and channel 4 are the default sampling channels of ia, ib and C-phase current (ic) respectively. Channel 2 can be selected for bus voltage sampling.

15.1.8.1 Single-shunt Current Sampling Mode

FOC_CR1[CSM] is set to “0” to select the single-shunt current sampling mode. In this mode, FOC module samples itrip (channel 4) twice during the timer counting-up operation, and samples bus voltage during the timer counting-down operation and after FOC module completes the calculation.

Since deadtime affects the accuracy of current sampling, FOC module samples within T1' and T2', which is the applied time of active voltage vector with deadtime removed. FOC_TRGDLY is the register which advances or delays the current sampling time, and this register shall be configured reasonably to ensure sampling is completed within T1' and T2'. For example, when MCU clock rate = 24MHz(41.67ns) and FOC_TRGDLY = 5, the sampling time is delayed by $41.67 \times 5 = 208\text{ns}$; and when FOC_TRGDLY = 0xFB(-5), the sampling time is advanced by 208ns.

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

15.1.8.2 Dual/Triple-shunt Current Sampling Mode

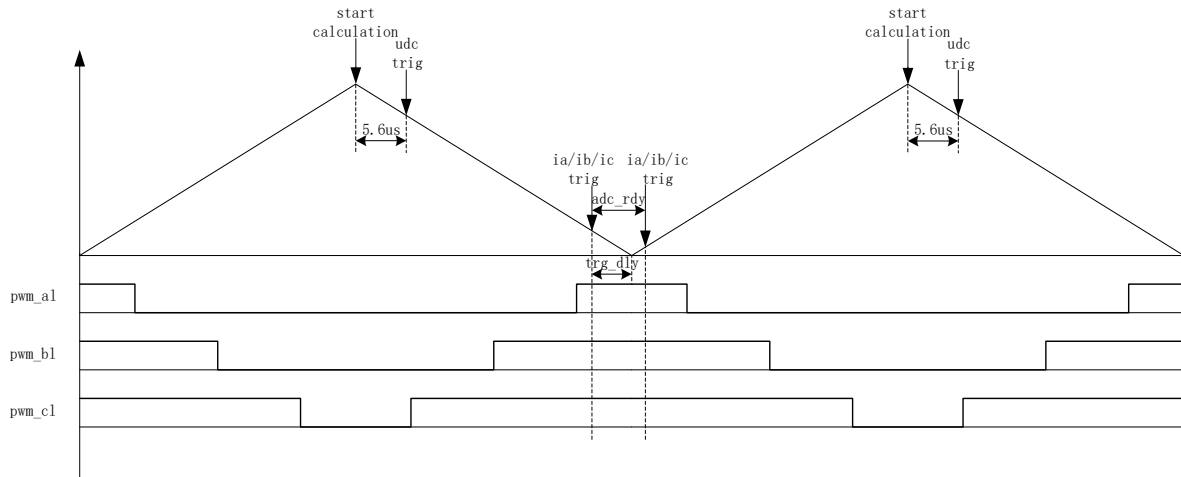


Figure 15-10 Dual/Triple-shunt Sequential Current Sampling Mode

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

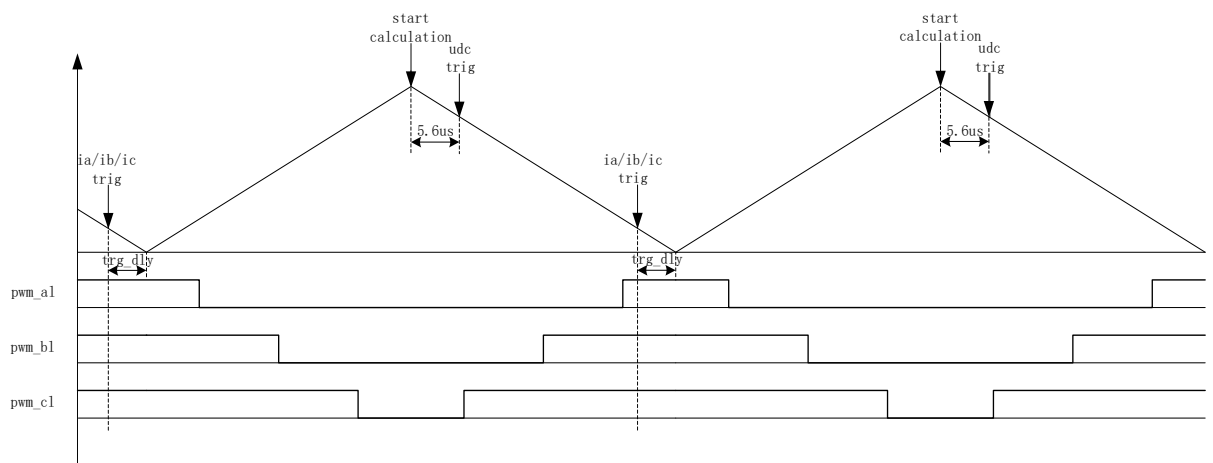


Figure 15-11 Dual/Triple-shunt Alternate Current Sampling Mode

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

carrier cycle samples one phase of the ia/ib/ic, and the second carrier cycle samples the current of the other phase, so as to alternately sample the current of two phases in three phases. FOC_TRGDLY is used to configure the sampling time of ia (channel 0), ib (channel 1) and ic (channel 4). FOC_TRGDLY shall be configured reasonably to ensure sampling time for the ia/ib/ic current is within zero voltage vector (i.e. pwm_al,pwm_bl,pwm_cl = 111). For example, when MCU clock rate = 24MHz(41.67ns), FOC_TRGDLY = 0xB2 and FOC timer counts-down, phase current is sampled at $41.67 \times 50 = 2.08\mu\text{s}$ before an underflow event.

15.1.8.3 Current Sampling Offset

The current sampling offset voltage shall be added to sample full range of current due to the existence of the positive and negative phase current. When phase current is 0, ADC result is the offset value. ADC result minus this value, 0x4000 default, is the sampling current. Since ADC reference voltage and hardware are nonideal, there is a deviation between the default value and the real value. Therefore, it is necessary to calibrate the offset. The calibration procedure is as follows. When FOC module does not work and there is no current in three phases, MCU starts to sample the corresponding channel several times, averages all the sampled value, and writes the averaged value to FOC_CSO. Providing ADC sampling range is 0 ~ 5V and the offset is 2.5V, $\text{FOC_CSO} = 2.5\text{V}/5\text{V} \times 32768 = 16384$ (0x4000).

- When FOC_CR2[CSOC] = 00/11, FOC_CSO is written to modify the offset of itrip and ic.
- When FOC_CR2[CSOC] = 01, FOC_CSO is written to modify the offset of ia.
- When FOC_CR2[CSOC] = 10, FOC_CSO is written to modify the offset of ib.

15.1.9 Angle Mode

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

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

Table 15-2 Sources of Angle

RFAE	ANGM	EFAE	Source
1	x	x	Forced ramping angle
0	0	x	Forced pulling angle
0	1	0	Estimated angle of estimator
0	1	1	$\omega > \text{FOC_EFREQMIN}$: Estimated angle of estimator $\omega < \text{FOC_EFREQMIN}$: Forced angle of estimator

15.1.9.1 Forced Ramping Angle

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

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

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

Forced ramping angle has the highest priority. Configuring FOC_CR1[RFAE] to “1” enables the ramping feature. Ramping module makes a ramping operation in every PWM cycle and the counter is added by 1. When the value of the counter reaches the set value by FOC__RTHECNT, FOC_CR1[RFAE] is cleared by hardware, and then the ramping is completed. Thereafter, according to the value of FOC_CR1[ANGM], the angle comes from estimator (FOC_CR1[ANGM] = 1) or forced pulling angle (FOC_CR1[ANGM] = 0).

15.1.9.2 Forced Pulling Angle

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

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

There are two scenarios for forced pulling angle.

- When FOC_CR1[RFAE] is set to “1” and FOC_CR1[ANGM] to “0”, MCU switches to forced pulling angle mode after forced ramping angle mode. The speed is the cumulative result after ramp force angle mode. This mode implements a forced uniform speed control.
- When FOC_CR1[RFAE] is set to “0” and FOC_CR1[ANGM] to “0”, the angle is the forced pulling angle and the speed FOC__RTHESTEP is the initial speed written by software. Configuring FOC__RTHESTEP = 0 enables the pre-position feature. Configuring FOC__RTHESTEP != 0 implements Hall-based FOC (Principle of Hall-based FOC: MCU calculates the angle and speed based on the received Hall signals, and writes them to FOC__THETA and FOC__RTHESTEP for calibration).

15.1.9.3 Estimator Output Angle

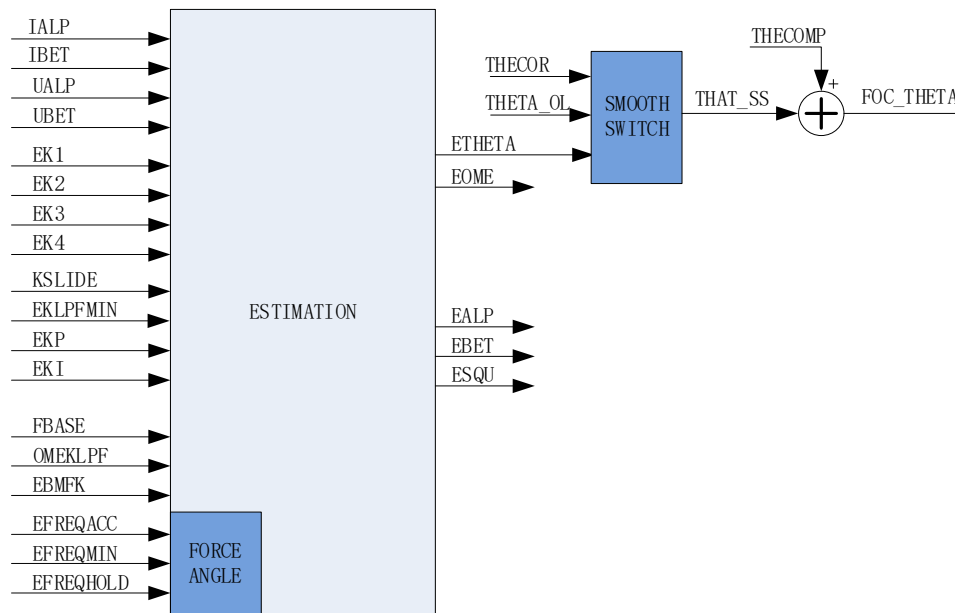


Figure 15-12 Schematic Block Diagram of Estimator

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

1. Estimated Angle of Estimator

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

2. Forced Angle of Estimator

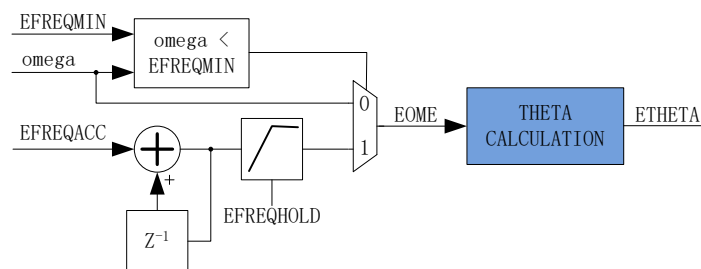


Figure 15-13 Schematic Diagram of Forced Angle of Estimator

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

When FOC_CR1[EFAE] = 1, if $\omega < \text{FOC_EFREQMIN}$, the forced speed is selected as EOME.

The forced speed starts with 0 and increases by FOC_EFREQACC in each PWM cycle, with the maximum value FOC_EFREQHOLD. When $\omega \geq \text{FOC_EFREQMIN}$, ω is selected as EOME.

3. Angle Smooth Switching

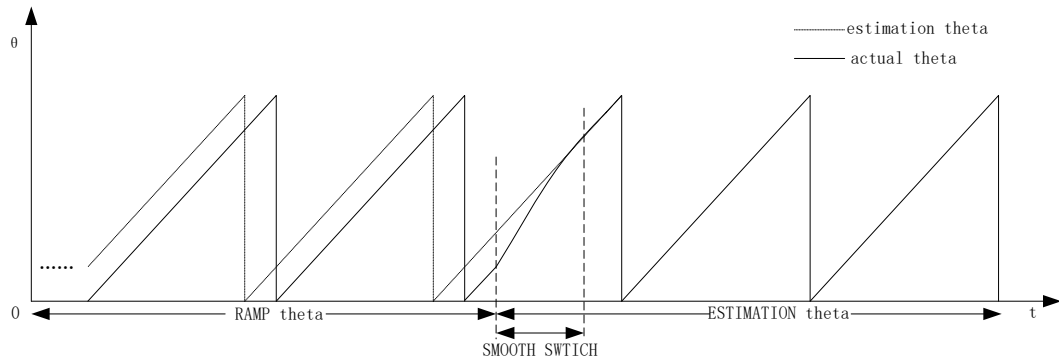


Figure 15-14 Angle Smooth Switching Curve

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

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

4. Angle Compensation

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

15.1.10 Motor Real-time Parameters

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

1. Used angle FOC__THETA
2. Estimated angle FOC__ETHETA, Estimated speed FOC__EOME
3. d-axis voltage FOC__UD, q-axis voltage FOC__UQ
4. d-axis current FOC__ID, q-axis current FOC__IQ
5. α -axis voltage FOC__VALP, β -axis voltage FOC__VBET
6. Bus voltage FOC__UDCFLT

7. Phase current FOC__IA, FOC__IB, FOC__IC
8. α -axis current (equal to FOC__IA), β -axis current FOC__IBET
9. α -axis BEMF FOC__EALP, β -axis BEMF FOC__EBET
10. Square of BEMF FOC__ESQU
11. Motor power FOC__POW

15.1.10.1 Tailwind/headwind Detection

FOC module provides tailwind/headwind detection feature. FOC module starts to operate when FOC_CR3[ESCMS] is set to “1”, FOC_IDREF to “0” and FOC_IQREF to “0”. Motor’s rotor state is detected by FOC__ETHETA and FOC__EOME. If FOC__ETHETA decreases or FOC__EOME is a negative value, the motor rotates in the headwind state and it is necessary to brake first and then start the motor with ramping forced angle mode. If FOC__ETHETA increases or FOC__EOME is a positive value, the motor rotates in the tailwind state and can be started using estimated angle directly.

15.1.10.2 BEMF Detection

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

15.1.10.3 Motor Power

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

15.1.11 FG Generation

FG signal is generated by FOC module and Timer4. FOC module calculates an FG result based on frequency base fbase, low-pass filtered speed FOC__EOMELPF and FG coefficient FOC__KFG in every PWM cycle. The result is updated to TIM4__ARR automatically and half of the result (TIM4__ARR/2) to TIM4__DR by hardware. It shall be noted that Timer4 must work in output mode and the clock division factor of Timer4 shall be configured according to the motor maximum speed.

FOC__KFG is computed by the following algorithm:

$$\text{FOC_KFG} = 24\text{MHz} / (2^{\text{T4PSC}} * \text{FBASE} * x)$$

If the result exceeds 65535, the clock division factor TIM4_CR0[T4PSC] shall be adjusted.

If FOC__KFG=0, this feature is disabled and TIM4__ARR/DR will not be updated automatically.

15.2 FOC Registers

15.2.1 FOC_CR1 (0x40A0)

Bit	7	6	5	4	3	2	1	0
Name	OVMDL	EFAE	RFAE	ANGM	CSM		RSV	SVPWMEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
Reset	0	0	0	0	0	0	-	0
Bit	Name	Description						
[7]	OVMDL	Overmodulation Enable 0: Disable 1: Enable						
[6]	EFAE	Forced Angle of Estimator Enable When this feature is enabled, angle mode is determined by the estimator, and it switches to estimated angle mode automatically. 0: Disable 1: Enable						
[5]	RFAE	Forced Ramping Angle Enable When this feature is enabled, angle mode is determined by the ramping module. After ramping, it switches to estimated mode or forced pulling mode according to FOC_CR1[ANGM]. FOC_CR1[RFAE] is cleared to "0" by hardware as well. 0: Disable 1: Enable						
[4]	ANGM	Angle Mode When FOC_CR1[RFAE] = 0, angle mode is determined by this bit. When FOC_CR1[RFAE] = 1, angle mode is determined by this bit after ramping. 0: Forced Pulling Angle Mode 1: Estimated Angle of Estimator Mode						
[3:2]	CSM	Current Sampling Mode x0: Single-shunt Current Sampling 01: Dual-shunt Current Sampling 11: Triple-shunt Current Sampling						
[1]	RSV	Reserved						
[0]	SVPWMEN	SVPWM Module Enable 0: Disable 1: Enable						

15.2.2 FOC_CR2 (0x40A1)

Bit	7	6	5	4	3	2	1	0
Name	ESEL	RSV	F5SEG	DSS	CSOC		UQD	UDD
Type	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	0
Bit	Name	Description						
[7]	ESEL	Angle Estimator Mode Selection 0: SMO 1: PLL (phase-locked loop). FOC_KSILDE register is FOC_PLLKP of PI controller, and FOC_KLPFMIN register is FOC_PLLKI of PI controller.						
[6]	RSV	Reserved						
[5]	F5SEG	SVPWM Mode Selection in Dual/Triple-shunt Current Sampling Mode 0: Continuous SVPWM 1: Discontinuous SVPWM (cannot be selected in single-shunt current sampling mode)						
[4]	DSS	Dual/Triple-shunt Current Sampling Mode 0: Sequential Sampling Mode, where the current of two phases are sampled in each						

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

15.2.3 FOC_TSMIN (0x40A2)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TSMIN							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	FOC_TSMIN	Single-shunt Sampling Mode: minimum window for sampling Dual/triple-shunt Sampling Mode: deadtime compensation Range (0,255)						
TSMIN = sampling window T_{window} + deadtime T_{DT} Example: Assuming that $T_{window} = 1\mu s$, $T_{DT} = 1\mu s$, TSMIN = $2\mu s$ and carrier period = $62.5\mu s$, then $FOC_TSMIN = (1 + 1)/62.5*4096 = 131$.								

15.2.4 FOC_TGLI (0x40A3)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TGLI							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	FOC_TGLI	Narrow Pulse Elimination for High Side of the Bridge This feature is designed for high-voltage applications. The high side of bridge must be longer than a certain time. After this bit is configured, high side of the bridge is not turned on when the conducting time is less than this value. Range (0,255)						
Example: Assuming that it is required to remove narrow pulses with less than 1μs width, deadtime T _{DT} = 1μs, and carrier period = 62.5μs, then FOC_TGLI = (1 + 1)/62.5*4096 =131.								

15.2.5 FOC_TBLO (0x40A4)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TBLO							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	FOC_TBLO	Sampling Masking Time in Triple-shunt Current Sampling Mode If low side of the bridge is turned on for less than FOC_TBLO, the current of this phase is not sampled and obtained through special process. Range (0,255)						
Example: Assuming that the phase current is not sampled if the low side is turned on for less than 1 μs, then FOC_TBLO = 1000ns/41.67ns = 24.								

15.2.6 FOC_TRGDLY (0x40A5)

Bit	7	6	5	4	3	2	1	0
Name	FOC_TRGDLY							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	FOC_TRGDLY	Single-shunt Current Sampling Mode: Trigger Delay for ADC Current Sampling Range (-128,127) Dual/Triple-shunt Current Sampling Mode: Time for Current Sampling TRGDLY[7] decides the sampling occurs in falling or rising interval of the timer: TRGDLY[7] = 0: rising interval; TRGDLY[7] = 1: falling interval Range (0, DRV_ARR[6:0]) Single-shunt Current Sampling Mode: If FOC_TRGDLY = 5, it delays by 5*T = 208ns to sample the current, and if FOC_TRGDLY = 0xFB (complement) or FOC_TRGDLY = -5, it advances by 5*T=208ns. Dual-shunt/Triple-shunt Current Sampling Mode: If FOC_TRGDLY = 0x85 (the highest bit, and the remaining 7 bits are absolute values) and Driver timer counts down, it samples the current at 5*T = 208ns before an overflow event occurs. If FOC_TRGDLY = 5 and Driver timer counts up, it samples the current at 5*T = 208ns after an overflow event occurs.						

15.2.7 FOC_CSO (0x40A6, 0x40A7)

FOC_CSOH(0x40A6)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_CSO [15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0
FOC_CSOL(0x40A7)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_CSO[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_CSO	Current Sampling Offset FOC_CR2[CSOC] is configured to select the current, and FOC_CSO is written to						

		calibrate current sampling offset of itrip in single-shunt current sampling mode, ia, ib in dual-shunt current sampling mode and ia, ib and ic in triple-shunt current sampling mode. Range (0,32767), the MSB is always 0
Example: Assuming that the ADC voltage falls within 0V ~ 5V with a reference value of 2.5V, then FOC_CS0 = $2.5V/5V \times 32768 = 16384(0x4000)$		

15.2.8 FOC_RTSTEP (0x40A8, 0x40A9)

FOC__RTHESTEP(0x40A8)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__RTHESTEP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__RTHESTEPL(0x40A9)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__RTHESTEP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC__RTHESTEP		Speed of Ramping Module; The format is the same as that of FOC__THETA Write: Initial speed Read: Current speed Range (-32768,32767) Note: FOC__RTHESTEP is an internal 32-bit variable. MSB is sign bit. High-order 16 bits are written by software.					
FOC__RTHESTEP (32 bits) = FOC__RTHESTEP (32 bits) + FOC__RTHEACC (32 bits, 16 high-order bits are always 0 and 16 low-order bits are configurable) THETA (16 bits) = THETA (16 bits) + FOC__RTHESTEP (16 high-order bits)								

15.2.9 FOC_RTSTEP (0x40AA, 0x40AB)

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

15.2.10 FOC_RTHERCNT (0x40AC)

Bit	7	6	5	4	3	2	1	0
Name	FOC_RTHERCNT							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	FOC_RTHERCNT		Max. ramping counts = FOC_RTHERCNT*256 When ramping feature is enabled (FOC_CR1[RFAE] = 1), the ramping angle increases in each PWM cycle. After FOC_RTHERCNT*256 times, ramping feature is disabled. Range (0,255)					

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

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

15.2.12 FOC_THECOMP (0x40AE, 0x40AF)

FOC_THECOMP(0x40AE)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_THECOMP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_THECOMP(0x40AF)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_THECOMP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_THECOMP		Angle Compensation Value The output angle FOC_THETA is derived from the estimator estimated angle + compensation value; the format is same with that of FOC_THETA. Range (-32768,32767)					

15.2.13 FOC_DMAX (0x40B0, 0x40B1)

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

Name	FOC_DMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_DMAX	Max. output UD of d-axis PI controller Range (-32768,32767)						

15.2.14 FOC_DMIN (0x40B2, 0x40B3)

FOC_DMINH(0x40B2)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_DMIN [15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DMINL(0x40B3)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_DMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_DMIN	Min. output UD of d-axis PI controller Range (-32768,32767)						

15.2.15 FOC_QMAX (0x40B4, 0x40B5)

FOC_QMAXH(0x40B4)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_QMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_QMAXL(0x40B5)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_QMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_QMAX	Max. output UQ of q-axis PI controller Range (-32768,32767)						

15.2.16 FOC_QMIN (0x40B6, 0x40B7)

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

Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_QMIN	Min. output UQ of q-axis PI controller; Range (-32768,32767)						

15.2.17 FOC__UD (0x40B8, 0x40B9)

FOC__UDH(0x40B8)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UD[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__UDL(0x40B9)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__UD	d-axis voltage calculated by d-axis PI controller Range (-32768,32767)						

15.2.18 FOC__UQ (0x40BA, 0x40BB)

FOC__UQH(0x40BA)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__UQL(0x40BB)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__UQ	q-axis voltage calculated by q-axis PI controller Range (-32768,32767)						

15.2.19 FOC__ID (0x40BC, 0x40BD)

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

Bit	Name	Description
[15:0]	FOC__ID	ID by Park Transformation Range (-32768,32767)

15.2.20 FOC__IQ (0x40BE, 0x40BF)

FOC__IQH(0x40BE)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IQ[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IQL(0x40BF)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IQ[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__IQ	IQ by Park Transformation Range (-32768,32767)						

15.2.21 FOC__IBET (0x40C0, 0x40C1)

FOC__IBETH(0x40C0)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IBETL(0x40C1)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IBET[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__IBET	Current IBETA Range (-32768,32767)						

15.2.22 FOC__VBET (0x40C2, 0x40C3)

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

[15:0]	FOC__VBET	VBETA by Inverse Park Transformation Range (-32768,32767)
--------	-----------	--

15.2.23 FOC__VALP (0x40C4, 0x40C5)

FOC__VALPH(0x40C4)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__VALP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__VALPL(0x40C5)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__VALP[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__VALP	VALPHA by Inverse Park Transformation Range (-32768,32767)						

15.2.24 FOC__IC (0x40C6, 0x40C7)

FOC__ICH(0x40C6)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IC[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__ICL(0x40C7)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IC[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__IC	Sampled Phase-C Current Range (-32768,32767)						

15.2.25 FOC__IB (0x40C8, 0x40C9)

FOC__IBH(0x40C8)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IB[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IBL(0x40C9)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IB[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__IB	Sampled Phase-B Current Range (-32768,32767)						

15.2.26 FOC__IA (0x40CA, 0x40CB)

FOC__IAH(0x40CA)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IA[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IAL(0x40CB)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IA[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__IA	Sampled Phase-A Current; Range (-32768,32767)						

15.2.27 FOC__THETA (0x40CC, 0x40CD)

FOC__THETAH(0x40CC)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__THETA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__THETAL(0x40CD)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__THETA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__THETA	Output Angle Write: Forced Pulling Angle Read: Current Angle by FOC Module Range (-32768,32767)						
The bit value -32768 ~ 32767 corresponds to angle range -180°~ 180°. Example: Assuming that FOC__THETA = 8192, the output angle is 8192/32768*180°= 45°.								

15.2.28 FOC__ETHETA (0x40CE, 0x40CF)

FOC__ETHETAH(0x40CE)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__ETHETA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__ETHETAL(0x40CF)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__ETHETA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__ETHETA	Read: Output Angle of Estimator (angle before FOC__THECOMP is applied); the format is same as that of FOC__THETA.						

15.2.29 FOC__EALP (0x40D0, 0x40D1)

FOC__EALPH(0x40D0)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__EALP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__EALPL(0x40D1)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__EALP[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__EALP	Estimated EALPHA by estimator Range (-32768,32767)						

15.2.30 FOC__EBET (0x40D2, 0x40D3)

FOC__EBETH(0x40D2)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__EBET[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__EBETL(0x40D3)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__EBET[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__EBET	Estimated EBETA by estimator Range (-32768,32767)						

15.2.31 FOC__EOME (0x40D4, 0x40D5)

FOC__EOMEH(0x40D4)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__EOME[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__EOMEL(0x40D5)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__EOME[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__EOME	Output Speed OMEGA of Estimator Range (-32768,32767)						

15.2.32 FOC__ESQU (0x40D6, 0x40D7)

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

15.2.33 FOC__POW (0x40D8, 0x40D9)

FOC__POWH(0x40D8)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__POW[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__POWL(0x40D9)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__POW[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__POW	Motor Power Range (-32768,32767); An error occurs if this value is negative.						

15.2.34 FOC__EKP (0x4074, 0x4075) (Shared with BLDC Motors)

FOC__EKPH(0x4074)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__EKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__EKPL(0x4075)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__EKP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__EKP	KP of PI controller used by the estimator. Range (0,32767). MSB is always 0. Q12 format.						

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

FOC_EKIH(0x4076)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EKI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EKIL(0x4077)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EKI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_EKI	KI of PI controller used by the estimator Range (0,32767); MSB is always 0; Q15 format.						

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

FOC_EBMFKH(0x407C)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EBMFK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EBMFKL(0x407D)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EBMFK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_EBMFK	Coefficient used to calculate BEMF low-pass filter coefficient EKLPF Range (0,32767); Q15 format.						
EKLPF = EBMFK * OMEGA								
EBMFK = 2 * pi * fbase*△T								

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

FOC_KSLIDEH(0x4078)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_KSLIDE / FOC_PLLKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_KSLIDEL(0x4079)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_KSLIDE / FOC_PLLKP [7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_KSLIDE / FOC_PLLKP	FOC_CR2[ESEL] = 0: KSLIDE coefficient; Q15 format FOC_CR2[ESEL] = 1: KP of PI controller on PLL; Q12 format Range (0,32767); MSB is always 0.						

15.2.38 FOC_EKLPMIN (0x407A, 0x407B) (Shared with BLDC Motors)

FOC_EKLPMINH(0x407A)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EKLPMIN / FOC_PLLKPI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EKLPMINL(0x407B)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EKLPMIN / FOC_PLLKPI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_EKLPMIN / FOC_PLLKPI		FOC_CR2[ESEL] = 0: The minimum value of BEMF low pass filter factor. EKLPF is forced to be this value when it is lower than this value. Q15 format. FOC_CR2[ESEL] = 1: PI controller KI coefficient on PLL. Q15 format. Range (0,32767); MSB is always 0.					

15.2.39 FOC_OMEKLPF (0x407E, 0x407F)

FOC_OMEKLPFH(0x407E)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_OMEKLPF[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_OMEKLPFL(0x407F)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_OMEKLPF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_OMEKLPF		LPF factor of estimated speed of the estimator. Range (0,32767); MSB is always 0. Q15 format.					

15.2.40 FOC_FBASE (0x4080, 0x4081)

FOC_FBASEH(0x4080)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_FBASE[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_FBASEL(0x4081)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_FBASE[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_FBASE	Coefficient of angular increment DELTA THETA that is calculated based on OMEGA						
FBASE = fbase * ΔT * 32768								
Assuming that fbase = 200Hz and ΔT = 62.5us, then FBASE = 409								

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

FOC_EFREQACCH(0x4082)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EFREQACC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EFREQACCL(0x4083)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EFREQACC[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_EFREQACC		Speed Increment of the Forced Angle Mode Range (0,65535) FOC_EFREQACC is an internal 24-bit variable and MSB is sign bit. Low-order 16 bits are written by software.					
Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. If speed increment = 3rpm, then FOC_EFREQACC = 3rpm/speed_base*32768*256 = 8388(0x20C4).								

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

FOC_EFREQMINH(0x4084)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EFREQMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EFREQMINL(0x4085)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EFREQMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_EFREQMIN		Minimum value of Omega: With Forced Angle of Estimator Mode enabled, FOC module outputs forced angle when the estimated angle is smaller than the bit value. Range (-32768,32767) Note: FOC_EFREQMIN is an internal 24-bit variable, and MSB is sign bit. High-order 16 bits are written by software.					
Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. Assuming that the minimum Omega = 30rpm, then FOC_EFREQMIN = 30rpm/speed_base*32768 = 327(0x147).								

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

FOC_EFREQHOLDH(0x4086)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EFREQHOLD[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EFREQHOLDL(0x4087)								

Bit	7	6	5	4	3	2	1	0
Name	FOC_EFREQHOLD[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_EFREQHOLD		Hold value of Omega: When Omega rises to this value, it is kept unchanged. Range (-32768,32768) FOC_EFREQHOLD is an internal 24-bit variable, and MSB is sign bit. High-order 16 bits are written by the software.					
Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. If hold value of Omega = 60rpm, then FOC_EFREQHOLD = 60rpm/speed_base*32768 = 655(0x028F).								

15.2.44 FOC_EK3 (0x4088, 0x4089)

FOC_EK3H(0x4088)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EK3[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK3L(0x4089)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EK3[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_EK3		The 3 rd coefficient of the current model in estimator; Range (0, 32767) and MSB is always 0. Q15 format					

15.2.45 FOC_EK4 0x408A, 0x408B)

FOC_EK4H(0x408A)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EK4[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK4L(0x408B)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EK4[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC_EK4		The 4 th coefficient of the current model in estimator.					

15.2.46 FOC_EK1 (0x408C, 0x408D)

FOC_EK1H(0x408C)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EK1[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK1L(0x408D)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_EK1	The 1 st coefficient of the current model in estimator; Range (0,32767) and MSB is always 0. Q15 format.						

15.2.47 FOC_EK2 (0x408E, 0x408F)

FOC_EK2H(0x408E)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EK2[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK2L(0x408F)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EK2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_EK2	The 2 nd coefficient of the current model in estimator; Range (0,32767) and MSB is always 0. Q15 format.						

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

FOC_IDREFH(0x4090)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_IDREF[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_IDREFL(0x4091)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_IDREF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_IDREF	User-defined Current ID Range (-32768,32767)						

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

FOC_IQREFH(0x4092)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_IQREF[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_IQREFL(0x4093)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_IQREF[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_IQREF	User-defined Current IQ Range (-32768,32767)						

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

FOC_DQKPH(0x4094)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_DQKP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DQKPL(0x4095)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_DQKP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_DQKP	KP of dq-axis PI Controller Range (0,32767); MSB is always 0. Q12 format.						

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

FOC_DQKIH(0x4096)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_DQKI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DQKIL(0x4097)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_DQKI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_DQKI	KI of dq-axis PI Controller Range (0,32767) ; MSB is always 0. Q15 format.						

15.2.52 FOC__UDCFLT (0x4098, 0x4099)

FOC__UDCFLTH(0x4098)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UDCFLT[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__UDCFLTL(0x4099)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UDCFLT[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__UDCFLT	Filtered Bus voltage FOC module samples the bus voltage and filters it for software use. The default channel is ADC channel 2. Range (0,32767)						
Example: The bus voltage is scaled down by 1/6 before feeding into the ADC module, ADC VREF = 5V (namely, the sampling range is [0V ~ 30V]) and FOC__UDCFLT = 19661(0x4CCD), then bus voltage = 19661/32768*5V*6 = 18V.								

15.2.53 FOC_CR3 (0x40ee)

FOC_CR3(0x40ee)								
Bit	7	6	5	4	3	2	1	0
Name	ICLR	RSV	MFP_EN	FOCUST_A	FOCFEN	ESCMS	TSMINH_9	TSMINH_8
Type	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	0
Bit	Name	Description						
[7]	ICLR	Clear FOC__IA/B/CMAX to "0" A write of "1" clears FOC__IA/B/CMAX to "0". The bit is automatically set to "0" after FOC__IA/B/CMAX are cleared to "0". 0: Disable 1: Enable						
[6]	RSV	Reserved						
[5]	MFP_EN	Adaptive Observer Enable 0: Disable 1: Enable						
[4]	FOCUSTA	Single-shunt Off Sampling Enable 0: Disable 1: Enable						
[3]	FOCFEN	FOC Force Enable Bit When DRV_CR[MESEL] is set to "1", FOC module performs calculation even if DRV_CR[OCS] = 0. 0: Disable 1: Enable						
[2]	ESCMS	ATAN Estimation Enable 0: Disable 1: Enable						
[1:0]	TSMINH_9:8	Scale up by two bits of FOC__TSMIN						

15.2.54 FOC_DKP (0x409c, 0x409d)

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

15.2.55 FOC_DKI (0x409e, 0x409f)

FOC_DKIH(0x409e)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_DKI[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_DKIL(0x409f)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_DKI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_DKI	KI of d-axis PI Controller Range (0,32767) MSB is always 0. Q15 format						

15.2.56 FOC_IAMAX (0x40da, 0x40db)

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

Bit	Name	Description
[15:0]	FOC__IAMAX	Max. Phase-A Current This value may be unreliable unless the motor rotates in a full electrical period. It is also incorrect if the phase current is collected improperly. This maximum value will not be cleared to “0” automatically unless FOC_CR3[ICLR] is set to “1”. Range (0,32767)

15.2.57 FOC__IBMAX (0x40dc, 0x40dd)

FOC__IBMAXH(0x40DC)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__IBMAX[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__IBMAXL(0x40DD)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__IBMAX[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__IBMAX	Max. Phase-B Current This value may be unreliable unless the motor rotates in a full electrical period. It is also incorrect if the phase current is collected improperly. This maximum value will not be cleared to “0” automatically unless FOC_CR3[ICLR] is set to “1”. Range (0.32767)						

15.2.58 FOC__ICMAX (0x40de, 0x40df)

FOC__ICMAXH(0x40DE)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__ICMAX[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__ICMAXL(0x40DF)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__ICMAX[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__ICMAX	Max. Phase-C Current This value may be unreliable unless the motor rotates in a full electrical period. It is also incorrect if the phase current is collected improperly. This maximum value will not be cleared to “0” automatically unless FOC_CR3[ICLR] is set to “1”. Range (0.32767)						

15.2.59 FOC__EMF (0x40e0, 0x40e1)

FOC__EMFH(0x40E0)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__EMF [15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__EMFL(0x40E1)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__EMF [7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__EMF	Estimated BEMF of Estimator This value is the root of sum of square of FOC__EALP and square of FOC__EBET Range (0,32767)						

15.2.60 FOC__UDCPS (0x40e2,0x40e3)

FOC__UDCPSH(0x40e2)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UDCPS [15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__UDCPSL(0x40e3)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UDCPS [7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC__UDCPS	d-axis Voltage Compensation Value The result of d-axis PI controller (FOC__UD) added to FOC__UDCPS is transferred to the next module. Range (-32768,32767)						

15.2.61 FOC__UQCPS (0x40e4, 0x40e5)

FOC__UQCPSH(0x40e4)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UQCPS [15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC__UQCPSL(0x40e5)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UQCPS [7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						

[15:0]	FOC__UQCPS	q-axis Voltage Compensation Value The result of q-axis PI controller (FOC__UD) added to FOC_UQCPS is transferred to the next module. Range (-32768,32767)
--------	------------	---

15.2.62 FOC__UQEX (0x40e6, 0x40e7)

FOC__UQEXH(0x40e6)								
Bit	15	14	13	12	11	10	9	8
Name	FOC__UQEX [15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC__UQEXL(0x40e7)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__UQEX[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	FOC__UQEX		q-axis Overflow Value for Flux-weakening Control Equation: FOC_UQ - FOC_QMAX FOC__UQEX is positive when FOC_UQ > FOC_QMAX; FOC__UQEX is negative when FOC_UQ < FOC_QMAX; FOC__UQEX can be used to realize flux-weakening control. Range (-32768, 32768)					

15.2.63 FOC_ID_LPFK (0x40e8)

Bit	7	6	5	4	3	2	1	0
Name	FOC_ID_LPFK [7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	Name		Description					
[7:0]	FOC_ID_LPFK		LPF coefficient of FOC_ID Range 0 ~ 255					

15.2.64 FOC_IQ_LPFK (0x40e9)

Bit	7	6	5	4	3	2	1	0
Name	FOC_IQ_LPFK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	Name		Description					
[7:0]	FOC_IQ_LPFK		LPF coefficient of FOC_IQ Range 0 ~ 255					

15.2.65 FOC_KFG (0x40ea, 0x40eb)

FOC_KFGH(0x40ea)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_KFG [15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_KFGL(0x40eb)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_KFG [7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	FOC_KFG	Coefficient of FG Calculation FOC module performs the calculation based on FOC_EOMELPF and FOC_KFG in each PWM cycle. The result is updated to TIM4__ARR and half of the result (TIM4__ARR/2) to TIM4__DR by hardware. Note: When FOC_KFG = 0, this feature is disabled. Range (0,65535); The clock division factor TIM4_CR0[T4PSC] of Timer4 shall be adjusted if FOC_KFG overflows.						

16 Timer1

16.1 Timer1 Operations

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

Timer1 features as follows.

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

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

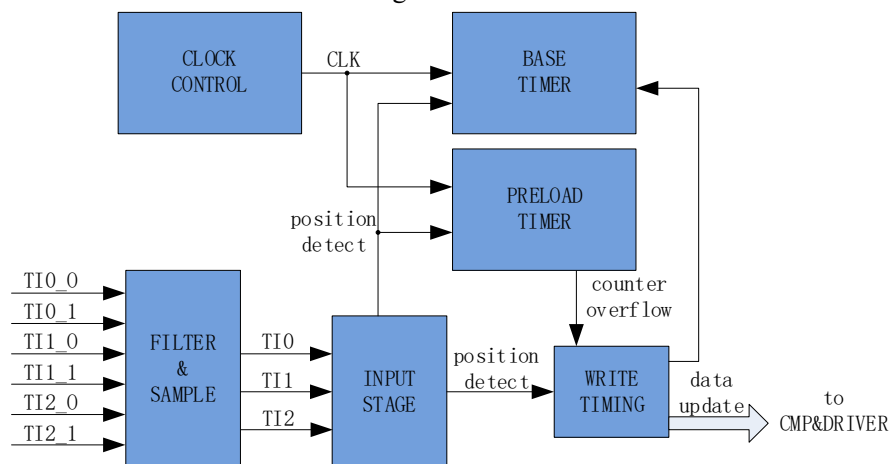


Figure 16-1 Timer1 Internal Structure

16.1.1 Timer1 Counter Module

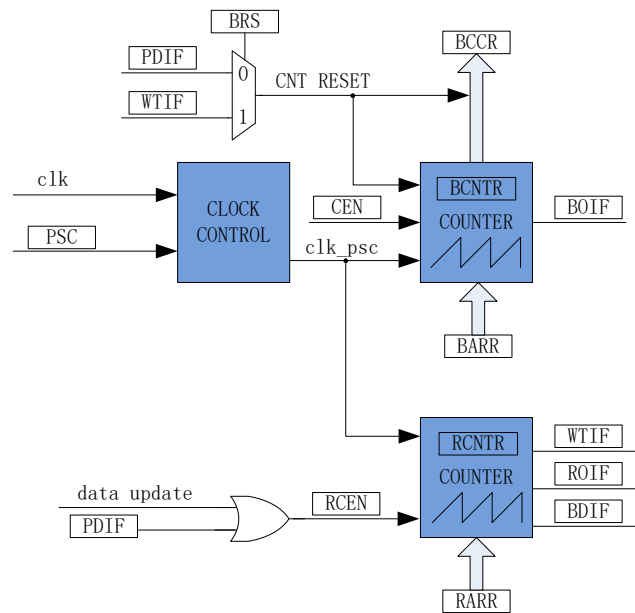


Figure 16-2 Timebase Unit

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

16.1.1.1 Prescaler

Prescaler divides the system clock frequency and generates the counter clock source for Base Timer and Reload Timer. It offers 8 division coefficients and can be selected through TIM1_CR3[T1PSC]. Since this register has no buffer, the clock rate is immediately updated after the division coefficient is written. Therefore, the division coefficient shall be configured when both the Basic Timer and Reload Timer are not working.

The clock rate can be calculated as follows:

$$f_{CK_CNT} = f_{CK_PSC} / PSC$$

Assuming that MCU clock runs at 24MHz(41.67ns):

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

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

16.1.1.2 Base Timer

The Base Timer is a 16-bit up timer with its count value held in TIM1__BCNTR. When count value of TIM1__BCNTR increases to TIM1__BARR, overflow interrupt flag TIM1_SR[T1BOIF] of the Basic Timer

is set to “1” and TIM1__BCNTR continues (instead of being cleared to “0” to restart the counter cycle). TIM1__BCNTR value is loaded into TIM1__BCCR upon a Position Detected Interrupt or a Write Timing Interrupt (selected by TIM1_CR2[T1BRS]). Meanwhile, TIM1__BCNTR is cleared to “0” and restarts the counter cycle.

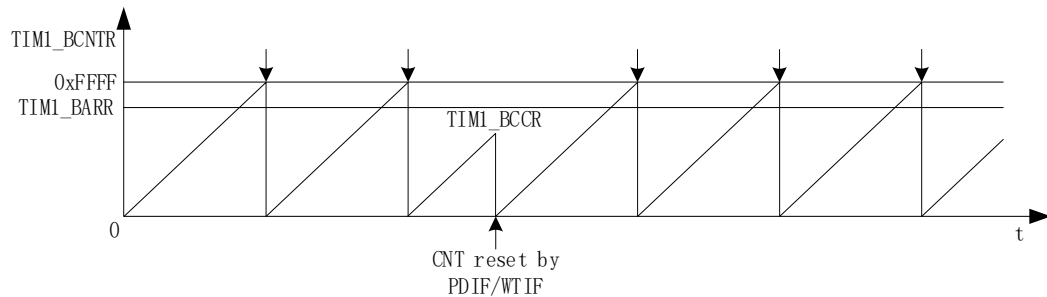


Figure 16-3 Waveform of Base Timer

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

16.1.1.3 Reload Timer

The Reload Timer is a 16-bit up timer with its count value held in TIM1__RCNTR. The timer overflows when TIM1__RCNTR increases to TIM1__RARR. In this case, TIM1_SR[T1ROIF] (overflow interrupt flag of the reload counter) is set to “1”, and TIM1__RCNTR and TIM1_CR0[T1RCEN] are cleared to “0”. The timer restarts after TIM1_CR0[T1RCEN] is set to “1”.

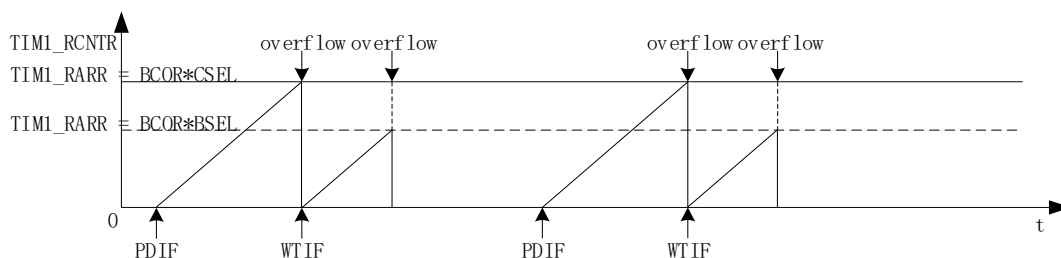


Figure 16-4 Waveform of Reload Timer

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

16.1.2 Input Filtering and Sampling

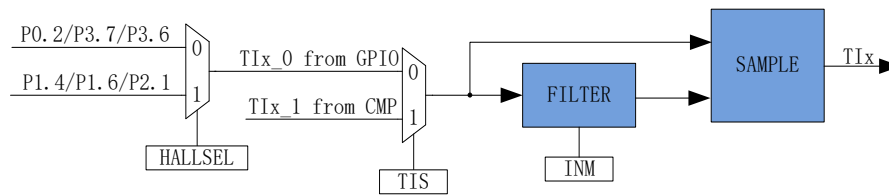


Figure 16-5 Functional Block Diagram of Input Signal Filtering and Sampling

The TIM1_CR3[T1TIS] bit selects the sources from comparator or GPIO. CMP_CR1[HALLSEL] bit is used to configure GPIO sourced by P1.4/P1.6/P2.1 or P0.2/P3.7/P3.6. TIM1_CR3[T1INM] bit decides whether CMP/GPIO signal is filtered and CMP_CR3[SAMSEL] bit determines sampling delay feature is enabled or disabled.

16.1.2.1 Filtering

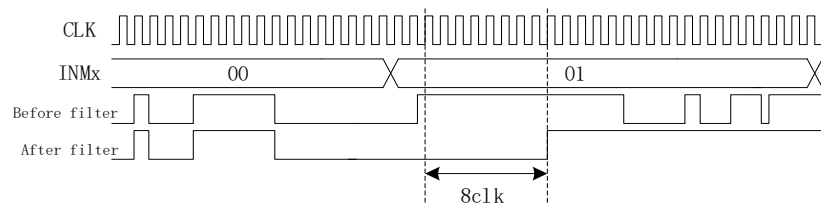


Figure 16-6 Timing Diagram of Filtering Module

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

16.1.2.2 Sampling

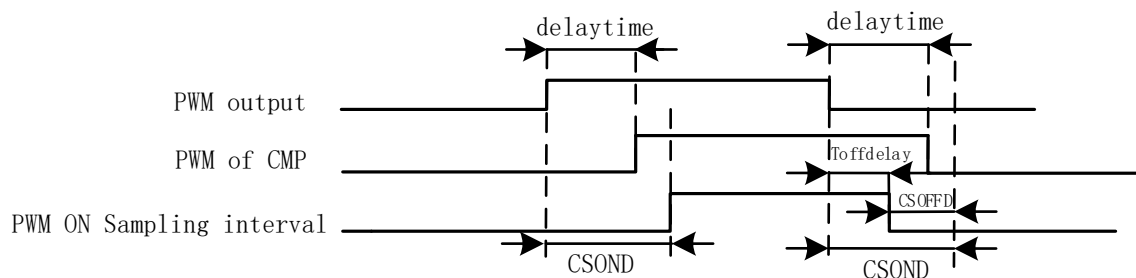


Figure 16-7 PWM ON Sampling Mode

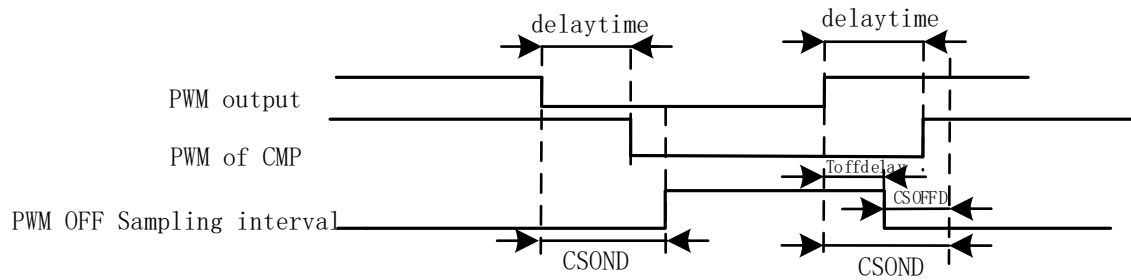


Figure 16-8 PWM OFF Sampling Mode

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

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

For example, the delay from PWM output to the output of the comparator is $2\mu\text{s}$ and the interference width is $1\mu\text{s}$, then

$$\text{CSOFFD} > 1\mu\text{s} = 1000\text{ns}/41.67\text{ns}/8 = 3$$

$$\text{CSOND} > (2 + 1)\mu\text{s} = 3000\text{ns}/41.67\text{ns}/8 = 9$$

Method for measuring the delay of PWM output to comparator: Set CMP_CR3[SAMSEL] = 00 to disable the comparator sampling delay feature. Set CMP_CR3[COMPSEL] to select the corresponding comparator output. Enable PWM output and comparator, manually rotate the motor to change the comparator value, and measure the delay between the PWM output and the comparator output.

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

16.1.3 Position Detection Event

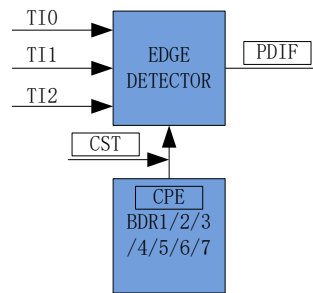


Figure 16-9 Functional Block Diagram of Position Detection

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

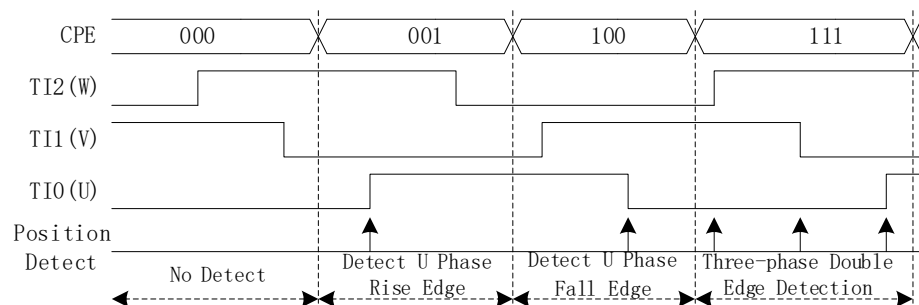


Figure 16-10 Timing Diagram of CMP/GPIO Position Detection

The relation between active edge and TIM1_DBR1/2/3/4/5/6/7[T1CPE] is shown as below.

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

16.1.4 Write Timing Interrupt

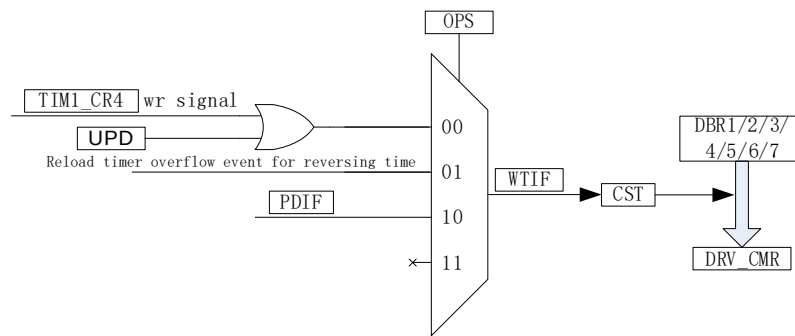


Figure 16-11 Write Timing Block Diagram

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

16.1.5 Timer1 Interrupt

Timer1 supports 6 interrupt sources:

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

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

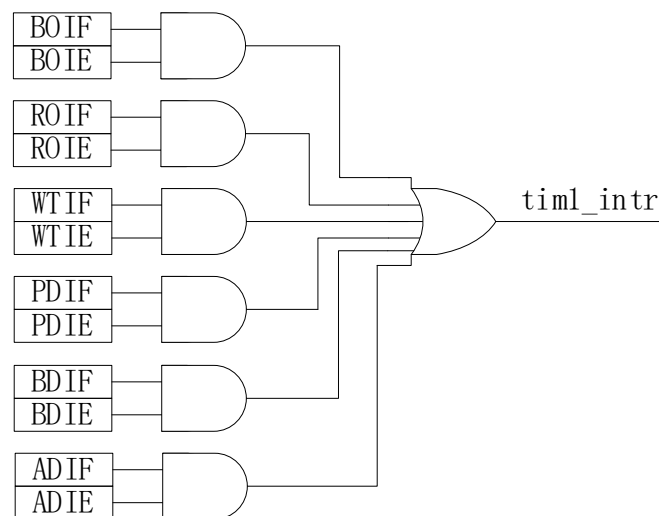


Figure 16-12 Timer1 Interrupt Sources

16.2 Square-wave Control Based BLDC Motor

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

- Automatic record of 60 degree time, filtered as 60 degree reference time
- Automatic forced phase commutation when position signal is not detected
- Automatic diode freewheeling masking, i.e., stopping comparator sampling during diode freewheeling
- Automatic control of the time from position detected to phase commutation to achieve automatic commutation
- Take over CMP_CR2[CMP0SEL] to control CMP0 automatically
- Comparator signals can be sampled during PWM ON/OFF, and the signals can be filtered as well
- Take over DRV_CMCR register to control three-phase 6 PWM outputs automatically

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

16.2.1 Six-Step Phase Commutation of Square Wave Control

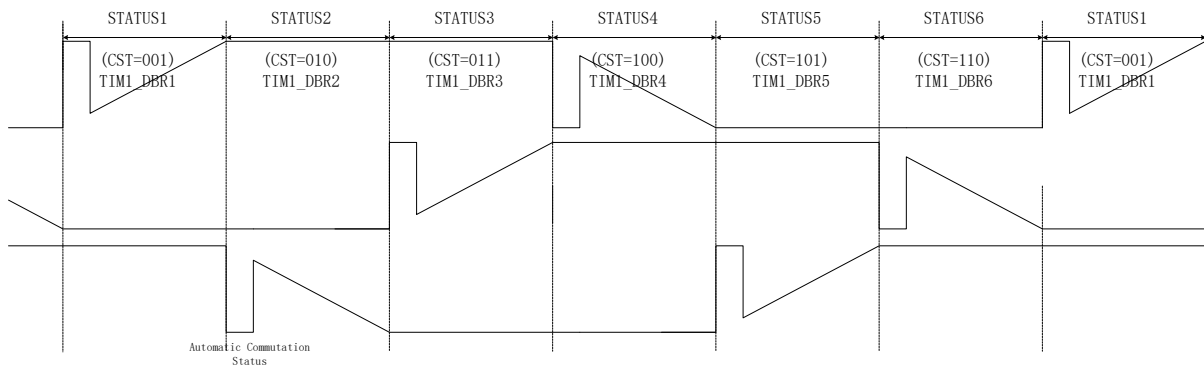


Figure 16-13 Diagram of Six-step Phase Commutation of Square Wave Control

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

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

16.2.2 Working Principle of BLDC Motor

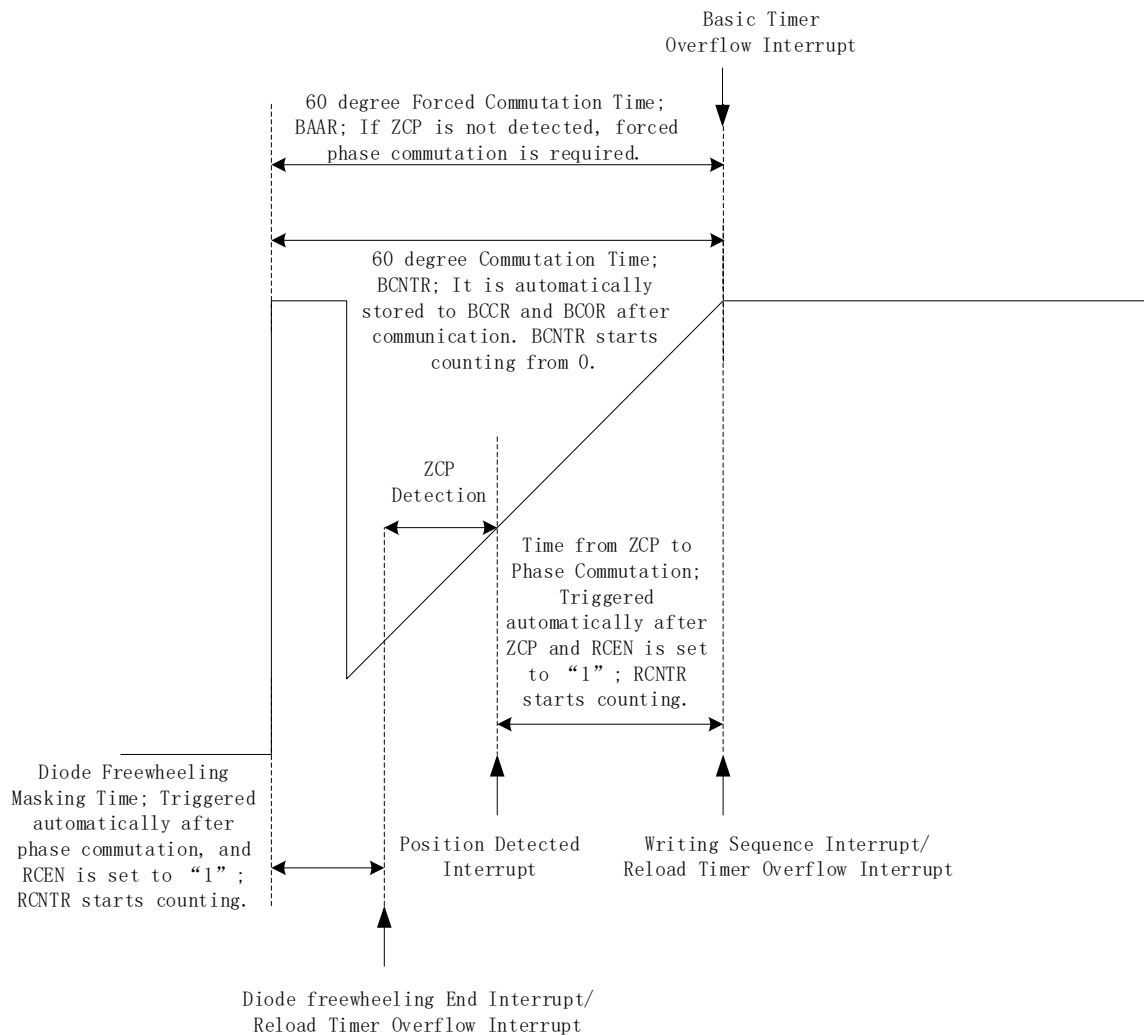


Figure 16-14 Working Principle of BLDC Motor

16.2.2.1 60° Commutation Base Time

TIM1_BCCR captures the time of last 60 degree. TIM1_CR2[T1BRS] is set to "0" to capture the time between two writing sequence interrupts and TIM1_CR2[T1BRS] to "1" to capture the time between two position detected interrupts.

TIM1_BCOR is the filtered 60 degree time, i.e., 60 degree base time. TIM1_CR0[T1CFLT] can select the last 1/2/4/8 TIM1_BCCR averaged to obtain TIM1_BCOR.

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

16.2.2.2 Phase Commutation

TIM1_CR1[T1OPS] decides phase commutation mode or writing timing event. Configuring

TIM1_CR1[T1OPS] = 00 enables sensorless control, TIM1_CR1[T1OPS] = 01 enables sensorless automatic commutation, or TIM1_CR1[T1OPS] = 10 enables sensored commutation

After phase commutation, Timer1 automatically performs the following operations:

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

16.2.2.3 Forced Commutation at 60°

When the motor rotates smoothly, ZCP is generally detected after 30 degrees of rotation after phase commutation. If ZCP is not detected in 60 degree after the phase commutation, a forced phase commutation is required. In this case, TIM1_CR0[T1FORC] is set to “1” to enable the forced commutation feature. If no ZCP is detected in 60 degree after commutation, TIM1_SR[T1BOIF] (overflow interrupt flag of the Basic Timer) is set to “1” for forced phase commutation (Note: If an ZCP is detected within 60 degrees after phase commutation, even when TIM1_BCNTR > TIM1_BARR, the forced commutation will not be triggered and TIM1_SR[T1BOIF] will not be set to “1”). When forced commutation feature is disabled (TIM1_CR0[T1FORC] = 0) and TIM1_BCNTR > TIM1_BARR, the interrupt flag TIM1_SR[T1BOIF] is set to “1” and TIM1_BCNTR continues. Phase commutation can be performed manually by Basic Timer overflow interrupt flag TIM1_SR[T1BOIF] and the position detected interrupt flag TIM1_SR[T1PDIF].

16.2.2.4 Diode Freewheeling Masking

After the commutation, inductance energy of the phase is released to the power supply or ground through the diode since the original active phase becomes a floating phase. By masking comparator signal during freewheeling time, wrong commutation caused by wrong signal generated by the freewheeling is avoided. After freewheeling masking, the freewheeling masking end interrupt flag TIM1_SR[T1BDIF] is generated.

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

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

16.2.2.5 Angle of ZCP to Phase Commutation (Delayed Commutation)

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

After commutation, a ZCP is detected (generating a position detected interrupt) and the hardware starts TIM1_RCNTNTR for counting according to the software-set time between ZCP and the commutation. After the counting ends, the hardware automatically implements phase commutation and generates the writing sequence interrupt flag TIM1_SR[T1WTIF].

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

16.2.2.6 Cycle-by-cycle Current Limiting

See section 30.1.1.2.

16.2.3 BLDC Motor Debugging

The chip supports the following debugging methods.

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

16.2.3.1 Comparator Debugging

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

Configure CMP_CR3[DBGSEL] = 11 to output comparator signals via GP01. See section 16.1.2.2 for waveforms. Select sampling delay by CMP_CR3[SAMSEL], as shown in the below table.

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

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

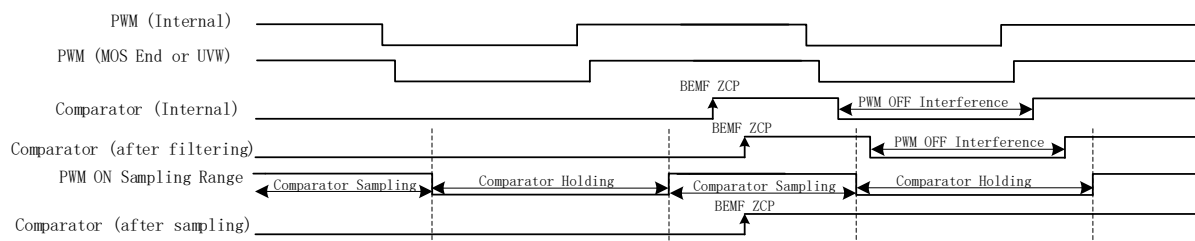


Figure 16-15 Comparator Debugging

16.2.3.2 Debugging on Diode Freewheeling Masking and Phase Commutation

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

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

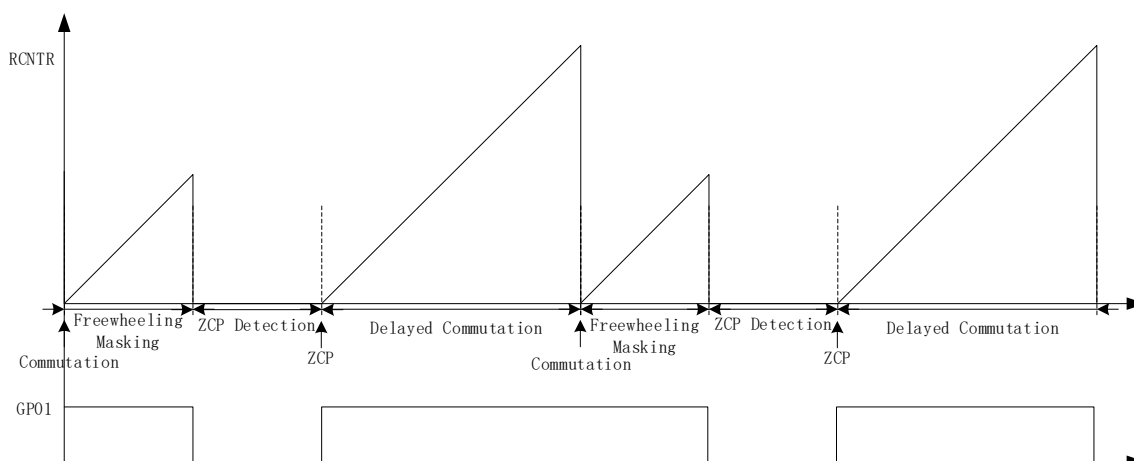


Figure 16-16 RCNTR/GP01 Waveform at 15° Freewheeling Masking and 30° Delayed Commutation

16.3 Phase Commutation Signals by ADC

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

T1CPE	Description		
000	No sampling	100	$A=V, B=U, K=TIM1_KF$
001	$A=W, B=U, K=TIM1_KR$	101	$A=V, B=W, K=TIM1_KR$
010	$A=U, B=W, K=TIM1_KF$	110	$A=W, B=V, K=TIM1_KF$
011	$A=U, B=V, K=TIM1_KR$	111	Reserved

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

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

16.4 Timer1 Registers

16.4.1 TIM1_CR0 (0x4068)

Bit	7	6	5	4	3	2	1	0
Name	T1RWEN	T1CFLT		T1FORC	T1OPS		T1BCEN	T1RCEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	T1RWEN	When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] must be set to “1” to enable TIM1_CR0[T1RCEN]. This bit can be read to “0” only.						
[6:5]	T1CFLT	60 Degree Base Time Filtering Selection The average of previous x times 60 degree is used as the base time 00: 1 times 60 degree 01: 2 times 60 degree 10: 4 times 60 degree 11: 8 times 60 degree						
[4]	T1FORC	Forced Phase Commutation at 60° Enable If no ZCP is detected after phase commutation, the hardware implements forced phase commutation. But if a ZCP is detected, forced phase commutation will not be implemented even if TIM1__BCNTR exceeds TIM1__BARR. Note: If TIM1_CR0[T1FORC] = 0, TIM1__BCNTR continues (does not restart from “0”) and forced phase commutation will not be implemented even if TIM1__BCNTR exceeds TIM1__BARR. 0: Disable 1: Enable						
[3:2]	T1OPS	Data Transfer Mode Selection The bit selects the trigger signal for TIM1_DBRx to transfer data to DRV_CMRR, i.e., write timing even or phase commutation. 00: The transfer is triggered upon a write of “1” to TIM1_IER[T1UPD] in software or on a write to TIM1_CR4[T1CST] (sensorless square-wave control) 01: The transfer is triggered upon an overflow interrupt of 16-bit reload timer during phase commutation (sensorless square-wave control) 10: The transfer is triggered upon a Position Detected Interrupt (sensored square-wave control) or ADC calculation result 11: The transfer is triggered upon an overflow interrupt of 16-bit reload timer during phase commutation or ADC calculation result						
[1]	T1BCEN	Base Timer Enable 0: Disable 1: Enable						
[0]	T1RCEN	Reload Timer Enable When TIM1_CR0 is updated, TIM1_CR0[T1RWEN] must be set to “1” to enable TIM1_CR0[T1RCEN]. TIM1_CR0[T1RCEN] is automatically enabled upon a Position Detected Interrupt and a Write Timing Interrupt. TIM1_CR0[T1RCEN] is cleared to “0” by hardware upon a Reload Timer Overflow Interrupt. 0: Disable 1: Enable						

16.4.2 TIM1_CR1 (0x4069)

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

Bit	Name	Description
[7]	T1BAPE	TIM1_BARR Register Auto-load Enable With this bit enabled, TIM1_BCOR is written to TIM1_BARR when Basic Timer is reset due to a Position Detected Interrupt or a Write Timing Interrupt. It is used for forced phase commutation at 60° when no ZCP is detected. 0: Disable 1: Enable
[6:0]	BSEL	Diode Freewheeling Masking Angle Selection The bit is used to configure the angle (time) of diode freewheeling masking after phase commutation. Position is not detected during diode freewheeling masking. Equation: Diode freewheeling masking angle = $TIM1_CR1[BSEL]/128 \times 60^\circ$ Note: This bit shall be set to “1” if the masking angle is required at 0°.

16.4.3 TIM1_CR2 (0x406A)

Bit	7	6	5	4	3	2	1	0
Name	T1BRS	CSEL						
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	T1BRS	Base Timer Reset Source Selection 0: Write Timing Reset and forced phase commutation at 60° (common settings) 1: Position Detected Interrupt Reset (ADC cannot be reset)						
[6:0]	CSEL	Phase Commutation Angle Selection After a position detected event, phase commutation is implemented after the degree configured by TIM1_CR2[CSEL]. Equation: Commutation angle = $TIM1_CR2[CSEL]/128 \times 60^\circ$ Note: This bit shall be set as “1” if the phase commutation angle is required at 0.						

16.4.4 TIM1_CR3 (0x406B)

Bit	7	6	5	4	3	2	1	0
Name	T1AFL	T1PSC			T1TIS		T1INM	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0
Bit	Name	Description						
[7]	T1AFL	ADC Sampled Voltage Calculation Filtering Counts 0: 4 1: 2						
[6:4]	T1PSC	Timer Clock Source Frequency Selection These bits are configured to divide the system clock as the clock source for Base Timer and Reload Timer. Assuming that MCU clock runs at 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	Input Source (TI0/TI1/TI2) Selection Timer1 filters, samples and generates a position detection on the selected input source. This bit affects CMP0/1/2_OUT and CMP0/1/2_IF results of the comparator module CMP_SR. 00: GPIO as the input (P1.4/P1.6/P2.1 or P0.2/P3.7/P3.6 according to CMP_CR1[7]). The results of CMP_SR is generated through the GPIO. 01: Output signal of CMP0/CMP1/CMP2 as the input. The results of CMP_SR is generated through the comparator. 1x: Reserved						

[1:0]	T1INM	Noise Pulse Width Selection for TI0/TI1/TI2 When pulse width of the noise is less than the set value, it is filtered as noise. Assuming that MCU clock runs at 24MHz(41.67ns): 00: Disable 01: 8 system clock cycles, 8 x 41.67ns 10: 32 system clock cycles, 32 x 41.67ns 11: 64 system clock cycles, 64 x 41.67ns
-------	-------	--

16.4.5 TIM1_CR4 (0x406C)

Bit	7	6	5	4	3	2	1	0
Name	RSV					T1CST		
Type	-	-	-	-	-	R/W	R/W	R/W
Reset	-	-	-	-	-	0	0	0
Bit	Name	Description						
[7:3]	RSV	Reserved						
[2:0]	T1CST	Commutation State Machine The state machine corresponds to different TIM1_DBRx at different states. When TIM1_CR4[T1CST] reads 001~111, Timer1 automatically enables or disables CMP0/1/2 according to the TIM1_DBRx[T1CPE]. When TIM1_CR4[T1CST] reads 001~110, Timer1 automatically adds by “1” each cycle upon a Write Timing Interrupt.						
		T1CST	TIM1_DBRx		T1CST	TIM1_DBRx		
		000	0		100	TIM1_DBR4		
		001	TIM1_DBR1		101	TIM1_DBR5		
		010	TIM1_DBR2		110	TIM1_DBR6		
		011	TIM1_DBR3		111	TIM1_DBR7		

16.4.6 TIM1_IER (0x406D)

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

Bit	Name	Description
[7]	T1UPD	When TIM1_CR0[T1OPS] = 00, a write of “1” to this bit enables data transfer. This bit is Write only, and automatically cleared to “0” by hardware after “1” is written.
[6]	T1MAME	Manual Mode Enable With this bit enabled, Basic Timer and Reload Timer acts as separate counters. Details: TIM1__BCNTR of the Basic Timer is cleared by a Base Timer Overflow Interrupt instead of TIM1_CR2[T1BRS] TIM1_CR0[T1RCEN] of the Reload Timer cannot be cleared to “0” or set to “1” automatically, and is controlled by software only. TIM1__RCNTR of the Reload Timer can be cleared to “0” upon a Reload Timer Overflow Interrupt only. TIM1__RARR of the Reload Timer cannot be updated automatically, and is controlled by software only. 0: Disable 1: Enable
[5]	T1ADIE	ADC Position Detected Interrupt Enable 0: Disable 1: Enable
[4]	T1BOIE	Base Timer Overflow Interrupt Enable 0: Disable 1: Enable

[3]	T1ROIE	Reload Timer Overflow Interrupt Enable 0: Disable 1: Enable
[2]	T1WTIE	Write Timing Interrupt Enable 0: Disable 1: Enable
[1]	T1PDIE	Position Detected Interrupt Enable 0: Disable 1: Enable
[0]	T1BDIE	Diode Freewheeling Masking Interrupt Enable 0: Disable 1: Enable

16.4.7 TIM1_SR (0x406E)

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

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	T1ADIF	ADC Position Detected Interrupt Flag This bit is set to “1” by hardware when a Position Detected Interrupt is generated. It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[4]	T1BOIF	Base Timer Overflow Interrupt Flag An overflow event occurs when Basic Timer counts up and TIM1__BCNTR matches with TIM1__BARR. If TIM1_CR0[T1FORC] = 1, TIM1__BCNTR is cleared to “0”, otherwise, TIM1__BCNTR continues. This bit is set to “1” by hardware, and cleared to “0” by software. Note: If TIM1__BCNTR is to be cleared during an interrupt, UPD or TIM1_CR4 is written when TIM1_CR2[T1BRS] = 0. 0: No Interrupt Pending 1: Interrupt Pending
[3]	T1ROIF	Reload Timer Overflow Interrupt Flag An overflow event occurs and TIM1__RCNTR is cleared to “0” when TIM1__RCNTR matches TIM1__RARR. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[2]	T1WTIF	Writing Sequence Interrupt Flag Writing Sequence Interrupt is generated when TIM1_DBRH/TIM1_DBRL is transferred to TIM1_DRH/TIM1_DRL. This bit is set to “1” by hardware, and cleared to “0” by software. Note: When OPS = 00, a write of “1” to this bit generates a writing sequence interrupt. 0: No Interrupt Pending 1: Interrupt Pending
[1]	T1PDIF	Position Detected Interrupt Flag A position detected interrupt is generated when Position Detection matches TIM1_DBRx[T1CPE]. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[0]	T1BDIF	Diode Freewheeling Masking End Interrupt Flag Diode freewheeling masking starts after phase commutation. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending

16.4.8 TIM1_BCOR (0x4070, 0x4071)

TIM1_BCORH(0x4070)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_BCOR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_BCORL(0x4071)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCOR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	TIM1_BCOR	The bit is configured to capture filtered count values held in the Base Timer. TIM1_BCCR holds the filtered count value, i.e. 60 Degree Base Time. Note: TIM1_BCCR and TIM1_BCOR shall be initialized when the 60 Degree Base Time is initialized. For TIM1_BCCR, the 60 Degree Base Time can be directly written. For TIM1_BCOR, please note the followings: T1CFLT=00, 60 Degree Base Time; T1CFLT=01, 60 Degree Base Time /2; T1CFLT=10, 60 Degree Base Time /4; T1CFLT=11, 60 Degree Base Time /8.						

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

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

TIM1_DBR1H(0x4074)								
Bit	15	14	13	12	11	10	9	8
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
TIM1_DBR1L(0x4075)								
Bit	7	6	5	4	3	2	1	0
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15]	RSV	Reserved						
[14:12]	T1CPE	TI0/TI1/TI2 Input Edge Polarity and Comparator Enable Select This bit is used to define the edge of Position Detection Input and enable or disable the associated comparators. The detected edge in input signal, corresponding to the configuration, generates a position detected interrupt.						

		T1CPE	Description	T1CPE	Description
		000	0	100	Phase-U corresponding comparator is enabled when falling edge of phase-U is detected.
		001	Phase-U corresponding comparator is enabled when rising edge of phase-U is detected.	101	Phase-W corresponding comparator is enabled when rising edge of phase-W is detected.
		010	Phase-W corresponding comparator is enabled when falling edge of phase-W is detected.	110	Phase-V corresponding comparator is enabled when falling edge of phase-V is detected.
		011	Phase-V corresponding comparator is enabled when rising edge of phase-V is detected.	111	Phase-U/W/V corresponding comparator is enabled when rising or falling edge of phase-U/W/V is detected.
[11]	T1WHP	High-side Output Polarity of Phase-W 0: Active High 1: Active Low			
[10]	T1WLP	Low-side Output Polarity of Phase-W 0: Active High 1: Active Low			
[9]	T1VHP	High-side Output Polarity of Phase-V 0: Active High 1: Active Low			
[8]	T1VLP	Low-side Output Polarity of Phase-V 0: Active High 1: Active Low			
[7]	T1UHP	High-side Output Polarity of Phase-U 0: Active High 1: Active Low			
[6]	T1ULP	Low-side Output Polarity of Phase-U 0: Active High 1: Active Low			
[5]	T1WHE	High-side Output Enable of Phase-W 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-W are complementary and deadtime is automatically added when WLE and WHE are set to "1".			
[4]	T1WLE	Low-side Output Enable of Phase-W 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-W are complementary and deadtime is automatically added when WLE and WHE are set to "1".			
[3]	T1VHE	High-side Output Enable of Phase-V 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-V are complementary and deadtime is automatically added when VLE and VHE are set to "1".			
[2]	T1VLE	Low-side Output Enable of Phase-V 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-V are complementary and deadtime is automatically added when VLE and VHE are set to "1".			
[1]	T1UHE	High-side Output Enable of Phase-U 0: Disable			

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

16.4.10 TIM1__BCNTR (0x4082, 0x4083)

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

16.4.11 TIM1__BCCR (0x4084, 0x4085)

TIM1__BCCRH(0x4084)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1__BCCR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__BCCRL(0x4085)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1__BCCR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	TIM1__BCCR		The bit is configured to capture count values held in Base Timer. When the Base Timer is reset on a Position Detected Interrupt or a Write Timing Interrupt, the count values before the reset are stored into TIM1__BCCR.					

16.4.12 TIM1__BARR (0x4086, 0x4087)

TIM1__BARRH(0x4086)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1__BARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__BARRL(0x4087)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1__BARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	TIM1__BARR		Auto-Reload Value in Base Timer When the count value of the Base Timer equals to TIM1__BARR value, an overflow interrupt is generated and the counter is cleared to “0”.					

16.4.13 TIM1__RARR (0x4088, 0x4089)

TIM1__RARRH(0x4088)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1__RARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__RARRL(0x4089)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1__RARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	TIM1__RARR		Auto-Reload Value in Reload Timer When count of the Reload Timer is equal to TIM1__RARR, an overflow interrupt is generated and the value of timer is cleared to “0”.					

16.4.14 TIM1__RCNTR (0x408A, 0x408B)

TIM1__RCNTRH(0x408A)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1__RCNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__RCNTRL(0x408B)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1__RCNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	TIM1__RCNTR		Count value of the Reload Timer for diode freewheeling masking and ZCP to phase commutation.					

16.4.15 TIM1__ITRIP (0x4098, 0x4099)

TIM1__ITRIPH(0x4098)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1__ITRIP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
TIM1__ITRIPL(0x4099)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1__ITRIP[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	TIM1__ITRIP		Filtered Bus Current The hardware automatically samples the bus current and filters it for software application. The default channel is ADC channel 4. Range (0,32767)					

16.4.16 TIM1__UCOP (0x408C, 0x408D)

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

16.4.17 TIM1__UFLP (0x408E, 0x408F)

TIM1__UFLPH(0x408E)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1__UFLP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__UFLPL(0x408F)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1__UFLP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	TIM1__UFLP		Floating Phase Voltage					

16.4.18 TIM1__URES (0x4090, 0x4091)

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

16.4.19 TIM1__UIGN (0x4092, 0x4093)

TIM1__UIGNH(0x4092)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1__UIGN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1__UIGNL(0x4093)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1__UIGN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[15:0]	TIM1__UIGN		Ignored Voltage of Active Phase When the voltage of active phase is lower than TIM1__UIGN, the voltage is ignored.					

16.4.20 TIM1__KF (0x4094, 0x4095)

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

16.4.21 TIM1_KR (0x4096, 0x4097)

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

16.4.22 EXT0 (0x40f0)

Bit	7	6	5	4	3	2	1	0
Name	RSV		FAEN	MOEMD 2	T1COM_ MD	CMPXO P11	TIM4_C T	EXT0_P1 1
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0
Bit	Name	Description						
[7:6]	RSV	Reserved						
[5]	FAEN	After this feature is enabled, CMP_SAMR is scaled up by 4 times. 0: Disable 1: Enable						
[4]	MOEMD2	10μs of MOE Cycle-by-cycle Current Limiting Enable (MOEMD must be configured as 1X) 0: Disable 1: Enable						
[3]	T1COM_MD	Freewheeling Masking by ADC in Square-wave Control Mode 0: Disable. Freewheeling masking is not implemented by ADC. 1: Enable. Freewheeling masking is implemented by ADC (TIM1_CR1[BSEL] ≥ 1).						
[2]	CMPXO_P11	Comparator Output Switching 0: P0.7 1: P1.1						
[1]	TIM4_CT	DBG/Timer4 Function Switching 0: Reserved 1: P0.6						
[0]	EXT0_P11	P1.1 INT0 Enable 0: Disable 1: Enable						

17 Timer2

17.1 Timer2 Instructions

Timer2 has the following five working modes:

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

Timer2 features:

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

17.1.1 Prescaler

Prescaler divides the system clock frequency and generates clock source for Base Timer. 8 frequency division coefficients of prescaler are available and can be selected by TIM2_CR0[T2PSC]. Since this register has no buffer, the clock source frequency is updated immediately after TIM2_CR0[T2PSC] is written. Therefore, the frequency division coefficients shall be configured when Basic Timer is not working.

Clock source frequency formula:

$$f_{CK_CNT} = f_{CK_PSC} / T2PSC$$

When MCU clock rate is at 24MHz(41.67ns):

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

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

17.1.2 Reading, Writing and Counting of TIM2__CNTR

When TIM2_CR1[T2EN] = 1, TIM2__CNTR starts to count. The write operation to TIM2__CNTR directly changes the value of the register, so Base Timer shall be disabled before the write operation. When reading TIM2__CNTR, software reads the high-order bits first, and the hardware synchronously caches the low-order bits. When reading the low-order bits, the software reads the cached data.

17.1.3 Output Mode

When TIM2_CR0[T2MOD] = 01B, Timer2 works in output mode.

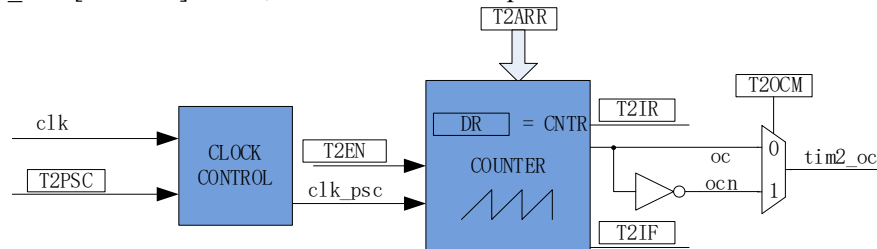


Figure 17-1 Output Mode Block Diagram

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

17.1.3.1 Reading and Writing of TIM2__ARR/TIM2__DR

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

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

17.1.3.2 High-/Low-level Output Mode

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

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

17.1.3.3 PWM Generation

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

$TIM2_CNTR \leq TIM2_DR$, and the high level is output when $TIM2_CNTR > TIM2_DR$. If $TIM2_CR0[T2OCM] = 1$, the high level is output when $TIM2_CNTR \leq TIM2_DR$, and the low level is output when $TIM2_CNTR > TIM2_DR$.

17.1.3.4 Interrupt Event

- When $TIM2_CNTR = TIM2_DR$, a compare match event is generated and the interrupt flag bit $TIM2_CR1[T2IR]$ is set to “1”. The timer continues.
- When $TIM2_CNTR = TIM2_ARR$, an overflow event is generated, and the interrupt flag bit $TIM2_CR1[T2IF]$ is set to “1”. The timer is cleared to “0” and then restarts.

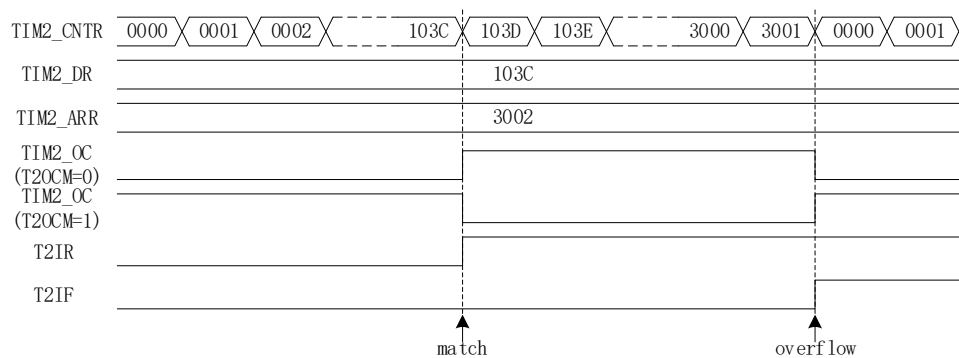


Figure 17-2 Output Mode Waveform

17.1.4 Input Signal Filtering and Edge Detection

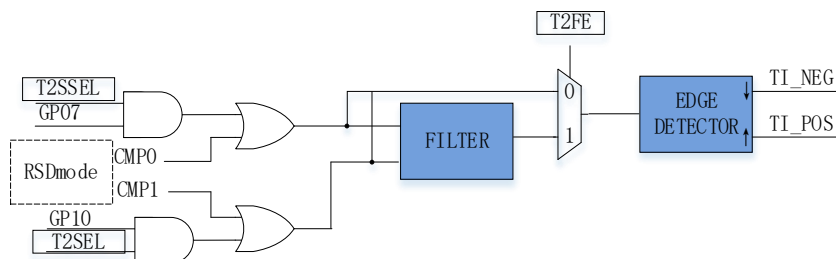


Figure 17-3 Block Diagram of Input Signal Filtering and Edge Detection

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

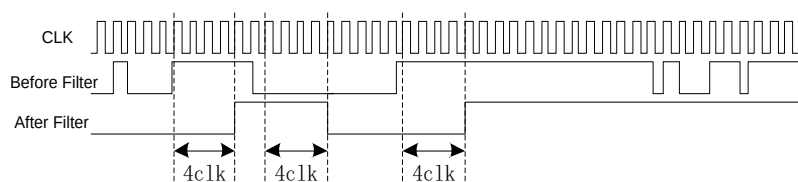


Figure 17-4 Timing Diagram of Filter Module

The filtering circuit removes the input noise with a pulse width of 4 clock cycles. The filtering feature is enabled when TIM2_CR1[T2_FE] is set to “1”. The filtered signal is about 4~5 clock cycles later than the signal before filtering.

17.1.5 Input Capture Mode

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

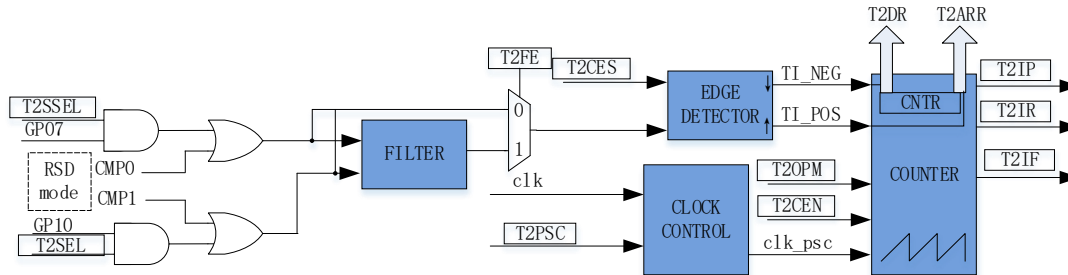


Figure 17-5 Schematic Diagram of Input Capture Mode

The input capture mode detects duty cycle and period of the PWM signal. When TIM2_CR0[T2CES] = 0, the time between two adjacent rising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). When TIM2_CR0[T2CES] = 1, the time between two adjacent falling edges forms one cycle, and the time from falling edge to rising edge forms the pulse width (LOW). When the predefined edge arrives, the count value TIM2_CNTR is stored in TIM2_DR and TIM2_ARR respectively to calculate the period and duty cycle of PWM waveform. The filter of input signal is optional.

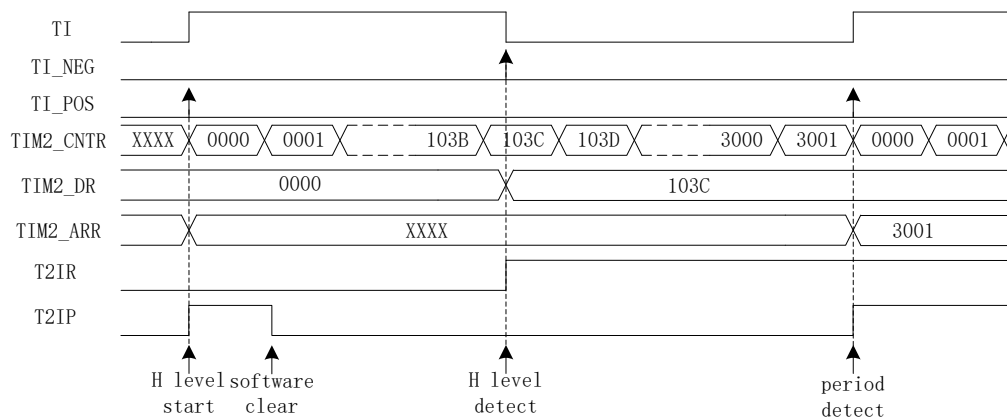


Figure 17-6 Timing Diagram of Input Capture Mode (TIM2_CR0[T2CES] = 0)

For example, when TIM2_CR0[T2CES] = 0, TIM2_CR1[T2EN] is set to “1” to enable the Base Timer. When the first rising edge of the input (falling edge is invalid) is detected, TIM2_CNTR is cleared and restarts.

When falling edge of the input is detected, the value of TIM2_CNTR is stored in TIM2_DR, and the interrupt flag TIM2_CR1[T2IR] is set to “1”, and TIM2_CNTR continues to count.

When the second rising edge of input is detected, the value of TIM2_CNTR is stored in TIM2_ARR.

Meanwhile, the interrupt flag TIM2_CR1[T2IP] is set to “1”, and TIM2_CNTR is cleared to “0” and restarts.

An overflow event occurs if Timer2 does not detect the second rising edge of the input and TIM2_CNTR reaches 0xFFFF. In this case, the interrupt flag TIM2_CR1[T2IF] is set to “1”, and TIM2_CNTR is cleared to “0” and restarts.

17.1.6 Input Counter Mode

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

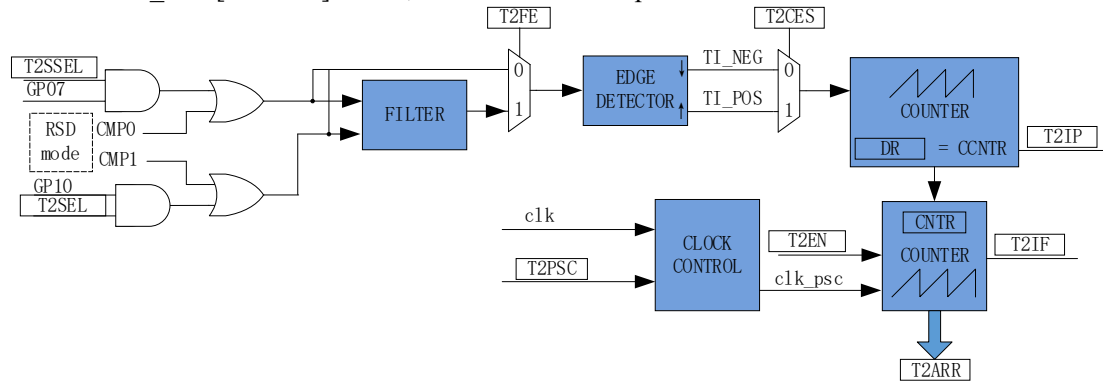


Figure 17-7 Schematic Diagram of Input Counter Mode

In input counter mode, TIM2_DR includes preload register and shadow register. When the software writes TIM2_DR register, the data is saved in the preload register first, and then sent to the shadow register in case of compare match event (TIM2_CR1[T2IP] = 1), overflow event (TIM2_CR1[T2IF] = 1) or timer disable (TIM2_CR1[T2EN] = 0). TIM2_DR is a 16-bit register, which requires the software writes the high byte first and then the low byte. The hardware ensures that the data in the preload register is not updated to the shadow register after the high byte is written and before the low byte is written.

The input counter mode is used to detect the time to input the set PWM wave. In this mode, TIM2_CNTR of the Base Timer is stored in TIM2_ARR. The filter of input signal is optional. When TIM2_CR0[T2CES] is set to “1”, the rising edge of the input PWM signal serves as the active counting edge of the special timer; when TIM2_CR0[T2CES] is set to “0”, the falling edge of the input signal as the active edge.

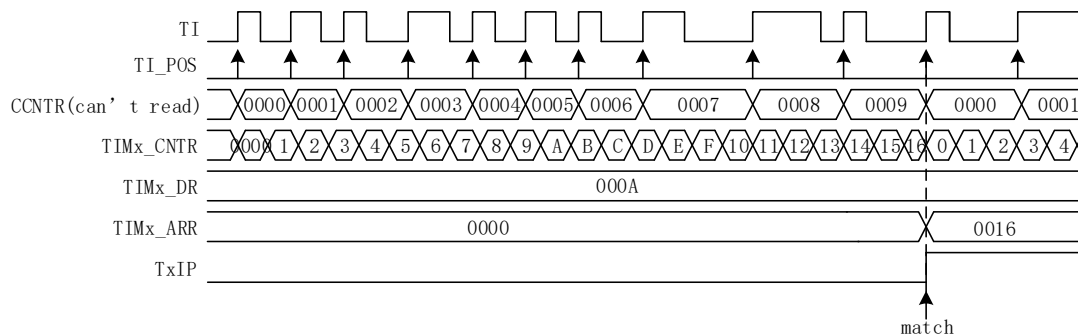


Figure 17-8 Timing Diagram of Input Counter Mode

The Basic Timer is enabled when TIM2_CR1[T2EN] is set to “1”. If the first active edge of the input signal is detected, TIM2_CNTR is cleared to “0” and restarts.

Whenever active edge of the input signal arrives, one is added to the count value of the special timer CCNTR. When the count value reaches TIM2_DR, TIM2_CNTR is stored in TIM2_ARR. When TIM2_CR1[T2IP] is set to “1”, TIM2_CNTR and CCNTR are cleared to “0” and restart.

When the number of input PWM does not reach the set value and TIM2_CNTR reaches 0xFFFF, an overflow event generates, and the interrupt flag TIM2_CR1[T2IF] is set to “1”. TIM2_CNTR is cleared to “0” with CCNTR uncleared. TIM2_CNTR starts counting from 0, and CCNTR continues counting with the previous value.

17.1.7 QEP&RSD Mode

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

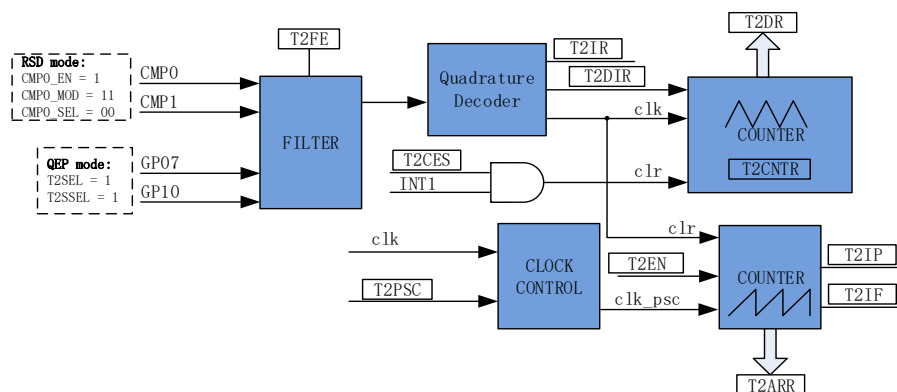


Figure 17-9 Schematic Diagram of QEP&RSD Mode

QEP & RSD mode obtains relative position, direction and speed of the motor by detecting orthogonal signals on two channels. GP07 and GP10 (QEP mode. Two inputs generally differ in phase by 90 degrees) or CMP0 and CMP1 (RSD mode. Two inputs generally differ in phase by 60 degrees) are the input signal sources, which are sent to the quadrature decoding module from the filtering module to obtain active edge and direction (TIM2_CR1[T2DIR]). TIM2_CR1[T2IF] interrupt flag is generated when the direction changes.

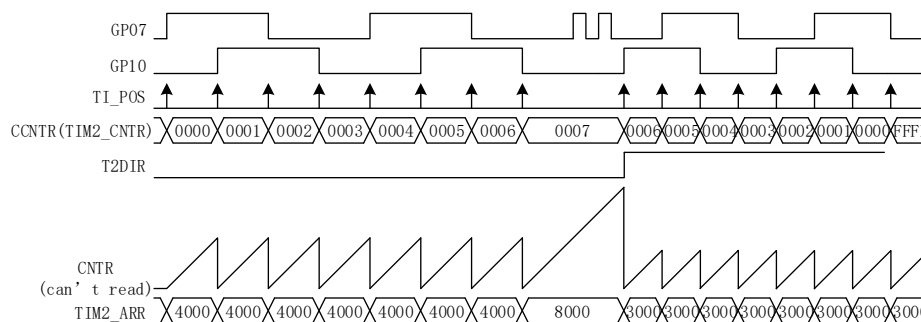


Figure 17-10 Timing Diagram of QEP Mode

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

The special timer is an up/down timer, and the signal source is the active edge from orthogonal decoding module. If TIM2_CR1[T2DIR] = 0, the direction is positive, and special timer counts upward. When the active edge arrives, the timer increases by one. If TIM2_CR1[T2DIR] = 1, the direction is reverse and special timer counts down. When the active edge arrives, the timer decreases by one. The special timer can be cleared by external interrupt INT1 and the value held by the timer is stored in TIM2_DR, after mechanical zero of the encoder is connected with any port of INT1, INT1 interrupt is enabled and TIM2_CR0[T2CES] = 1. If count value of the special timer reaches 65535 from 0, it is automatically cleared to “0”. If it decreases from 65535 to 0, it is automatically set to 65535. TIM2_CNTR is read to obtain the value of special timer.

The Base Timer is an up counter, which uses the output of prescaler as the clock source to record the time between two active counting edges. When active counting edge arrives, the value of Basic Timer is stored in TIM2_ARR and then cleared to “0”, and TIM2_CR1[T2IP] interrupt flag bit is set to “1”. When Base Timer counts to 0xFFFF, the count overflows and (TIM2_CR1[T2IF]) interrupt flag is generated.

17.1.7.1 RSD Comparator Sampling

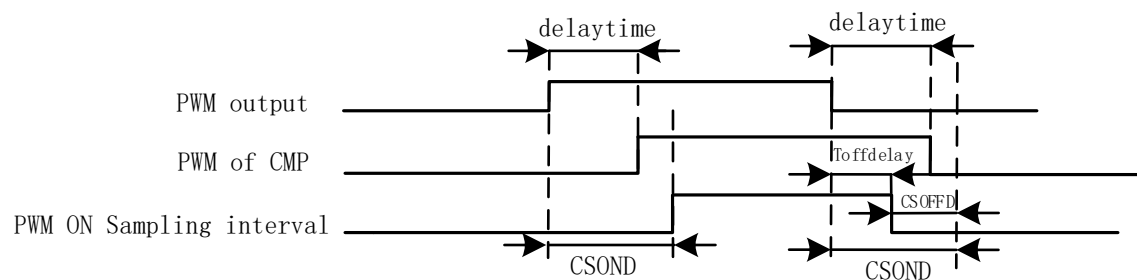


Figure 17-11 PWM ON Sampling Mode

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

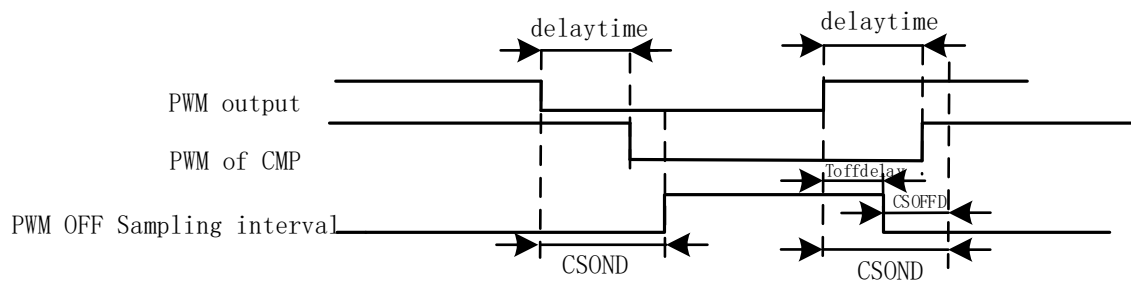


Figure 17-12 PWM OFF Sampling Mode

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

Method for measuring the delay of PWM output to comparator: Set `CMP_CR3[SAMSEL] = 00` to disable the comparator sampling delay feature. Set `CMP_CR3[CMPSSEL]` to select the corresponding comparator output. Enable the PWM output and comparator, manually rotate the motor to change the comparator value, and measure the delay between the PWM output and the comparator output.

17.1.8 Step Mode

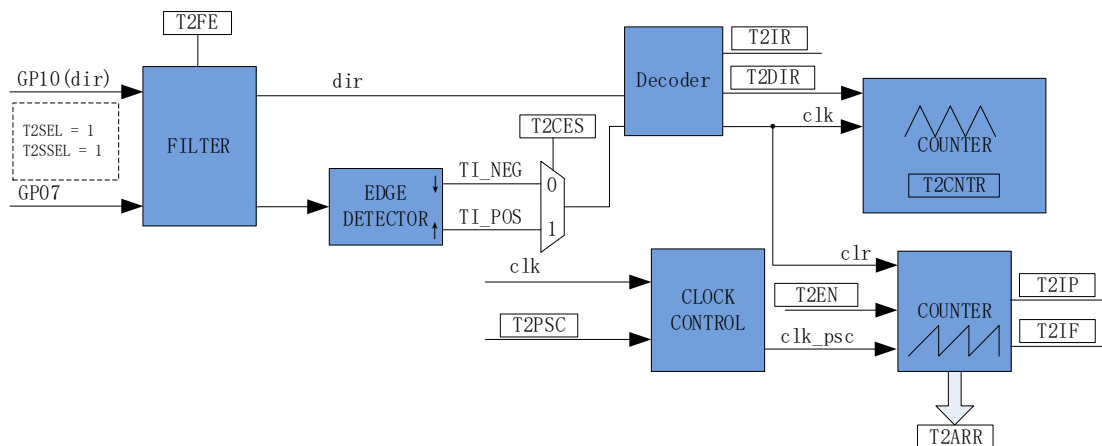


Figure 17-13 Step Mode Schematic Diagram

In step mode, relative position, direction and speed of the step motor are obtained by detecting inputs of the two channel. GP10 is direction input, and GP07 is pulse input. Setting `TIM2_CR0[T2CES]` to select the rising edge or falling edge as the active edge. The input signals are sent to decoding module from the filtering module to obtain the active edge and direction `TIM2_CR1[T2DIR]`. `TIM2_CR1[T2IF]` interrupt flag is generated when the direction changes. Note: `TIM2_CR1[T2DIR]` and `TIM2_CR1[T2IR]` will not change unless transition occurs at GP10 and active edge is detected at GP07. To generate an interrupt immediately

after GP10 changes, use INT1.

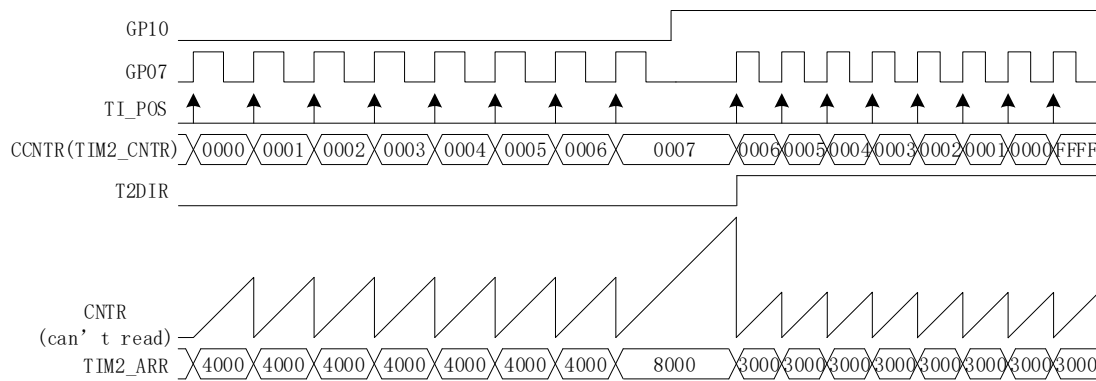


Figure 17-14 Timing Diagram of Step Mode

The special timer is an up/down-counter, and the signal source is active edge of the encoding module. When GP10=0 and TIM2_CR1[T2DIR] = 0, the direction is forward. If active edge of GP07 arrives, the special timer increases by 1. When GP10 = 1 and TIM2_CR1[T2DIR] = 1, the direction is reverse. If active edge of GP07 arrives, the special timer decreases by 1. If count value of the special timer reaches 65535 from 0, it is automatically cleared to "0". If it decreases from 65535 to 0, it is automatically set to 65535. TIM2_CNTR is read to obtain the value of special timer.

The Base Timer is an up counter, which uses the output of prescaler as the clock source to record the time between two active counting edges. When active counting edge arrives, the value of Basic Timer is stored in TIM2_ARR and then cleared to "0", and TIM2_CR1[T2IP] interrupt flag bit is set to "1". When Base Timer counts to 0xFFFF, the count overflows and (TIM2_CR1[T2IF]) interrupt flag is generated.

17.2 Timer2 Registers

17.2.1 TIM2_CR0(0xA1)

Bit	7	6	5	4	3	2	1	0
Name	T2PSC			T2OCM	T2IRE	T2CES	T2MOD	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:5]	T2PSC	Base Timer Clock Prescaler Selection It is configured to divide the system clock frequency and generate the clock source for Base Timer. When MCU clock runs at 24MHz(41.67ns), the prescaled clock rates are configured as follows: 000:0x1 (24MHz) 001:0x2 (12MHz) 010:0x4 (6MHz) 011:0x8 (3MHz) 100:0x10 (1.5MHz) 101:0x20 (750kHz) 110:0x40 (375kHz) 111:0x80 (187.5kHz)						
[4]	T2OCM	Output Mode: Output Mode Selection 0: “0” is output when TIM2_CNTR ≤ TIM2_DR; “1” is output when TIM2_CNTR > TIM2_DR. 1: “1” is output when TIM2_CNTR ≤ TIM2_DR; “0” is output when TIM2_CNTR > TIM2_DR. Input Counter Mode: No effect Input Capture Mode: No effect QEP&RSD Mode and Step Mode Selection 0: QEP&RSD Mode 1: Step Mode						
[3]	T2IRE	Output Mode: Match Interrupt Enable Input Capture Mode: Pulse Width Detection Interrupt Enable Input Counter Mode: No effect QEP&RSD Mode and Step Mode: Direction Change Interrupt Enable 0: Disable 1: Enable						
[2]	T2CES	Input Capture Mode: Counting Edge Selection 0: The time between two adjacent raising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). 1: The time between two adjacent falling edges forms one cycle, and the time from falling edge to raising edge forms the pulse width (LOW). Input Counter Mode and Step Mode: Active Edge Selection 0: Falling Edge Count 1: Raising Edge Count QEP&RSD Mode: Pulse Counter Cleared Enable upon Z Signal Interrupt INT1 0: Disable 1: Enable						
[1:0]	T2MOD	Mode Selection 00: Input Capture Mode 01: Output Mode 10: Input Counter Mode 11: QEP&RSD Mode or Step Mode						

17.2.2 TIM2_CR1(0xA9)

Bit	7	6	5	4	3	2	1	0
Name	T2IR	T2IP	T2IF	T2IPE	T2IFE	T2FE	T2DIR	T2EN
Type	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	T2IR	Output Mode: Match Interrupt Flag The bit is set to “1” by the hardware when TIM2__CNTR matches TIM2__DR. It is cleared to “0” by software. Input Capture Mode: Pulse Width Detection Interrupt Flag This bit is set to “1” when an input pulse width is detected (Setting TIM2_CR0[T2CES] to select the rising edge or falling edge as the active edge). It is cleared to “0” by software. Input Counter Mode: No effect QEP&RSD Mode and Step Mode: Direction Change Flag Bit 0: No Interrupt Pending 1: Interrupt Pending						
[6]	T2IP	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag This bit is set to “1” when an input PWM cycle is detected (Setting TIM2_CR0[T2CES] to select the rising edge or falling edge as the active edge). It is cleared to “0” by software. Input Counter Mode: PWM Input Counter Match Interrupt Flag This bit is set to “1” by hardware when the number of input PWM reaches TIM2__DR. It is cleared to “0” by software. QEP&RSD Mode and Step Mode: Active Edge Detection Interrupt Flag This bit is set to “1” by hardware when the input edge is detected as an active edge. It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[5]	T2IF	Output Mode: Timer Overflow Interrupt Flag. This bit is set to “1” and TIM2__CNTR is cleared to “0” when TIM2__CNTR matches TIM2__ARR. It is cleared to “0” by software. Input Capture Mode: Timer Overflow Interrupt Flag Setting TIM2_CR0[T2CES] to select the rising edge or falling edge as the active edge. This bit is set to “1” and TIM2__CNTR is cleared to “0” when the Timer has not detected an input PWM cycle but the timer TIM2__CNTR reaches 0xFFFF (overflow occurs). It is cleared to “0” by software. Input Counter Mode: Base Timer Overflow Interrupt Flag This bit is set to “1” and TIM2__CNTR is cleared to “0” when the input PWM cycle has not reached the preset TIM2__DR value but Base Timer TIM2__CNTR value reaches 0xFFFF (overflow occurs). It is cleared to “0” by software. QEP&RSD Mode and Step Mode: Base Timer Overflow Interrupt Flag This bit is set to “1” and Basic Timer is cleared to “0” when Basic Timer reaches to 0xFFFF (overflow occurs). It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[4]	T2IPE	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Enable Input Counter Mode: PWM Input Counter Match Interrupt Enable QEP&RSD Mode and Step Mode: Active Edge Detection Interrupt Enable 0: Disable 1: Enable						
[3]	T2IFE	Output Mode: Timer Overflow Interrupt Enable Input Capture Mode: Timer Overflow Interrupt Enable Input Counter Mode: Base Timer Overflow Interrupt Enable QEP&RSD Mode and Step Mode: Base Timer Overflow Interrupt Enable						

		0: Disable 1: Enable
[2]	T2FE	Input Signal Filter Selection When input signals are filtered out as noise if the pulse width is less than 4 clock cycle. Assuming that MCU clock runs at 24MHz (41.67ns) , then the pulse width for filtering is 166.67ns. 0: Disable 1: Enable
[1]	T2DIR	QEP&RSD Mode and Steo Mode: Indicator of Motor Rotation Direction 0: Forward 1: Backward
[0]	T2EN	Timer Enable 0: Disable 1: Enable

17.2.3 PI_LPF_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2SS	RSV				PIRANGE	PISTA	LPFSTA
Type	R/W	-	-	-	-	R/W	R/W	R/W
Reset	0	-	-	-	-	0	0	0
Bit	Name	Description						
[7]	T2SS	Input Mode Selection of TIM2 Stepper Motor 0: P10 is direction input and P07 is pulse input 1: P10 is reverse pulse input and P07 is the forward pulse input						
[6:3]	RSV	Reserved						
[2:0]		See section 12.2.1.						

17.2.4 TIM2__CNTR(0xAA,0xAB)

TIM2__CNTRH (0xAB)								
Bit	15	14	13	12	11	10	9	8
Name	TIM2__CNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2__CNTRL (0xAA)								
Bit	7	6	5	4	3	2	1	0
Name	TIM2__CNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	TIM2__CNTR	Output Mode/Input Capture Mode/Input Counter Mode: Count values held in the Base Timer QEP&RSD Mode/Step Mode: Count values held in the special timer When TIM2__CNTR is 0xFFFF, this bit is automatically cleared to "0".						

17.2.5 TIM2__DR(0xAC,0xAD)

TIM2__DRH (0xAD)								
Bit	15	14	13	12	11	10	9	8
Name	TIM2__DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2__DRL (0xAC)								
Bit	7	6	5	4	3	2	1	0
Name	TIM2__DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	TIM2__DR	Output Mode: Compare match value (written by software) Input Capture Mode: Count value of the detected input pulse width based on TIM2_CR0[T2CES] (written by hardware) Input Counter Mode: PWM cycles to be counted (written by software) QEP&RSD Mode: Value of the special timer when TIM2_CR0[T2CES] = 1 and INT1 (zero point) is detected (written by hardware) Step Mode: No effect						

17.2.6 TIM2__ARR(0xAE,0xAF)

TIM2__ARRH (0xAF)								
Bit	15	14	13	12	11	10	9	8
Name	TIM2__ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2__ARRL (0xAE)								
Bit	7	6	5	4	3	2	1	0
Name	TIM2__ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	TIM2__ARR	Output Mode: Overload value (written by software) Input Capture Mode: Count value of a PWM cycle (written by hardware) Input Counter Mode: Count value held in Base Timer when the input PWM count matches (written by hardware) QEP&RSD Mode and Step Mode: Count value held in Base Timer when the input signal is detected as an active edge (written by hardware)						

18 Timer3/Timer4

18.1 Timer3/Timer4 Instructions

Timer3/Timer4 support output and input modes:

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

Timer3/Timer4 includes:

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

18.1.1 Prescaler

Prescaler divides the system clock frequency and generates counter clock source for Base Timer. 8 frequency division coefficients of prescaler are available and can be selected by TIMx_CR0[TxPSC]. Since this register has no buffer, the clock source frequency is updated immediately after TIMx_CR0[TxPSC] is written. Therefore, the frequency division coefficients shall be configured when Basic Timer is not working.

The clock source frequency formula:

$$f_{CK_CNT} = f_{CK_PSC} / TxPSC$$

When MCU clock runs at 24MHz(41.67ns):

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

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

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

18.1.2 Reading, Writing and Counting of TIMx__CNTR

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

low-order bits simultaneously. When reading the low-order bits, the software reads the cached data.

18.1.3 Output Mode

When $TIMx_CR0[TxMOD] = 1$, Timer3/Timer4 works in output mode.

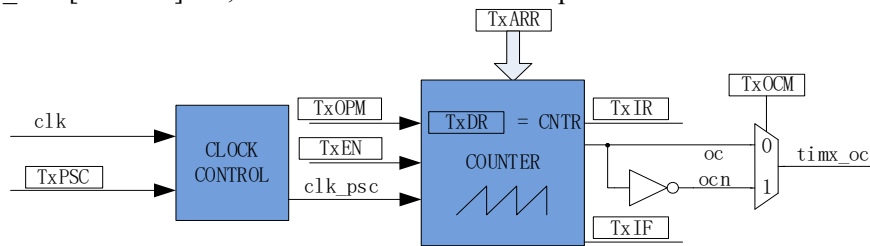


Figure 18-1 Output Mode Block Diagram

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

18.1.3.1 High-/Low-level Output Mode

When $TIMx_CR0[TxOCM] = 0$ and $TIMx_DR = TIMx_ARR$, the output signals are always low. When $TIMx_CR0[TxOCM] = 1$ and $TIMx_DR = TIMx_ARR$, the output signals are always high.

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

18.1.3.2 PWM Generation

In PWM generation mode, $TIMx_ARR$ determines PWM cycle, and $TIMx_DR$ determines the duty cycle, and duty cycle = $TIMx_DR / TIMx_ARR * 100\%$. If $TIMx_CR0[TxOCM] = 0$, the low level is output when $TIMx_CNTR \leq TIMx_DR$, and the high level is output when $TIMx_CNTR > TIMx_DR$. If $TIMx_CR0[TxOCM] = 1$, the high level is output when $TIMx_CNTR \leq TIMx_DR$, and low level is output when $TIMx_CNTR > TIMx_DR$.

18.1.3.3 Interrupt Event

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

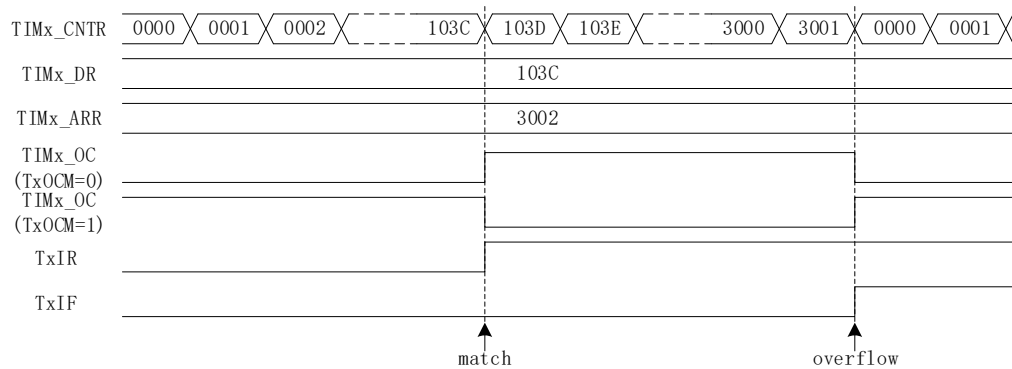


Figure 18-2 Output Waveform of Output Mode

18.1.4 Input Signal Filtering and Edge Detection

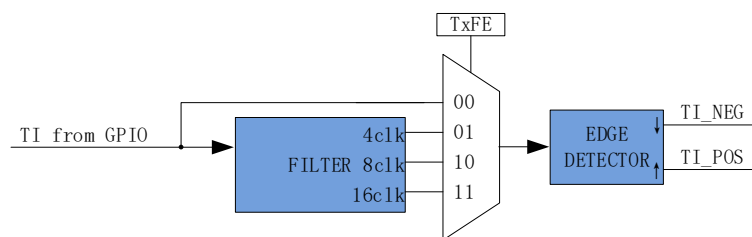


Figure 18-3 Block Diagram of Input Signal Filtering and Edge Detection

The input signal TI of Timer3/Timer4 comes from GP11/GP01. The filter of input signal is optional. The edge detection module detects filtered input signals and records rising edge and falling edge for use by the next module.

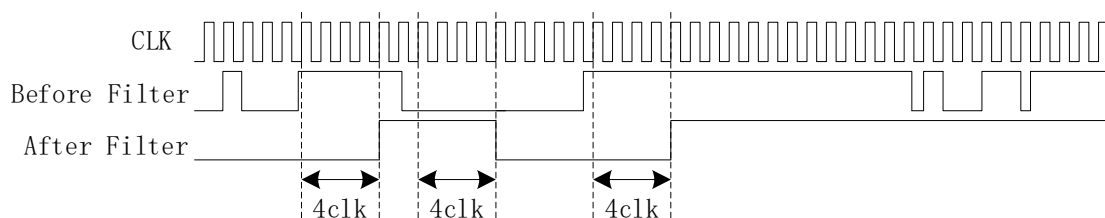


Figure 18-4 4clk Timing Diagram of Filter Module

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

18.1.5 Input Capture Mode

When $TIMx_CR0[TxMOD] = 0$, Timer3/Timer4 works in input capture mode.

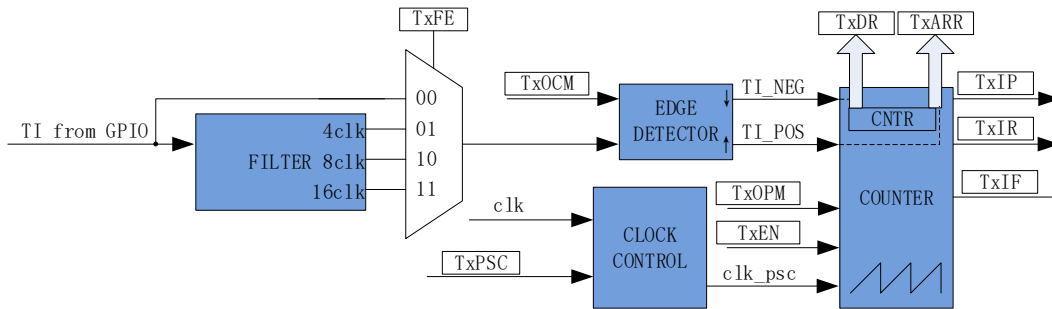


Figure 18-5 Schematic Diagram of Input Capture Mode

The Input Capture Mode detects pulse width and waveform period of the input PWM signals. When $TIMx_CR0[TxOCM] = 0$, the time between two adjacent rising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). When $TIMx_CR0[TxOCM] = 1$, the time between two adjacent falling edges forms one cycle, and the time from falling edge to rising edge forms the pulse width (LOW). The pulse width and the period obtained by $TIMx_CNTR$ are stored in $TIMx_DR$ and $TIMx_ARR$ respectively.

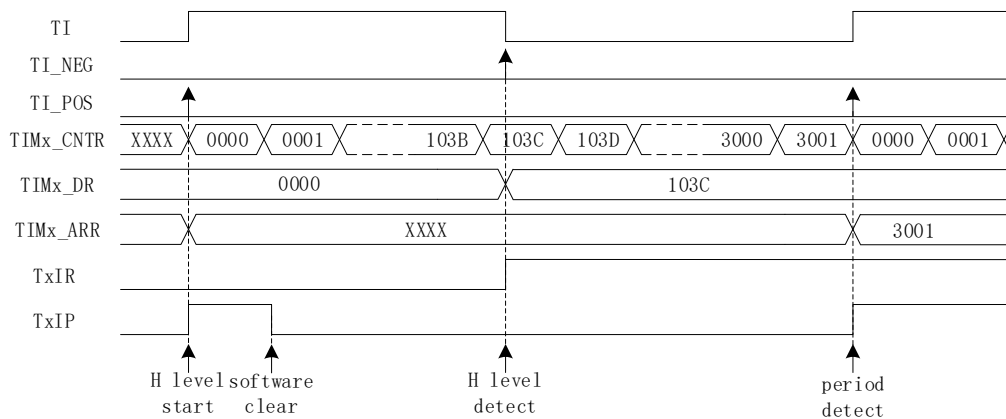


Figure 18-6 Timing Diagram of Input Capture Mode ($TIMx_CR0[TxOCM] = 0$)

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

When the falling edge is detected, the value of $TIMx_CNTR$ is stored into $TIMx_DR$. Meanwhile, the interrupt flag $TIMx_CR1[TxIR]$ is set to "1", and $TIMx_CNTR$ continues to count.

When the second rising edge is detected, the value of $TIMx_CNTR$ is saved into $TIMx_ARR$. The interrupt flag $TIMx_CR1[TxIP]$ is set to "1" and $TIMx_CNTR$ is cleared to "0". $TIMx_CR0[TxOPM]$ determines whether the timer restarts. If $TIMx_CR0[TxOPM] = 1$, the timer stops; and if $TIMx_CR0[TxOPM] = 0$, it restarts.

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

TIMx_CR0[TxOCM]= 1, the timer stops counting, and if TIMx_CR0[TxOPM] = 0, it restarts.

18.1.6 FG Generation (Timer4)

See section FG Generation.

18.2 Timer3/Timer4 Registers

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

Bit	7	6	5	4	3	2	1	0
Name	TxPSC			TxOCM	TxIRE	RSV	TxOPM	TxMOD
Type	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:5]	TxPSC	Base Timer Clock Prescaler Selection It is configured to divide the system clock frequency and generate the clock source for Base Timer. When MCU clock runs at 24MHz(41.67ns), the prescaled clock rates are configured as follows: 000:0x1 (24MHz) 001:0x2 (12MHz) 010:0x4 (6MHz) 011:0x8 (3MHz) 100:0x10 (1.5MHz) 101:0x20 (750kHz) 110:0x40 (375kHz) 111:0x80 (187.5kHz)						
[4]	TxOCM	Output Mode: Output Mode Selection 0: "0" is output when TIMx_CNTR≤TIMx_DR; "1" is output when TIMx_CNTR>TIMx_DR. 1: "1" is output when TIMx_CNTR≤TIMx_DR; "0" is output when TIMx_CNTR>TIMx_DR. Input Capture Mode: Active Edge Selection 0: The time between two adjacent raising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). 1: The time between two adjacent falling edges forms one cycle, and the time from falling edge to raising edge forms the pulse width (LOW).						
[3]	TxIRE	Output Mode: Compare Match Interrupt Enable Input Capture Mode: Pulse Width Detection Interrupt Enable 0: Disable 1: Enable						
[2]	RSV	Reserved						
[1]	TxOPM	Single Mode The timer stops in any of the following events: Output Mode: Timer overflow event Input Capture Mode: PWM Cycle Detection or Timer Overflow Event 0: The Timer does not stop 1: The Timer stops (TIMx_CR1[TxEN] is reset to "0")						
[0]	TxMOD	Working Mode Selection 0: Input Capture Mode 1: Output Mode						

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

Bit	7	6	5	4	3	2	1	0
Name	TxIR	TxIP	TxIF	TxIDE	TxIFE	TxFE		TxEN
Type	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	TxIR	Output Mode: Match Interrupt Flag The bit is set to “1” by the hardware when TIM3__CNTR matches TIM3__DR. It is cleared to “0” by software. Input Capture Mode: Pulse Width Detection Interrupt Flag This bit is set to “1” when an input pulse width is detected (Setting TIM3_CR0[T2CES] to select the rising edge or falling edge as the active edge). This bit is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[6]	TxIP	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag This bit is set to “1” when an input PWM cycle is detected (Setting TIMx_CR0[T2CES] to select the rising edge or falling edge as the active edge). It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[5]	TxIF	Output Mode: Timer Overflow Interrupt Flag. This bit is set to “1” and TIMx__CNTR is cleared to “0” when TIMx__CNTR matches TIMx__ARR. It is cleared to “0” by software. Input Capture Mode: Timer Overflow Interrupt Flag This bit is set to “1” and TIMx__CNTR is cleared to “0” when the timer does not detect an input PWM cycle but the timer TIMx__CNTR reaches 0xFFFF (overflow occurs). It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[4]	TxIPE	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Enable 0: Disable 1: Enable						
[3]	TxIFE	Output Mode: Timer Overflow Interrupt Input Capture Mode: Timer Overflow Interrupt Enable 0: Disable 1: Enable						
[2:1]	TxFE	Input Signal Filtering Pulse Width Selection Input signals are filtered as noise if pulse width is less than the defined value. Assuming that MCU clock runs at 24MHz(41.67ns): 00: Disable 01: 4 system clock cycles, 4 x 41.67ns 10: 8 system clock cycles, 8 x 41.67ns 11: 16 system clock cycles, 16 x 41.67ns						
[0]	TxEN	Base Timer Enable 0: Disable 1: Enable						

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

TIMx_CNTRH (0xA3/0x93)								
Bit	15	14	13	12	11	10	9	8
Name	TIMx_CNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIMx_CNTRL (0xA2/0x92)								
Bit	7	6	5	4	3	2	1	0
Name	TIMx_CNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	TIMx_CNTR	Count Values Held in Base Timer When TIMx_CNTR reaches 0xFFFF, this bit is automatically cleared to “0”.						

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

TIMx_DRH (0xA5/0x95)								
Bit	15	14	13	12	11	10	9	8
Name	TIMx_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIMx_DRL (0xA4/0x94)								
Bit	7	6	5	4	3	2	1	0
Name	TIMx_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	TIMx_DR	Output Mode: Compare match values (written by software) Input Capture Mode: Count value of the detected input pulse width based on TIMx_CR0[TxOCM] (written by hardware)						

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

TIMx_ARRH (0xA7/0x97)								
Bit	15	14	13	12	11	10	9	8
Name	TIMx_ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIMx_ARRL (0xA6/0x96)								
Bit	7	6	5	4	3	2	1	0
Name	TIMx_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	TIMx_ARR	Output Mode: Overload value (written by software) Input Capture Mode: Count value of a PWM cycle based on TIMx_CR0[TxOCM] (written by hardware)						

19 Systick

19.1 Systick Instructions

The chip can generate Systick interrupts at a fixed interval, and the interrupt cycle is controlled by SYST_ARR. Systick interrupt is enabled when DRV_SR[SYSTIE] is set to “1”, and the interrupts are accessed by P10 (multiplexed with TIM4 interrupt input).

19.2 Systick Registers

19.2.1 DRV_SR (0x4061)

Bit	7	6	5	4	3	2	1	0
Name	SYSTIF	SYSTIE	FGIF	DCIF	FGIE	DCIP	DCIM	
Type	R/W0	R/W	R/W0	R/W0	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	SYSTIF	Systick Interrupt Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[6]	SYSTIE	Systick Interrupt Enable 0: Disable 1: Enable						
[5:0]		See section 20.2.2.						

19.2.2 SYST_ARR (0x4064,0x4065)

SYST_ARRH(0x4064)								
Bit	15	14	13	12	11	10	9	8
Name	SYST_ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	1	1	0	1
SYST_ARRL(0x4065)								
Bit	7	6	5	4	3	2	1	0
Name	SYST_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	1	1	1	1	1
Bit	Name	Description						
[15:0]	SYST_ARR	Systick Cycle This bit determines the cycle at which Systick interrupts are generated, which defaults to 1ms. Calculation formula is as follows: Systick rate = 24M/(SYST_ARR + 1) Range (0,65535)						

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

20.1.2.1 Count and Compare Module

$DRV_CR[OCS]$ is configured to select the comparison value of PWM from $FOC_CMPU/V/W$ of FOC module or DRV_DR set by software. The comparison value is sent to the counter for comparison to obtain a 4-way PWM signal $OCxREF$, and DRV_DR is used for motor pre-charging, braking and square-wave control. If DRV_CNTR is smaller than the comparison value, $OCxREF$ outputs high-level signal, and if DRV_CNTR is larger than DRV_DR , $OCxREF$ outputs low-level signal.

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

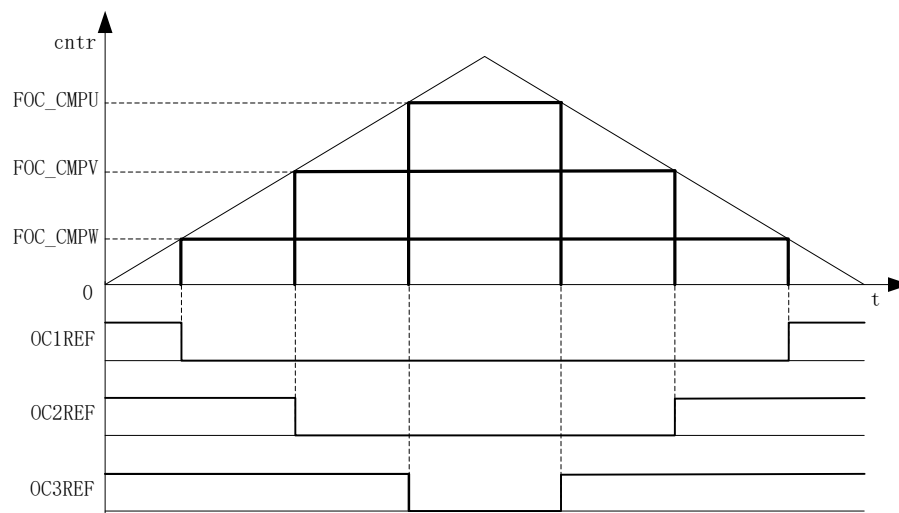


Figure 20-3 PWM Generation

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

Duty cycle = $DRV_DR/DRV_ARR \times 100\%$ (For example, if $DRV_ARR=750$ and $DRV_DR = 375$, then duty cycle = 50%)

20.1.2.2 Deadtime Module

$OCxREF$ supports deadtime insertion. For complementary outputs, the deadtime insertion is enabled when DRV_DTR is not "0". Each channel has an 8-bit deadtime generator, and four channels have the same deadtime, which is set by DRV_DTR . When rising edge signals are detected, output high level of $OCxL$ is

delayed for a period of time set in DRV_DTR. When falling edge signals are detected, output high level of OCxH is delayed for a period of time set in DRV_DTR.

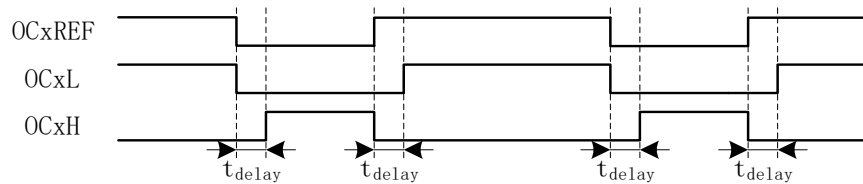


Figure 20-4 Complementary Outputs with Deadtime Insertion

20.1.2.3 Enable and Polarity of Output Signals

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

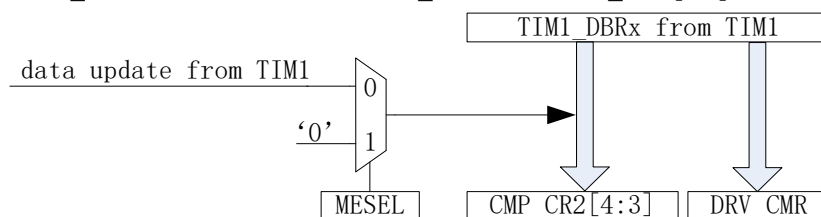


Figure 20-5 Timer1 Automatic Control of DRV_CMR and CMP_CR2[4:3]

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

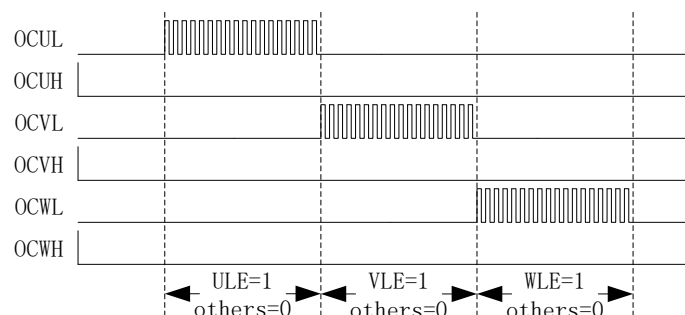


Figure 20-6 Pre-charge Waveform

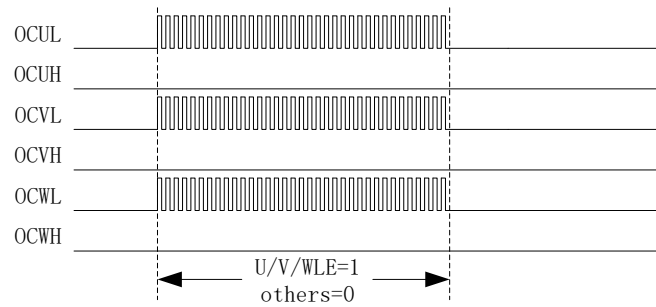


Figure 20-7 Brake Waveform

20.1.2.4 Main Output Enable (MOE)

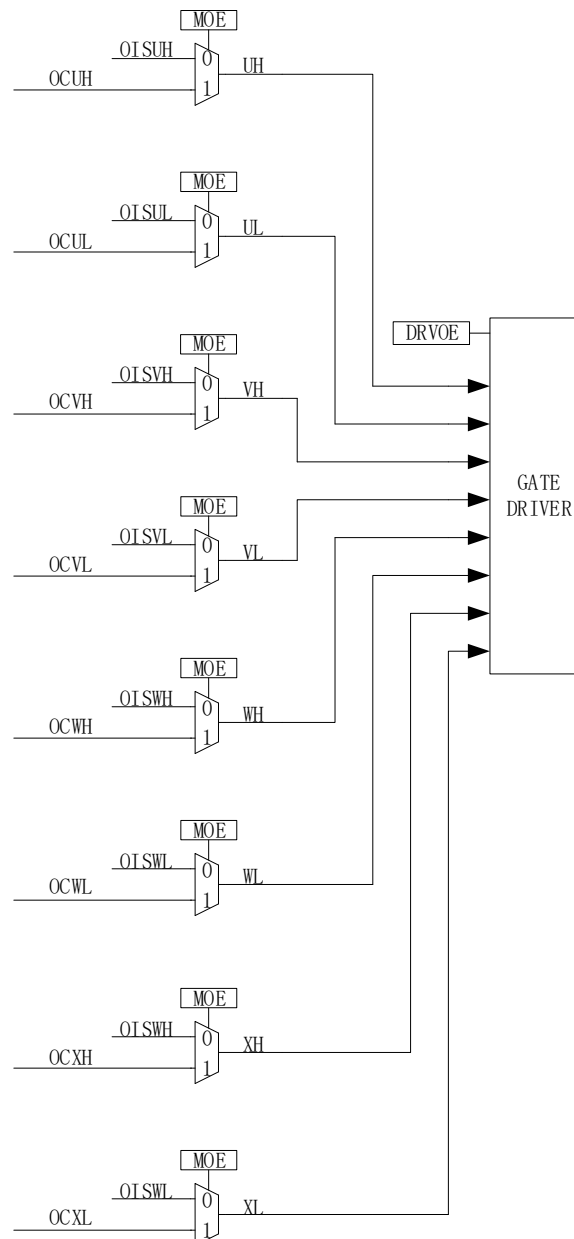


Figure 20-8 Block Diagram of Output Control Module

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

20.1.2.5 Interrupt

20.1.2.5.1 Compare Match Interrupt

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

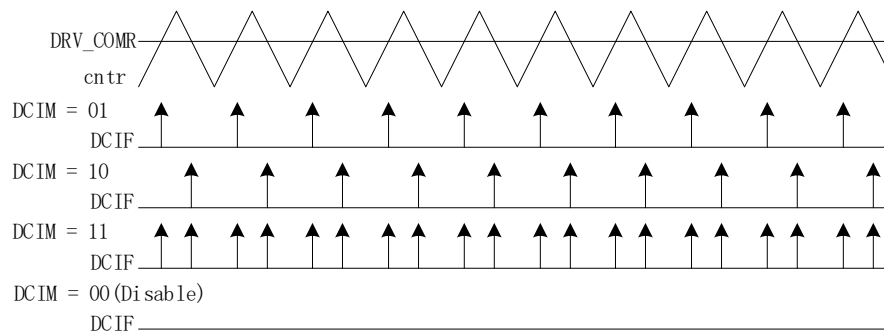
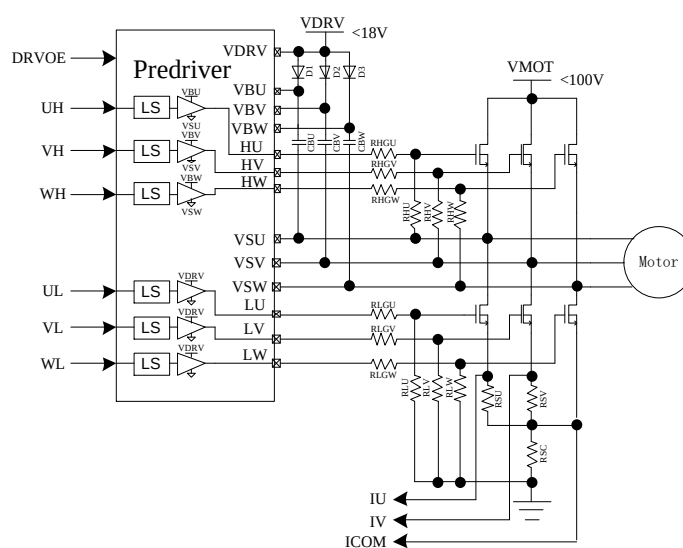


Figure 20-9 Driver Compare Match Interrupt

20.1.2.5.2 FG Interrupt

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



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

20.2 Driver Registers

20.2.1 DRV_CR (0x4062)

Bit	7	6	5	4	3	2	1	0
Name	SYSTIF	SYSTIE	FGIF	DCIF	FGIE	DCIP	DCIM	
Type	R/W0	R/W	R/W0	R/W0	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	SYSTIF	SYS TICK Interrupt Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[6]	SYSTIE	SYS TICK Interrupt Enable 0: Disable 1: Enable						
[5:0]		See DRV_SR(0x4061).						

20.2.2 DRV_SR(0x4061)

Bit	7	6	5	4	3	2	1	0
Name	SYSTIF	SYSTIE	FGIF	DCIF	FGIE	DCIP	DCIM	
Type	R/W0	R/W	R/W0	R/W0	R	R/W	R/W	R/W0
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	SYSTIF	SYS TICK Interrupt Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[6]	SYSTIE	SYS TICK Interrupt Enable 0: Disable 1: Enable						
[5]	FGIF	FG Interrupt Flag The interrupt flag is generated in every electrical cycle under FOC/square-wave control mode. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[4]	DCIF	Driver Match Interrupt Flag When the Driver count value is equal to DRV_COMR, the system decides whether to generate an interrupt according to DRV_SR[DCIM]. This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending						
[3]	FGIE	FG Interrupt Enable After the interrupt feature is enabled, an FG Interrupt is generated in each electrical cycle under FOC/square-wave control mode. 0: Disable 1: Enable						
[2]	DCIP	Number of PWM cycles to generate an interrupt 0: 1 PWM cycle 1: 2 PWM cycles						
[1:0]	DCIM	DRV Compare Match Interrupt Mode Selection When the Driver count value is equal to DRV_COMR, the system decides whether to generate an interrupt according to DRV_SR[DCIM].						

		00: No interrupt is generated. 01: An interrupt is generated when the timer counts up. 10: An interrupt is generated when the timer counts down. 11: An interrupt is generated when the timer counts up/down.
--	--	--

20.2.3 DRV_OUT (0xF8)

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

Bit	Name	Description
[7]	MOE	Main Output Enable This bit selects the sources for high and low sides of the bridge of phase-U/V/W/X output signals. It can be set to “1” and “0” by software. When bus current protection occurs, it is automatically cleared to “0” to turn off the output (see section 30.1.1.1). 0: Disable, with output sourced from the idle levels set by DRV_OUT[OISUH]/DRV_OUT[OISVH]/DRV_OUT[OISWH] and DRV_OUT[OISUL]/DRV_OUT[OISVL]/DRV_OUT[OISWL]. 1: Enable, with output sourced from the comparison value of the timer.
[6]	RSV	Reserved
[5]	OISWXL	Output idle level of WL/XL See descriptions on OISUH register
[4]	OISWXH	Output idle level of WH/XH See descriptions on OISUH register
[3]	OISVL	Output idle level of VL See descriptions on OISUH register
[2]	OISVH	Output idle level of VH See descriptions on OISUH register
[1]	OISUL	Output idle level of UL See descriptions on OISUH register
[0]	OISUH	Output idle level of UH This bit sets the UH output in idle state. When DRV_OUT[MOE] = 0, it outputs idle level to disable MOS. 0: Low 1: High

20.2.4 DRV_CMRR (0x405C, 0x405D)

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

DRV_CMRR(0x405C)								
Bit	15	14	13	12	11	10	9	8
Name	XHP	XLP	XHE	XLE	WHP	WLP	VHP	VLP
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

DRV_CMRL(0x405D)								
Bit	7	6	5	4	3	2	1	0
Name	UHP	ULP	WHE	WLE	VHE	VLE	UHE	ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15]	XHP	High-side Polarity Control of phase-X 0: Active High 1: Active Low

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

		Note: The high-side and low-side outputs of phase-U are complementary and deadtime is automatically added when ULE and UHE are set to “1”. Low-side output is the reference polarity.
[0]	ULE	Low-side Output of phase-U Enable 0: Disable 1: Enable Note: The high-side and low-side outputs of phase-U are complementary and deadtime is automatically added when ULE and UHE are set to “1”. Low-side output is the reference polarity.

20.2.5 DRV_ARR (0x405E, 0x405F)

DRV_ARR(0x405E)								
Bit	15	14	13	12	11	10	9	8
Name	DRV_ARR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Name	DRV_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	DRV_ARR	Timer reload value, which determines PWM frequency (center-aligned) Driver timer up-counts from 0 to DRV_ARR and an overflow event occurs. Then it down-counts to 0. Calculation formula: $f_{carrier} = f_{mcu}/2/(DRV_ARR)$ Range (0,4095)						

20.2.6 DRV_COMR (0x405A, 0x405B)

DRV_COMRH(0x405A)								
Bit	15	14	13	12	11	10	9	8
Name	DRV_COMR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DRV_COMRL(0x405B)								
Bit	7	6	5	4	3	2	1	0
Name	DRV_COMR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	DRV_COMR	Timer Compare Match Value The compare match interrupt is generated when the count value is equal to DRV_COMR and the conditions set in DRV_SR[DCIM] are met. Range (0, 4095)						

20.2.7 DRV_DR (0x4058, 0x4059)

DRV_DRH(0x4058)								
Bit	15	14	13	12	11	10	9	8
Name	DRV_DR[15:8]							
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DRV_DRL(0x4059)								
Bit	7	6	5	4	3	2	1	0
Name	DRV_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:0]	DRV_DR	PWM Duty Cycle Setting in Software When DRV_CR[OCS] = 0, DRV_CNTR is compared with DRV_DR to output PWM. "1" is output when DRV_CNTR is smaller than DRV_DR, and "0" is output when DRV_CNTR is larger than DRV_DR. Note: When this register is used as a comparison source, PWM is referenced to high side of the bridge and a deadtime is inserted in the complementary output of the low side of bridge. Range (0, 4095)						

20.2.8 DRV_DTR (0x4060)

Bit	7	6	5	4	3	2	1	0
Name	DRV_DTR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	DRV_DTR	Deadtime Setting This bit controls deadtime duration between complementary outputs. Assuming that MCU clock runs at 24MHz(41.67ns), deadtime = (DRV_DTR + 1) x 41.67ns Note: When DRV_DTR = 0, deadtime insertion is disabled.						

21 WDT

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

After being enabled, WDT starts counting from 0. When it reaches 0xFFFC, watchdog outputs a signal that is 4 internal slow clock cycles wide to reset MCU, and the program starts running from address 0. WDT has to be initialized at regular intervals during operation, and cannot reset the MCU.

21.1 WDT Notes

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

21.2 WDT Operations

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

21.3 WDT Registers

21.3.1 WDT_CR (0x4026)

Bit	7	6	5	4	3	2	1	0
Name	RSV							WDTRF
Type	-	-	-	-	-	-	-	R/W
Reset	-	-	-	-	-	-	-	0
Bit	Name	Description						
[7:1]	RSV	Reserved						
[0]	WDTRF	0: No effect 1: WDT is initialized.						

21.3.2 WDT_REL (0x4027)

Bit	7	6	5	4	3	2	1	0
Name	WDT_REL							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	WDT_REL	This bit sets 8 high-order bits of WDT after initialization.						

22 RTC and Clock Calibration

22.1 RTC Functional Block Diagram

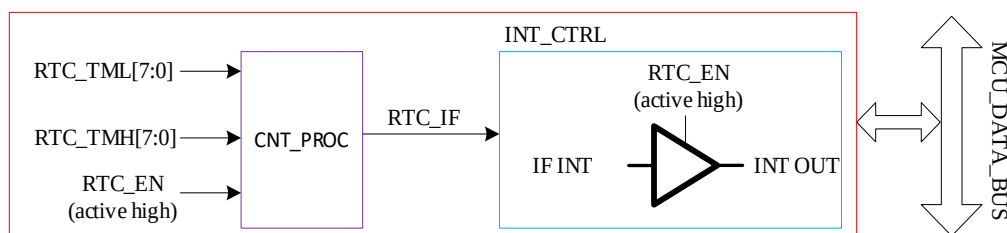


Figure 22-1 RTC Functional Block Diagram

22.2 RTC Operations

A write to RTC_TMH and RTC_TML sets RTC reload value.

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

22.3 RTC Registers

22.3.1 RTC_TM (0x402c,0x402d)

RTC_TMH (0x402C)								
Bit	15	14	13	12	11	10	9	8
Name	RTC0TM[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
RTC_TML (0x402D)								
Bit	7	6	5	4	3	2	1	0
Name	RTC0TM[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	Name	Description						
[15:0]	RTC_TM	RTC Count Register Write: RTC timer up-counts at a rate of 32768Hz from 0 to RTC_TM[15:0] and becomes overflowed . Meanwhile, an interrupt request is generated, causing the timer to be cleared and restart counting. Read: Instantaneous value of the timer						

22.3.2 RTC_STA (0x402e)

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

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

22.4 Clock Calibration

22.4.1 Introduction

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

Calibration operations: Set CAL_CR0[CAL_STA] = 1 in software to start the calibration. Read CAL_CR0[CAL_BUSY] flag bit to check if the calibration process is completed. When the calibration is completed (CAL_CR0[CAL_BUSY] = 0), the readout of CAL_CR0[CAL_ARR] is the value of the length of counting 8 slow clock cycles.

22.4.2 Clock Calibration Register

22.4.2.1 CAL_CR (0x4044, 0x4045)

CAL_CR0, CAL_CR1(0x4040, 0x4041)								
CAL_CR0(0x4040)								
Bit	15	14	13	12	11	10	9	8
Name	CAL_STA/ CAL_BUSY	RSV		CAL_ARR[12:8]				
Type	R/W	-	-	R	R/W	R/W	R/W	R/W
Reset	1	-	-	0	0	0	0	0
CAL_CR1(0x4041)								
Bit	7	6	5	4	3	2	1	0
Name	CAL_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15]	CAL_STA/ CAL_BUSY	Clock Calibration Enable Read: 0: Calibration is completed. 1: Calibration is in progress. Write: 0: No effect 1: Clock Calibration starts.						
[14:13]	RSV	Reserved						
[12:0]	CAL_ARR	Calibration Counts The count values of the fast clock to continuously count eight slow clock cycles When this value is 0, it indicates that no corresponding slow clock input exists, and when this value is 0x1FFF, it indicates that the count overflows (slow clock is too slow or fast clock is too fast).						

23 IO

23.1 IO Introduction

FU6361Q has up to 30 GPIO pins, including P0.0~P0.2, P0.4~P0.7, P1.0~P1.7, P2.0~P2.7 and P3.0~P3.6.

23.2 IO Operations

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

Notes:

- The enable bits of pull-up resistors and pull-down resistors are configured to “1”. See 23.3.9 P0_PU (0x4053) ~ 23.3.13 P4_PU (0x4057) for port pins and registers.
- See 5.4 GPIO Electrical Characteristics for the values of pull-up resistors and pull-down resistors.
- Pull-up resistors of P0.0~P0.2, P1.3~P1.6, P2.1 and P3.6~P3.7 are automatically disabled when the port pins are configured as analog mode.
- The relevant bits of P1_AN, P2_AN and P3_AN registers are configured to “1” to activate analog signal mode. See 23.3.6 P1_AN (0x4050) ~ 23.3.8 P3_AN (0x4052) for port pins and registers. After the port pins are configured to analog mode, all their digital features are disabled and the port state is 0 by reading relevant bits in P1, P2 and P3 registers.
- IO Priority:
 - GPIO has the lowest priority
 - P0.1: I²C > TIMER4 > GPIO
 - P0.5: SPI > UART > GPIO
 - P0.6: SPI > UART > GPIO
 - P0.7: TIMER2 > CMP > SPI > GPIO

23.3 IO Registers

23.3.1 P0_OE (0xFC)

Bit	7	6	5	4	3	2	1	0
Name	P0_OE							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	P0_OE		P0.0~P0.7 Digital I/O Selection 0: Input 1: Output					

23.3.2 P1_OE (0xFD)

Bit	7	6	5	4	3	2	1	0
Name	P1_OE							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	P1_OE		P1.0~P1.7 Digital I/O Selection 0: Input 1: Output					

23.3.3 P2_OE (0xFE)

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

23.3.4 P3_OE (0xFF)

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

23.3.5 P4_OE (0xE9)

Bit	7	6	5	4	3	2	1	0
Name	RSV					P4_OE[2:0]		
Type	-	-	-	-	-	R/W	R/W	R/W
Reset	-	-	-	-	-	0	0	0
Bit	Name		Description					
[7:3]	RSV		Reserved					
[2:0]	P4_OE		P4.0~P4.2 Digital I/O Selection 0: Input 1: Output					

23.3.6 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
Bit	Name		Description					
[7:4]	P1_AN		P1.7~P1.4 Analog Mode Enable 0: Disable 1: Enable					
[3]	HBMOD		P1.3 mode configuration, which determines the functional mode of P1.3 in combination with P1 OE.3.					
			HBMODE		P1 OE.3		P1.3 Pin Mode	
			0		0		Digital Input (DI)	
			0		1		Digital Output (DO)	
			1		0		Analog Input (AI)	
[2]	HDIO		1		1		Digital enhanced drive output mode. The drive mode of high level output provides enhanced drive, and the drive mode of low level output is the same as that of the digital output mode “01”.	
			IO Driver Capability for PWM Output. It is valid for L_DU, L_DV, L_DW, H_DU, H_DV and H_DW 0: Normal drive capability 1: High drive capability					
[1]	ODE1		P0.1 Collector Open-Drain Output Enable 0: Disable 1: Enable					
[0]	ODE0		P0.0 Collector Open-Drain Output Enable 0: Disable 1: Enable					

23.3.7 P2_AN (0x4051)

Bit	7	6	5	4	3	2	1	0
Name	P2_AN							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	P2_AN		P2.7~P2.0 Analog Mode Enable 0: Disable 1: Enable					

23.3.8 P3_AN (0x4052)

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

23.3.9 P0_PU (0x4053)

Bit	7	6	5	4	3	2	1	0
Name	P0_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	P0_PU		P0.7~P0.0 Pull-up Resistor Enable 0: Disable 1: Enable					

23.3.10 P1_PU (0x4054)

Bit	7	6	5	4	3	2	1	0
Name	P1_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	P1_PU		P1.7~P1.0 Pull-up Resistor Enable 0: Disable 1: Enable					

23.3.11 P2_PU (0x4055)

Bit	7	6	5	4	3	2	1	0
Name	P2_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	P2_PU		P2.7~P2.0 Pull-up Resistor Enable 0: Disable 1: Enable					

23.3.12 P3_PU (0x4056)

Bit	7	6	5	4	3	2	1	0
Name	P3_PU							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name		Description					
[7:0]	P3_PU		P3.7~P3.0 Pull-up Resistor Enable 0: Disable 1: Enable					

23.3.13 P4_PU (0x4057)

Bit	7	6	5	4	3	2	1	0
Name	RSV					P4_PU[2:0]		
Type	-	-	-	-	-	R/W	R/W	R/W
Reset	-	-	-	-	-	0	0	0
Bit	Name		Description					
[7:3]	RSV		Reserved					
[2:0]	P4_PU		P4.2~P4.0 Pull-up Resistor Enable 0: Disable 1: Enable					

23.3.14 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		Description					
[7]	SPITMOD		MISO port status after SPI slave device completes transmission 0: Output State 1: High-impedance State					
[6]	UARTEN		UART Enable 0: Disable 1: Enable					
[5]	UARTCH		UART Function Switching 0: No function switching with P0.6 serving as RXD and P0.5 as TXD. 1: Function switching, with P3.3 serving as RXD and P3.4 as TXD.					

[4]	T4SEL	Port Pins Multiplexed as Timer4 0: P0.1 is multiplexed as GPIO 1: P0.1 is multiplexed as Timer4 I/O pins. Note: I ² C has a higher priority than Timer4. When I ² C is enabled, P0.1 acts as SCL for I ² C communication.
[3]	T3SEL	Port Pins Multiplexed as Timer3 0: P1.1 is multiplexed as GPIO 1: P1.1 is multiplexed as Timer3 I/O pins.
[2]	T2SEL	Port Pins Multiplexed as Timer2 0: P1.1 is multiplexed as GPIO 1: P1.1 is multiplexed as Timer2 I/O pins.
[1]	T2SSEL	Port Pins Multiplexed as Timer2 Port 2 0: P0.7 is multiplexed as GPIO. 1: P0.7 is multiplexed as I/O pins of Timer2 port 2. Note: Timer2 has the highest priority, then the comparator output and SPI MISO.
[0]	XOE	X-phase H/L Side Enable 0: P4.2/P4.1 is multiplexed as GPIO 1: P4.2/P4.1 is multiplexed as X-phase high-side/low-side output pin respectively.

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

Port output register P0/1/2/3/4 supports read and write access. RMW commands are used to access the register value (see Table 23-2 for RMW commands), and other commands are used to access PORT pin.

Table 23-1 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

Register Name	Function Description	R/W	Initial Value
P0[7:0]	GPIO Register0	R/W	0x00
P1[7:0]	GPIO Register1	R/W	0x00
P2[7:0]	GPIO Register2	R/W	0x00
P3[7:0]	GPIO Register3	R/W	0x00
P4[2:0]	GPIO Register4	R/W	0x00

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

Table 23-2 RMW Commands

Command	Description
ANL	Bitwise logical AND operation
ORL	Bitwise logical OR operation
XRL	Bitwise logical XOR operation
JBC	Jump if the bit is set to “1” and then cleared to “0”
CPL	Bitwise logical converse operation
INC, DEC	+1, -1 logical operation
DJNZ	Jump if the bit is not “0”
MOV Px, y, C	Assign carry bit C to Px, y
CLR Px, y	Px, y is cleared to “0”
SETB Px, y	Px, y is set to “1”

24 ADC

24.1 ADC Block Diagram

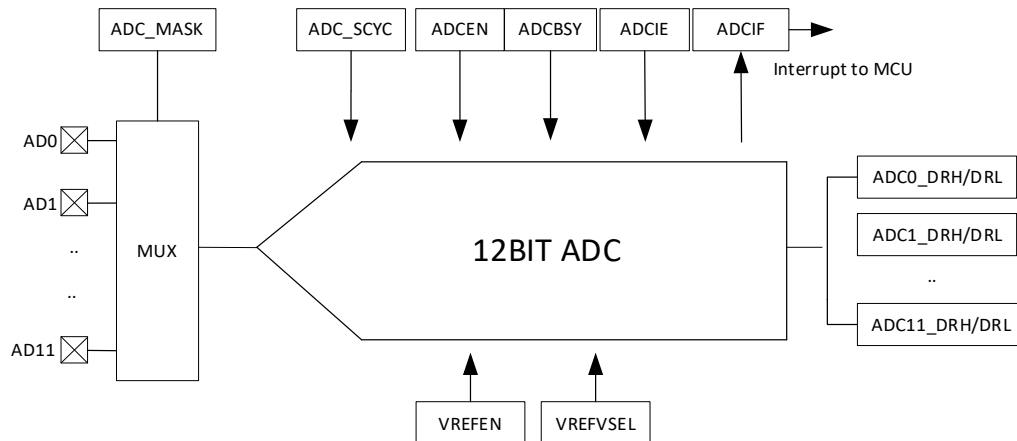


Figure 24-1 ADC Block Diagram

24.2 ADC Operations

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

24.2.1 Sequential Sampling Mode

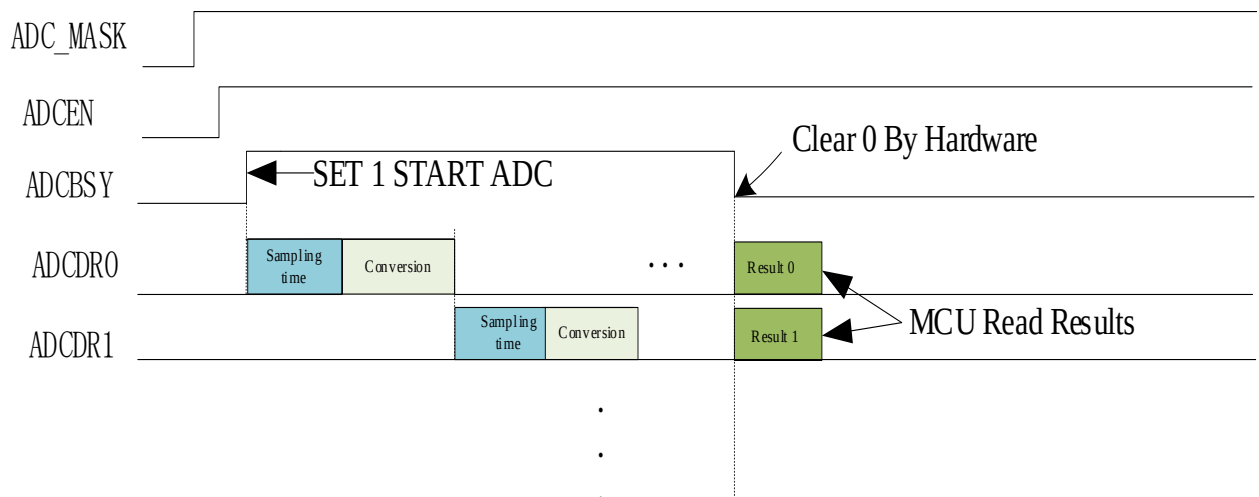


Figure 24-2 ADC Sequential Sampling Timing

ADC operations:

1. Set the appropriate ADC VREF;
2. Configure ADC_MASK to enable the corresponding channel required to sample;

3. Set sampling delay for each channel (minimum value is 3);
4. Configure `ADC_CR[ADCEN] = 1` to enable ADC;
5. Configure `ADC_CR[ADCBSY] = 1` to start ADC;
6. When `ADC_CR[ADCBSY] = 0`, ADC results are read by `ADCx_DR`.
7. The ADC conversion sequence is from low to high based on the enabled channel (i.e., when channel 2/3/4 is enabled, the signal is sampled in order of 2/3/4, and then a single conversion result is read after confirming `ADC_CR[ADCBSY] = 0`).

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

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

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

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

24.2.2 ADC Triggered Sampling Mode

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

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

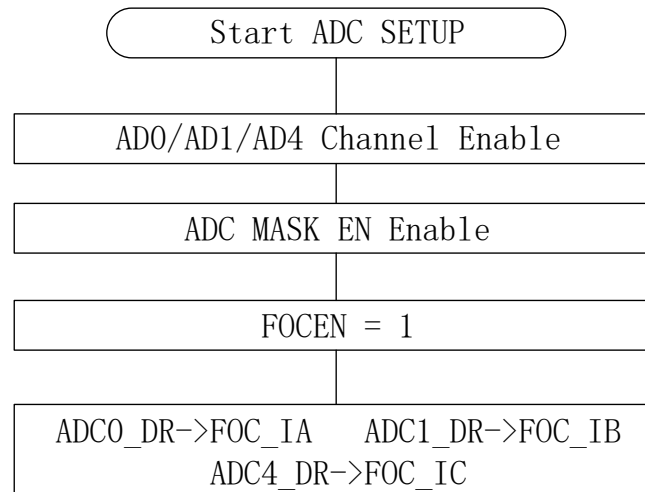


Figure 24-3 ADC Triggered Sampling Process

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

24.3 ADC Registers

24.3.1 ADC_CR (0x4039)

Bit	7	6	5	4	3	2	1	0
Name	ADCEN	ADCBSY	RSV			ADCALIGN	ADCIE	ADCIF
Type	R/W	R/W1	-	-	-	R/W	R/W	R/W
Reset	0	0	-	-	-	0	0	0
Bit	Name	Description						
[7]	ADCEN	ADC Enable 0: Disable 1: Enable						
[6]	ADCBSY	ADC Busy Flag ADC_CR[ADCEN] shall be configured first to enable the ADC module, and then ADC_CR[ADCBSY] for ADC conversion. A write of “1” to this bit starts ADC conversion. After ADC conversion, this bit is automatically cleared to “0” by hardware. MCU can read this bit to check the operating status of ADC conversion. When this bit is set to “1”, it has no effect to write “1” to this bit. A write of “0” to this bit has no effect. Writing “1” to this bit has no effect when ADC_MASK = 0.						
[5:3]	RSV	Reserved						
[2]	ADCALIGN	ADC Data Format Selection 0: ADC output is right-aligned 1: ADC output is left-second-high-aligned (invalid in triggered sampling mode)						
[1]	ADCIE	ADC Interrupt Enable (excluding triggered sampling mode interrupt) This bit determines whether ADC_CR[ADCIF] sends ADC interrupt to MCU. 0: Disable 1: Enable						
[0]	ADCIF	ADC Interrupt Flag After the ADC conversion is completed, if ADC_CR[ADCIE] = 1, an ADC interrupt is generated and sent to MCU. 0: The ADC conversion is not completed. 1: The ADC conversion is completed.						

24.3.2 ADC_MASK={ADC_MASKH,ADC_MASKL} (0x4036 ~ 0x4037)

ADC_MASKH(0x4036)								
Bit	15	14	13	12	11	10	9	8
Name	ADC_SCYC[11:8]				CH11EN	CH10EN	CH9EN	CH8EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	0	0
ADC_MASKL(0x4037)								
Bit	7	6	5	4	3	2	1	0
Name	CH7EN	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[11]	CH11EN	ADC Channel 11 Enable See ADC_MASK[CH0EN]						
[10]	CH10EN	ADC Channel 10 Enable See ADC_MASK[CH0EN]						
[9]	CH9EN	ADC Channel 9 Enable See ADC_MASK[CH0EN]						

[8]	CH8EN	ADC Channel 8 Enable See ADC_MASK[CH0EN]
[7]	CH7EN	ADC Channel 7 Enable See ADC_MASK[CH0EN]
[6]	CH6EN	ADC Channel 6 Enable See ADC_MASK[CH0EN]
[5]	CH5EN	ADC Channel 5 Enable See ADC_MASK[CH0EN]
[4]	CH4EN	ADC Channel 4 Enable See ADC_MASK[CH0EN]
[3]	CH3EN	ADC Channel 3 Enable See ADC_MASK[CH0EN]
[2]	CH2EN	ADC Channel 2 Enable See ADC_MASK[CH0EN]
[1]	CH1EN	ADC Channel 1 Enable See ADC_MASK[CH0EN]
[0]	CH0EN	ADC Channel 0 Enable 0: Disable 1: Enable

24.3.3 ADC_MASKH (0x4036)

ADC_MASKH(0x4036)								
Bit	7	6	5	4	3	2	1	0
Name	ADC_SCYCH[7:4]				CH11EN	CH10EN	CH9EN	CH8EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	0	0
Bit	Name		Description					
[7:4]	ADC_SCYCH[7:4]		ADC Sampling Cycle for ADC Channel 8, 9, 10 and 11 ADC_SCYCH[7] = 0: The sampling cycle is ADC_SCYCH[7:4] ADC clock cycles. ADC_SCYCH[7] = 1: The sampling cycle is (ADC_SCYC [7:4]*8 + 7) ADC clock cycles.					
[3]	CH11EN		ADC Channel 11 Enable					
[2]	CH10EN		ADC Channel 10 Enable					
[1]	CH9EN		ADC Channel 9 Enable					
[0]	CH8EN		ADC Channel 8 Enable					

24.3.4 ADC_SCYCL (0x4038)

ADC_SCYCL(0x4038)								
Bit	7	6	5	4	3	2	1	0
Name	ADC_SCYC[7:4]				ADC_SCYC[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	1
Bit	Name		Description					
[7:4]	ADC_SCYC[7:4]		ADC Sampling Cycle for ADC Channel 5, 6 and 7 ADC_SCYC[7] = 0: The sampling cycle is ADC_SCYC [6:4] ADC clock cycles. ADC_SCYC[7] = 1: The sampling cycle is (ADC_SCYC [6:4]*8 + 7) ADC clock cycles.					

[3:0]	ADC_SCYC[3:0]	ADC Sampling Cycle for ADC Channel 0, 1, 2, 3 and 4 ADC_SCYC[3] = 0: The sampling cycle is ADC_SCYC[2:0] ADC clock cycles. ADC_SCYC[3] = 1: The sampling cycle is (ADC_SCYC[2:0]*8 + 7) ADC clock cycles.
-------	---------------	---

24.3.5 ADC0_DR={ADC0_DRH,ADC0_DRL} (0x0300 ~ 0x0301)

ADC0_DRH (0x0300)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC0_DRL (0x0301)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 0 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 0 after ADC conversion						

24.3.6 ADC1_DR={ADC1_DRH,ADC1_DRL} (0x0302 ~ 0x0303)

ADC1_DRH(0x0302)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	R	R	R	R	R
Reset	-	-	-	0	0	0	0	0
ADC1_DRL(0x0303)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 1 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 1 after ADC conversion						

24.3.7 ADC2_DR={ADC2_DRH,ADC2_DRL} (0x0304 ~ 0x0305)

ADC2_DRH(0x0304)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC2_DRL(0x0305)								

Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 2 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 2 after ADC conversion						

24.3.8 ADC3_DR={ADC3_DRH,ADC3_DRL} (0x0306 ~ 0x0307)

ADC3_DRH(0x0306)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC3_DRL(0x0307)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 3 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 3 after ADC conversion						

24.3.9 ADC4_DR={ADC4_DRH,ADC4_DRL} (0x0308 ~ 0x0309)

ADC4_DRH(0x0308)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC4_DRL(0x0309)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 4 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 4 after ADC conversion						

24.3.10 ADC5_DR={ADC5_DRH,ADC5_DRL} (0x030A ~ 0x030B)

ADC5_DRH(0x030A)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC5_DRL(0x030B)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 5 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 5 after ADC conversion						

24.3.11 ADC6_DR={ADC6_DRH,ADC6_DRL} (0x030C ~ 0x030D)

ADC6_DRH(0x030C)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC6_DRL(0x030D)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 6 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 6 after ADC conversion						

24.3.12 ADC7_DR={ADC7_DRH,ADC7_DRL} (0x030E ~ 0x030F)

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

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

24.3.13 ADC8_DR={ADC8_DRH,ADC8_DRL} (0x0310 ~ 0x0311)

ADC8_DRH(0x0310)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC8_DRL(0x0311)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 8 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 8 after ADC conversion						

24.3.14 ADC9_DR={ADC9_DRH,ADC9_DRL} (0x0312 ~ 0x0313)

ADC9_DRH(0x0312)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC9_DRL(0x0313)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 9 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 9 after ADC conversion						

24.3.15 ADC10_DR={ADC10_DRH,ADC10_DRL} (0x0314 ~ 0x0315)

ADC10_DRH(0x0314)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC10_DRL(0x0315)								

Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 10 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 10 after ADC conversion						

24.3.16 ADC11_DR={ADC11_DRH,ADC11_DRL} (0x0316 ~ 0x0317)

ADC11_DRH(0x0316)								
Bit	15	14	13	12	11	10	9	8
Name	RSV				DH[11:8]			
Type	-	-	-	-	R	R	R	R
Reset	-	-	-	-	0	0	0	0
ADC11_DRL(0x0317)								
Bit	7	6	5	4	3	2	1	0
Name	DL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[15:12]	RSV	Reserved						
[11:8]	DH	4 high-order bits of the result by ADC channel 11 after ADC conversion						
[7:0]	DL	8 low-order bits of the result by ADC channel 11 after ADC conversion						

25 DAC

25.1 DAC Functional Block Diagram

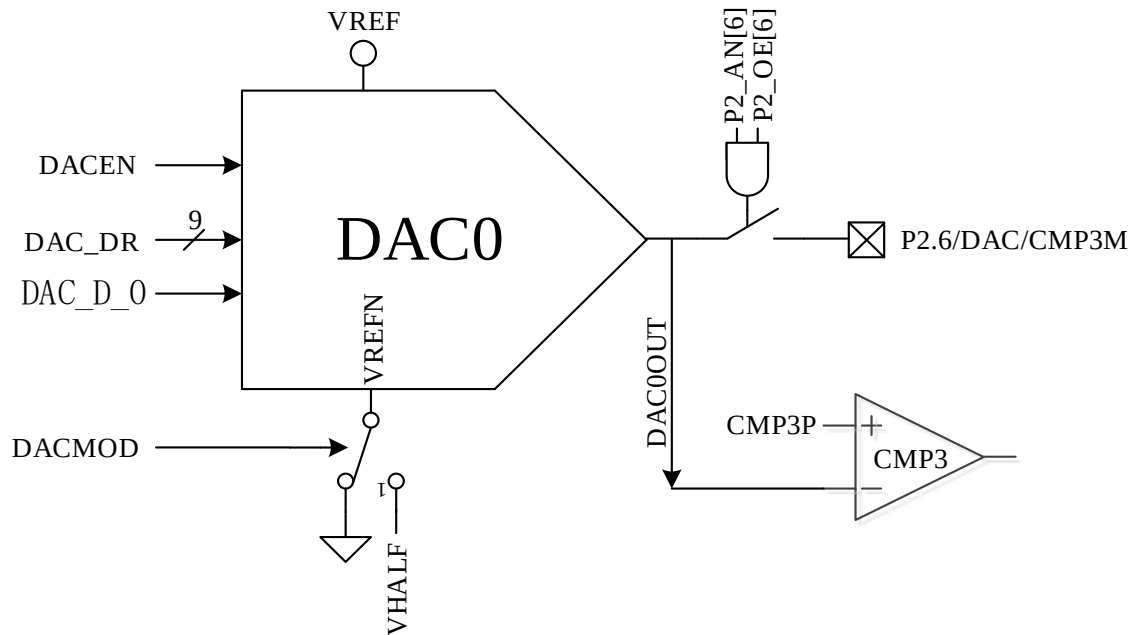


Figure 25-1 DAC Functional Block Diagram

Notes:

- DAC0 output has no current drive capability and can only carry capacitive load. To carry resistive load, operational amplifiers are used to follow the voltage output.
- Configure P2_AN[6] = 1 and P2_OE[6] = 1, and DAC0 output to P2.6/DAC pin;
- Configure VREF_CR[VREFEN] = 1 and DAC_CR[DACEN] = 1, and VREF is used as DAC0 reference voltage.

25.2 DAC Registers

25.2.1 DAC_CR (0x4035)

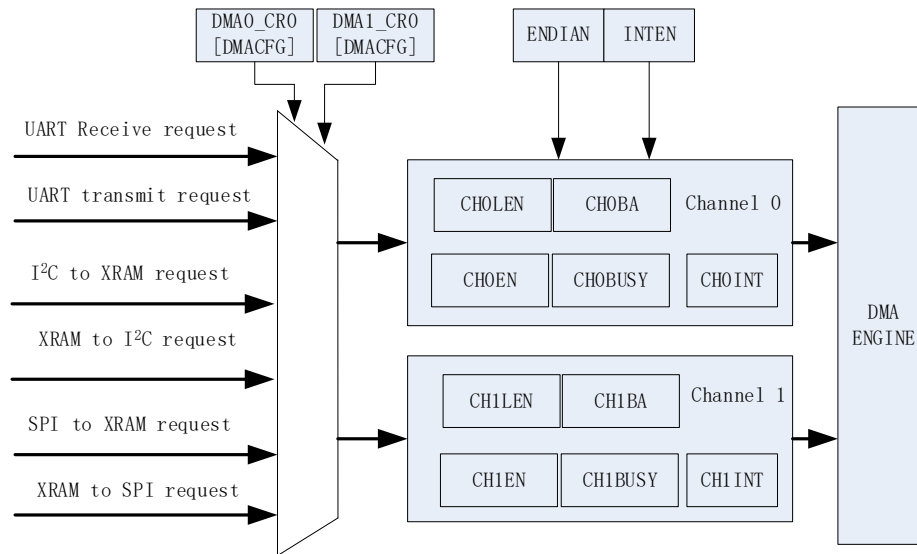
Bit	7	6	5	4	3	2	1	0
Name	DACEN	DACMOD	RSV					
Type	R/W	R/W	-	-	-	-	-	-
Reset	0	0	-	-	-	-	-	-
Bit	Name	Description						
[7]	DACEN	DAC Enable 0: Disable 1: Enable						
[6]	DACMOD	DAC Mode Setting 0: Full-voltage Output Mode. DAC output voltage ranges from 0 to VREF. 1: Half-voltage Output Mode. DAC output voltage ranges from VHALF to VREF.						
[5:0]	RSV	Reserved						

25.2.2 DAC_DR (0x404B)

Bit	7	6	5	4	3	2	1	0
Name	DAC_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	DAC_DR	Data Input of DAC Controller						

26 DMA

26.1 DMA Instructions



The DMA module is a dual-channel DMA controller, which performs direct data transfer between peripherals (SPI, UART or I²C) and XRAM. DMA accessing to XRAM does not interfere with the normal CPU read/write operation to XRAM. The length of the transferred data and the start address of XRAM access is configurable. Data transfer mode is configurable and interrupt can be enabled.

DMA instructions are as follows: Configure and enable the peripheral, and set input and output channels taken over by DMA by DMAx_CR0[DMACFG]; and then configure DMA interrupt enable, transfer order, transfer length and XRAM start address. Write “1” to DMAx_CR0[DMAEN] and DMAx_CR0[DMABSY] to start DMA. Clear the interrupt flag bit after data transfer, and set DMAx_CR0[DMABSY] to “1” to start DMA again.

26.2 DMA Registers

26.2.1 DMA0_CR0 (0x403A)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DMAIE	ENDIAN	DMAIF
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	DMAEN	DMA Channel 0 Enable 0: Disable 1: Enable						
[6]	DMABSY	DMA Channel 0 Start/Busy Flag Read: 0: Channel 0 is idle. 1: Channel 0 is busy. Write: 0: No effect 1: Channel 0 starts for data transfer						
[5:3]	DMACFG [2:0]	DMA Channel 0 Peripherals and Transfer Direction Selection 000: From UART to XRAM 001: From XRAM to UART 010: From I ² C to XRAM 011: From XRAM to I ² C 100: From SPI to XRAM 101: From XRAM to SPI Note: It cannot be configured when Channel 0 is busy.						
[2]	DMAIE	DMA Interrupt Request Enable 0: Disable 1: Enable. When the interrupt flag CH0INT or CH1INT is “1”, DMA module sends the interrupt request to MCU.						
[1]	ENDIAN	DMA Data Transfer Sequence 0: High bytes are received or sent first 1: Low bytes are received or sent first Note: This bit is set for 16-bit data mode, and shall be configured to “0” for 8-bit data mode. It cannot be configured when Channel 0 or 1 is busy.						
[0]	DMAIF	DMA Channel 0 Transfer Interrupt Event Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending (When DMA0_CR0[DMAIE] = 1, DMA module sends the interrupt request to MCU. Otherwise, only an interrupt flag is generated.)						

26.2.2 DMA1_CR0 (0x403B)

Bit	7	6	5	4	3	2	1	0
Name	DMAEN	DMABSY	DMACFG			DBGSW	DBGEN	DMAIF
Type	R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	DMAEN	DMA Channel 1 Enable 0: Disable 1: Enable						
[6]	DMABSY	DMA Channel 1 Start/Busy Flag Read: 0: Channel 1 is idle. 1: Channel 1 is busy.						

		Write: 0: No effect 1: Channel 1 starts for data transfer
[5:3]	DMACFG [2:0]	DMA Channel 1 Peripherals and Transfer Direction Selection 000: From UART to XRAM 001: From XRAM to UART 010: From I ² C to XRAM 011: From XRAM to I ² C 100: From SPI to XRAM 101: From XRAM to SPI Note: It cannot be configured when Channel 1 is busy.
[2]	DBGSW	Sector Targeted in Debug Mode 0: XSFR as the Debug area (export address space: 0x4020 ~ 0x40FF) 1: XRAM as the Debug area (export address space: 0x0000 ~ 0x0317)
[1]	DBGEN	Debug Mode Enable 0: Disable 1: Enable DMA module works in Debug mode when DMA1[DMACFG] is set to “101” and DMA1_CR0[DBGEN] to “1”. After SPI is enabled (SPI_CR1[SPIEN] = 1), DMA automatically sends relevant data in the sector defined by DMA1_CR0[DBGSW] via MOSI. CH1BA/CH1LEN defines the start address and range of the relevant data. NSS pin is automatically pulled low during data transmission. After each transmission, NSS pin is automatically pulled high. DMA Channel 1 Interrupt is automatically disabled in debug mode.
[0]	DMAIF	DMA Channel 1 Transfer Interrupt Event Flag This bit is set to “1” by hardware, and cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending (When DMA0_CR0[DMAIE] = 1, DMA module sends the interrupt request to MCU. Otherwise, only an interrupt flag is generated.)

26.2.3 DMA0_CR1 (0x403C)

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

26.2.4 DMA1_CR1 (0x403E)

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

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

27 VREF

27.1 VREF Instructions

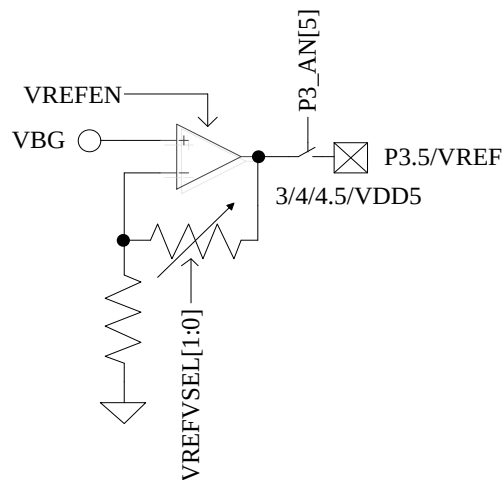


Figure 27-1 I/O Pins of VREF Module

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

VREF is enabled when VREF_VHALF_CR[VREFEN] is set to “1”. The output voltage is selected by configuring VREF_VHALF_CR[VREFVSEL]. See 27.2.1 VREF_VHALF_CR(XRAM: 0x404F) for details. When P3_AN[5] = 1, VREF is output to P3.5 pin.

27.2 VREF Register

27.2.1 VREF_VHALF_CR(XRAM: 0x404F)

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

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

28 VHALF

28.1 VHALF Instructions

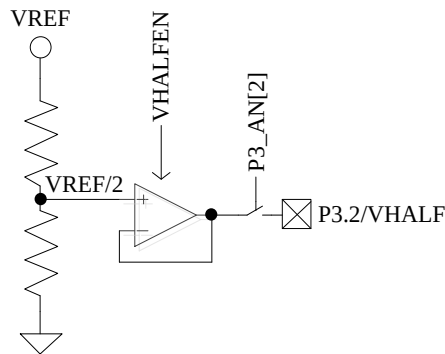


Figure 28-1 I/O Pins of VHALF Module

The input and output ports of VHALF module are shown in Figure 28-1. This module generates the voltage reference.

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

28.2 VHALF Register

See section 27.2.1.

29 Operational Amplifier

The chip integrates three high-speed independent operational amplifiers, AMP0, AMP1 and AMP2. They each contain a separate enable bit.

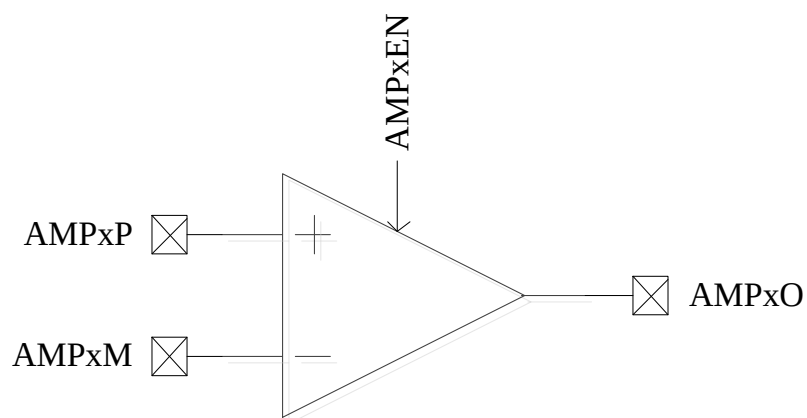


Figure 29-1 Schematic Diagram of Operational Amplifier Module

29.1 Operational Amplifier Instructions

29.1.1 Bus Current Sampling Operational Amplifier (AMP0)

Connection diagram of AMP0 is shown in Figure 29-2.

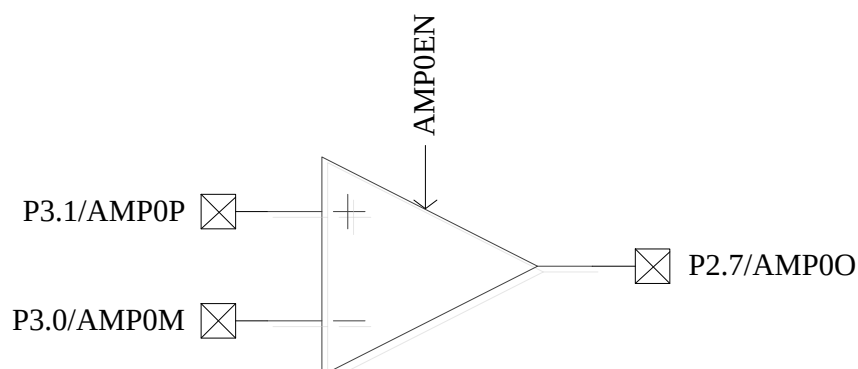


Figure 29-2 AMP0 I/O Pins

AMP0 is enabled when $\text{AMP_CR}[\text{AMP0EN}] = 1$.

The I/O pins of AMP0 are shown in Figure 29-2. Before AMP0 is enabled, P2.7, P3.0 and P3.1 shall be configured to analog signal mode, P2_AN[7] to “1” and P3_AN[1:0] to “11B”.

29.1.2 Phase Current Operational Amplifier (AMP1/AMP2)

29.1.2.1 AMP1

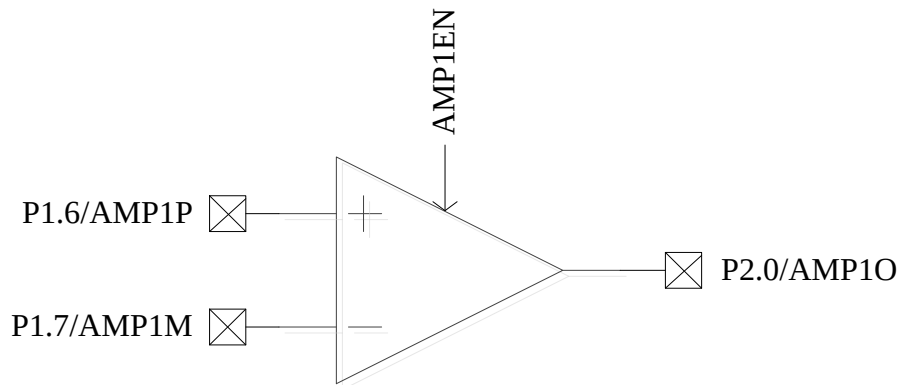


Figure 29-3 AMP1 I/O Pins

AMP1 is enabled when $\text{AMP_CR}[\text{AMP1EN}] = 1$.

The I/O pins of AMP1 are shown in Figure 29-3. Before AMP1 is enabled, P1.6, P1.7 and P2.0 shall be configured to analog signal mode, P1_AN[7:6] to “11B” and P2_AN[0] to “1B”.

29.1.2.2 AMP2

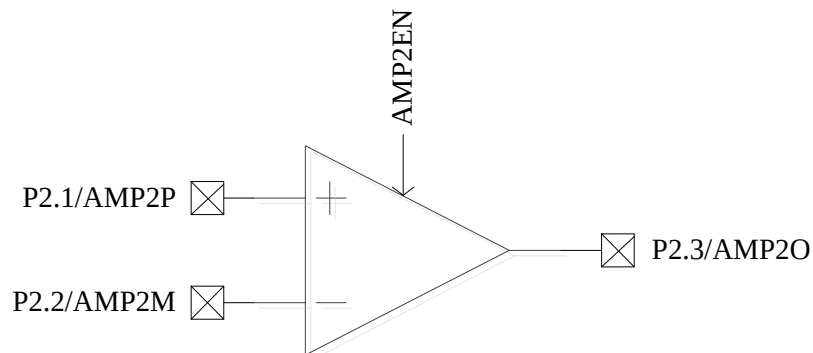


Figure 29-4 AMP2 I/O Pins

AMP2 is enabled when $\text{AMP_CR}[\text{AMP2EN}] = 1$.

The I/O pins of AMP2 are shown in Figure 29-4. Before AMP2 is enabled, P2.1, P2.2 and P2.3 shall be configured to analog signal mode and P2_AN[3:1] to “111B”.

29.1.3 PGA Differential Input Mode

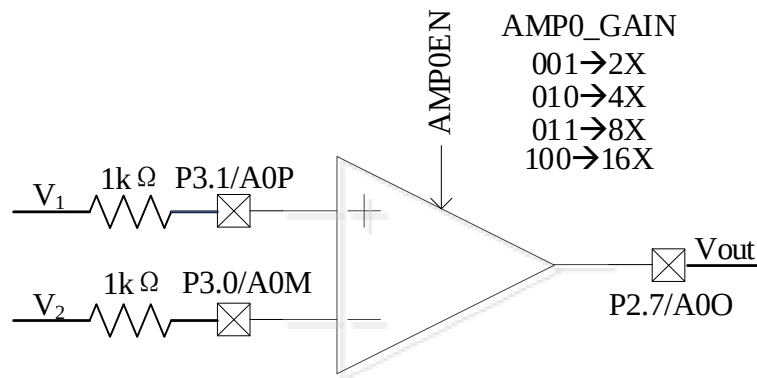


Figure 29-5 AMP0 Operating in PGA Differential Input Mode

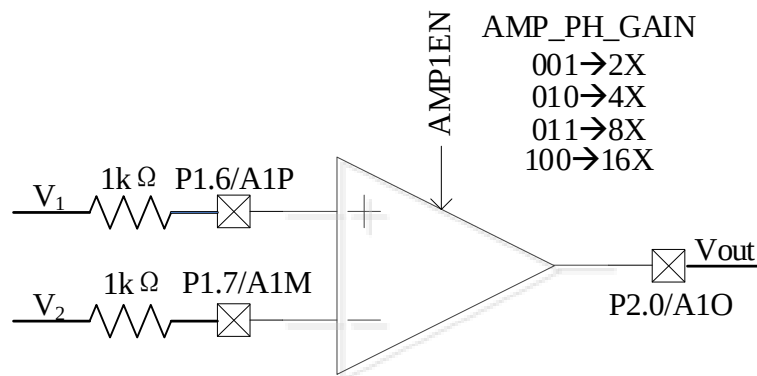


Figure 29-6 AMP1 Operating in PGA Differential Input Mode

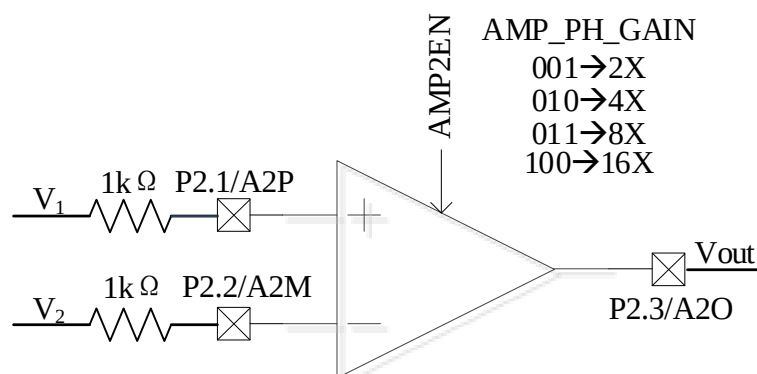


Figure 29-7 AMP2 Operating in PGA Differential Input Mode

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

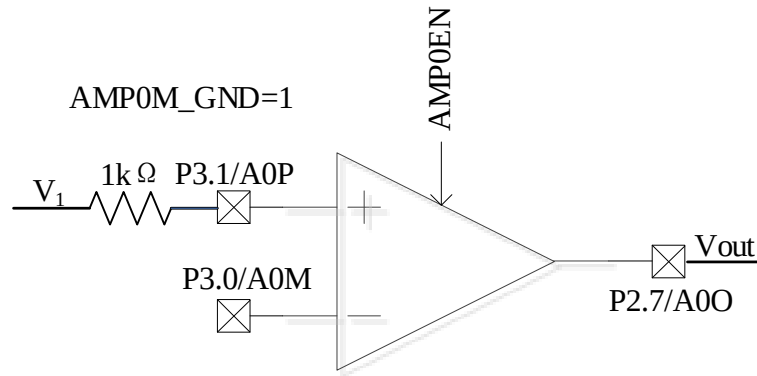


Figure 29-8 AMP0 Operating in PGA Single-ended Input Mode (Negative Input Connected to GND)

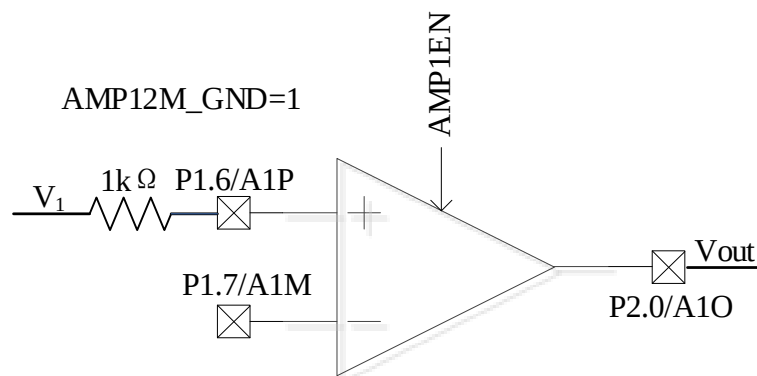


Figure 29-9 AMP1 Operating in PGA Single-ended Input Mode (Negative Input Connected to GND)

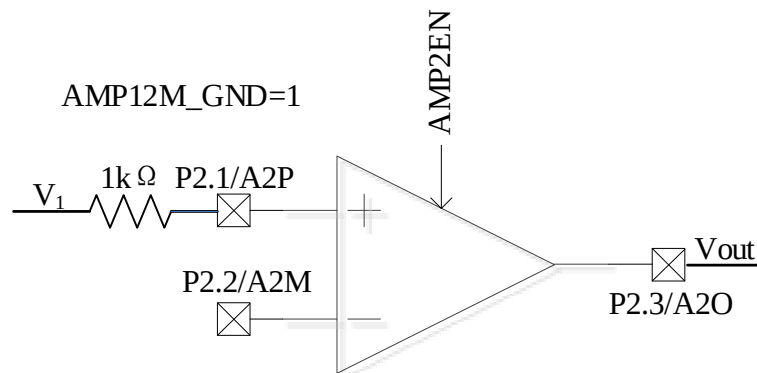


Figure 29-10 AMP2 Operating in PGA Single-ended Input Mode (Negative Input Connected to GND)

When PGA Single-ended Input Mode is selected, negative input of the operational amplifier shall be left floating. After AMP12M_GND/AMP0M_GND is configured in software, the negative input is connected to GND. In this case, V_{HALF} is equal to $25/64 \cdot V_{REF}$.

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

- When $AMP_GAIN = 2X$, $V_{out} = 7/6 \cdot V_{half} + 7/3 \cdot V1$

- When AMP_GAIN = 4X, $V_{out}=6/5 \cdot V_{half} + 24/5 \cdot V_1$
- When AMP_GAIN = 8X, $V_{out}=11/9 \cdot V_{half} + 88/9 \cdot V_1$
- When AMP_GAIN = 16X, $V_{out}=21/17 \cdot V_{half} + 336/17 \cdot V_1$

29.2 Operational Amplifier Registers

29.2.1 AMP_CR (0x404E)

Bit	7	6	5	4	3	2	1	0
Name	RSV					AMP2EN	AMP1EN	AMP0EN
Type	-	-	-	-	-	R/W	R/W	R/W
Reset	-	-	-	-	-	0	0	0

Bit	Name	Description
[7:3]	RSV	Reserved
[2]	AMP2EN	AMP2 Enable
[1]	AMP1EN	AMP1 Enable
[0]	AMP0EN	AMP0 Enable

29.2.2 CMP_AMP (0x40f2)

Bit	7	6	5	4	3	2	1	0
Name	DAC_D_0	AMP_PH_GAIN[2:0]			AMP0_GAIN[2:0]			CMP3P4M_FS
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	DAC_D_0	LSB of the 9-bit DAC data
[6:4]	AMP_PH_GAIN [2:0]	Amplification Gain Setting of AMP1&2 (with a 1kΩ external resistor) 000: The gain is configured by external circuit 001: 2X 010: 4X 011: 8X 100: 16X 101: Reserved 110/111: The gain is configured by external circuit
[3:1]	AMP0_GAIN [2:0]	Amplification Gain Setting of AMP0 (with a 1kΩ external resistor) 000: The gain is configured by external circuit 001: 2X 010: 4X 011: 8X 100: 16X 101: Reserved 110/111: The gain is configured by external circuit
[0]	CMP3P4M_FS	Function of CMP3 Positive End P2.7 and CMP4 Negative End Transferred to P3.4 0: Disable 1: Enable

29.2.3 TSD_ADJ (0x40f3)

TSD_ADJ(0x40f3)								
Bit	7	6	5	4	3	2	1	0
Name	RSV				AMP12M_GND	AMP0M_GND	TSDADJ3	TSDADJ0_0
Type	-	-	-	-	R/W	R/W	R/W	R/W
Reset	-	-	-	-	0	0	0	0
Bit	Name				Description			
[7:4]	RSV				Reserved			
[3]	AMP12M_GND				After this bit is enabled, negative input of AMP1&2 (P1.7/P2.2) is connected to GND internally in PGA input mode, and $V_{HALF} = 25/64 * V_{REF}$. 0: Disable 1: Enable			
[2]	AMP0M_GND				After this bit is enabled, negative input of AMP0 (P3.0) is connected to GND internally in PGA input mode, and $V_{HALF} = 25/64 * V_{REF}$. 0: Disable 1: Enable			
[1:0]	TSDADJ3/ TSDADJ0_0				It works with EVT_FILT. TSDADJ3 are the most significant bits, EVT_FILT[6:5] are the medium significant bits and TSDADJ0_0 are the least significant bits.			
					TSD_ADJ[3:0]	Temperature (°C)		
					1000	65		
					1001	70		
					1010	75		
					1011	80		
					1100	86		
					1101	91		
					1110	97		
					1111	103		
					0000	105		
					0001	115		
					0010	120		
					0011	128		
					0100	135		
					0101	142		
					0110	150		
					0111	Reserved		

30 Comparator

30.1 Comparator Introduction

30.1.1 Comparator CMP3

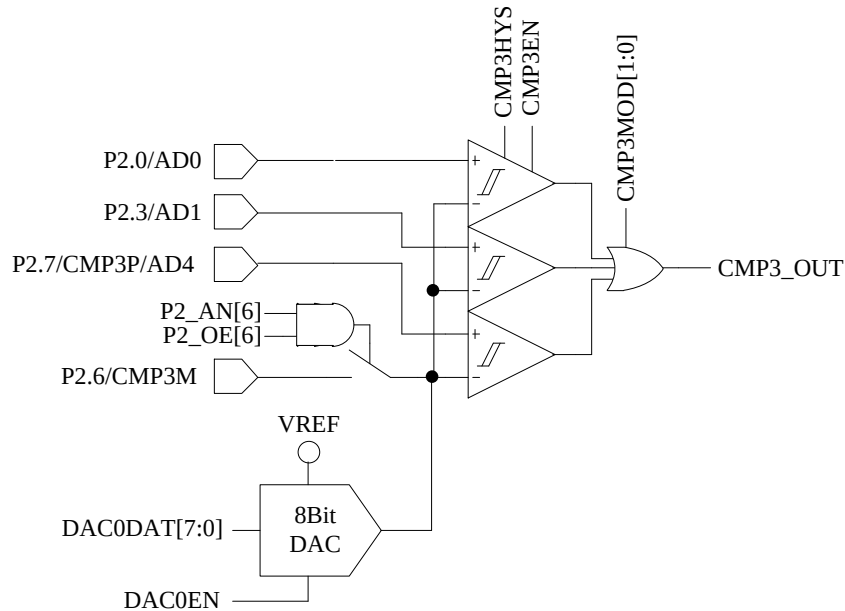


Figure 30-1 CMP3 I/O Pins

CMP3 configurations are as follows:

1. Set P2_AN[6] and P2_OE[6] to “1” to enable CMP3 and VREF on the negative input. The VREF source can be on-chip DAC0 output voltage or external circuit input voltage. Select DAC0 output, and place an external capacitor between P2.6 pin and GND (the recommended capacitance value is 100pF, and the output voltage stabilizes after DAC0 output for a period of time);
2. Configure CMP_CR1[CMP3MOD] to select single-comparator input, dual-comparator input or triple-comparator input mode;
3. Configure CMP_CR1[CMP3HYS] to enable or disable hysteresis;
4. Set CMP_CR1[CMP3EN] = 1 to enable CMP3.

The I/O pins of CMP3 are shown in Figure 30-1.

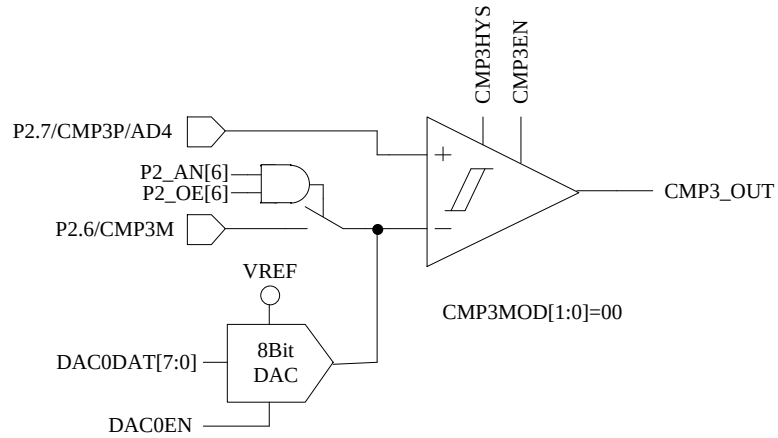


Figure 30-2 Single-comparator Input Mode (CMP_CR1[CMP3MOD] = 00B)

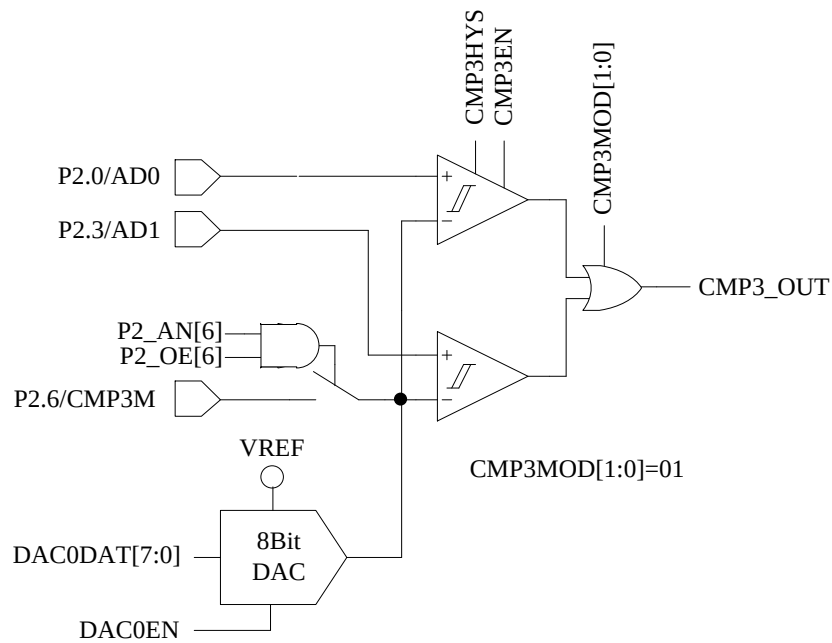


Figure 30-3 Dual-comparator Input Mode (CMP_CR1[CMP3MOD] = 01B)

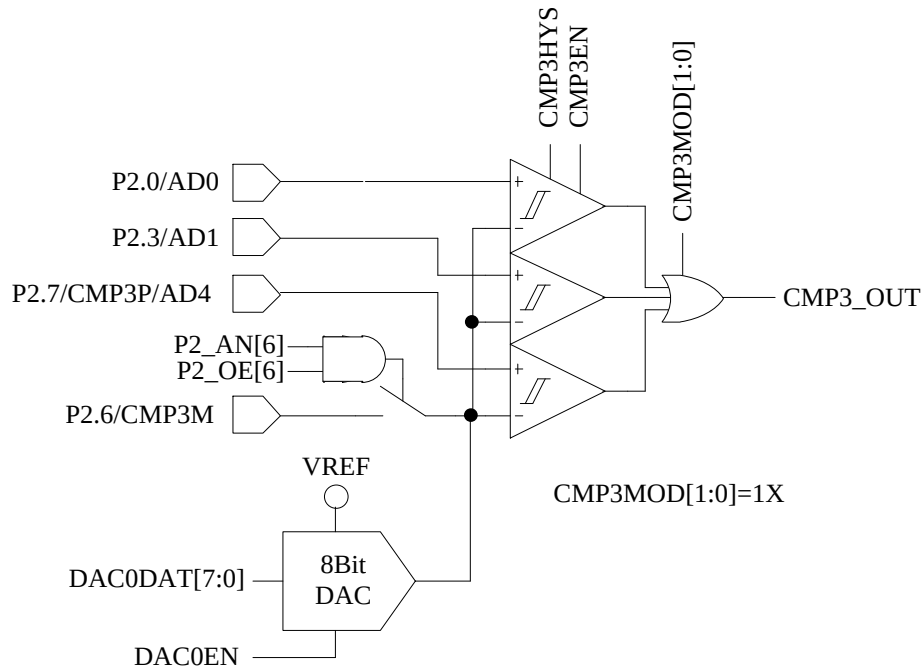


Figure 30-4 Triple-comparator Input Mode (CMP_CR1[CMP3MOD] = 1XB)

30.1.1.1 Overcurrent Protection (OCP)

When an overcurrent protection signal is generated, DRV_OUT[MOE] is automatically cleared to output idle voltage to stop motor drive for chip and motor protection. OCP feature is enabled when EVT_FILT[MOEMD] = 01, which automatically turns off the output and generates an OCP interrupt request if the current exceeds the threshold. When EVT_FILT[MOEMD]=00, the output is not automatically turned off if the current exceeds the threshold. However, an OCP interrupt is reported.

The source of OCP interrupt is selected by configuring EVT_FILT[EFSRC], namely CMP3 interrupt or external interrupt INT0. When EVT_FILT[EFSRC] = 1, TCON[IT0] bit is programmed to select the trigger edge of the external interrupt INT0 which generates an OCP output. At this time, the source of OCP interrupt is INT0. When EVT_FILT[EFSRC] = 0, the OCP output is generated by CMP3. At this time, the source of OCP interrupt is CMP3.

Configuring EVT_FILT[EFDIV] != 0 enables the filtering of interrupt signals for OCP, and programming EVT_FILT[EFDIV] = 01/10/11 selects filter width of 4/8/16 clock cycles. When the filtering feature is enabled, the filtered signal is delayed by 4~5/8~9/16~17 clock cycles compared to the signal before filtering.

When CMP3 works for bus current protection, the internal module can directly turn off motor output to achieve overcurrent protection.

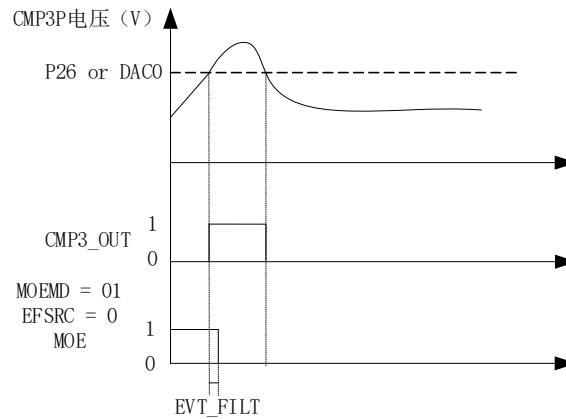


Figure 30-5 MOE Disabled upon Bus Overcurrent Protection

When $\text{CMP_CR0}[\text{CMP3IM}] = 11$, the OCP output is generated on the rising edge of CMP3. If $\text{EVT_FILT}[\text{MOEMD}] = 01$, the output is automatically turned off when a protection event occurs and a protection interrupt is reported. If $\text{EVT_FILT}[\text{EFSRC}] = 0$, the bus current protection signal is generated by CMP3, and sampled voltage on the bus is compared to generate the protection signal. Setting $\text{EVT_FILT}[\text{EFDIV}] = 01/10/11$ selects filter width of 4/8/16 clock cycles. When the filtering feature is enabled, the filtered signal is delayed by $4 \sim 5/8 \sim 9/16 \sim 17$ clock cycles compared to the signal before filtering. As shown in Figure 30-5, when voltage on positive input of the comparator is higher than that on negative input, CMP3_OUT1 is set to “1” to generate CMP3 comparison interrupt for filtering (based on $\text{EVT_FILT}[\text{EFDIV}]$). MOE ($\text{DRV_OUT}[\text{MOE}]$) is automatically cleared to “0” to turn off six-way outputs to implement overcurrent protection.

30.1.1.2 Cycle-by-cycle Current Limiting

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

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

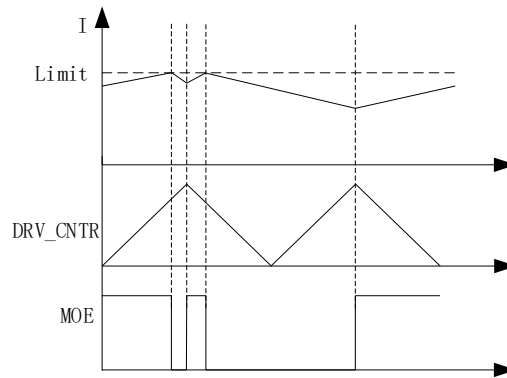


Figure 30-6 Cycle-by-cycle Current Limiting Waveform at EVT_FILT[MOEMD] = 10

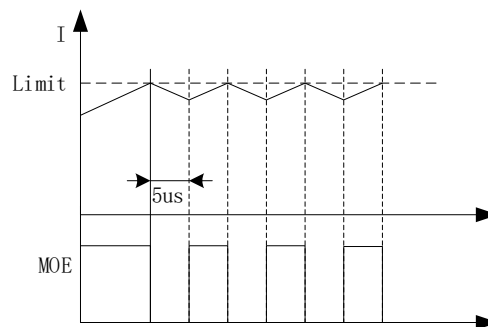


Figure 30-7 Cycle-by-cycle Current Limiting Waveform at EVT_FILT[MOEMD] = 11

30.1.2 Comparator CMP4

CMP4 is a hysteresis comparator, as shown in Figure 30-8. CMP4OUT can be read by software or reversed on external interrupt INT0. CMP_CR3[CMPSEL] is configured to select CMP4 for signal output. When CMP4 is enabled, CMP3MOD[1:0] cannot be set as 01B. CMP4 generally works with CMP3 implement cycle-by- cycle current limiting in square wave control mode.

CMP4 configurations are as follows:

1. Configure P2_AN[3] and P2_AN[7] to “1” to assign P2.3/CMP4P and P2.7/CMP4M pins to analog signal;
2. Configure CMP_CR2[CMP4EN] = 1 to enable CMP4;
3. Set LVSR[EXT0CFG] = 111 to select CMP4 as the source of INT0;
4. Clear INT0 flag bit to enable INT0;
5. External interrupt INT0 is triggered when CMP4OUT is reversed (from 1 to 0).

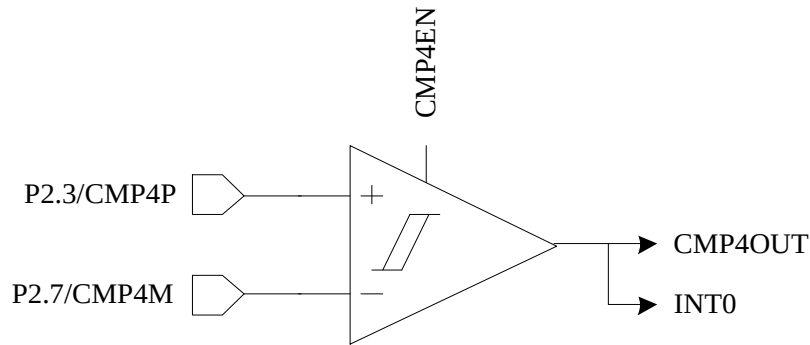


Figure 30-8 Schematic Diagram of CMP4 Module

30.1.3 Comparator CMP0

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

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

When CMP_CR2[CMP0MOD] = 00B, CMP0 works in the mode of three comparators without built-in resistor. The I/O pins are shown in Figure 30-9. It is used for BEMF detection with the external virtual neutral point resistors. The negative inputs of the comparators are connected together to P1.5/CMP0M pin, and the positive inputs are connected to P1.4/CMP0P, P1.6/CMP1P and P2.1/CMP2P respectively. The outputs are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.

Configurations are as follows:

1. Set TIM1_CR3[T1TIS] = 01B to select the comparator as the input;
2. Configure CMP_CR2[CMP0MOD] = 00B to select the mode of three comparators without built-in resistor;
3. Set P1_AN[6:4] = 111B and P2_AN[1] = 1 to configure the port pins as analog mode;
4. Set P1_PU[5:4] = 00B and Reset as 00B. Skip this step if no modification is required;
5. Configure CMP_CR1[CMP0HYS] select the hysteresis voltage and Reset as 000B;
6. Configure CMP_CR2[CMP0EN] = 1 to enable the comparator;
7. Configure CMP_CR2[CMP0SEL] to select one-way or multiple-way output(s). See descriptions on CMP_CR2[CMP0SEL] in section 30.2.3;
8. The outputs results are sent to registers CMP2OUT ~ CMP0OUT.

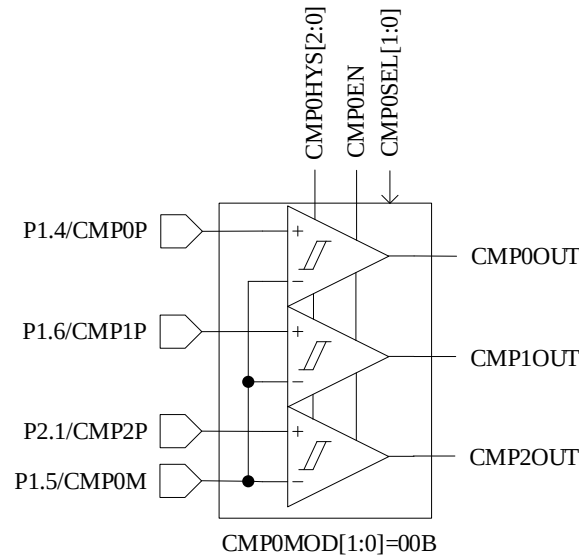


Figure 30-9 CMP0 with Built-in Three Comparators

(without Built-in Resistor)

When $\text{CMP_CR2}[\text{CMP0MOD}] = 01\text{B}$, CMP0 works in the mode of three comparators with built-in resistor. The I/O pins are shown in Figure 30-10. It is used for BEMF detection with the internal virtual neutral point resistors. The negative inputs of the comparators are connected the center point of the built-in resistor, and the positive inputs are connected to P1.4/CMP0P, P1.6/CMP1P and P2.1/CMP2P respectively. The outputs are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.

Configurations are as follows:

1. Set $\text{TIM1_CR3}[\text{T1TIS}] = 01\text{B}$ to select the comparator as the input;
2. Configure $\text{CMP_CR2}[\text{CMP0MOD}] = 01\text{B}$ to select the mode of three comparators with built-in resistor;
3. Set $\text{P1_AN}[6] = 1$, $\text{P1_AN}[4] = 1$ and $\text{P2_AN}[1] = 1$ to configure the port pins as analog mode;
4. Set $\text{P1_PU}[4] = 0$ and Reset as 0. Skip this step if no modification is required;
5. Configure $\text{CMP_CR1}[\text{CMP0HYS}]$ select the hysteresis voltage and Reset as 000B;
6. Configure $\text{CMP_CR2}[\text{CMP0EN}] = 1$ to enable the comparator;
7. Configure $\text{CMP_CR2}[\text{CMP0SEL}]$ to select one-way or multiple-way output(s). See descriptions on $\text{CMP_CR2}[\text{CMP0SEL}]$ in section 30.2.3;
8. The outputs results are sent to registers $\text{CMP2OUT} \sim \text{CMP0OUT}$.

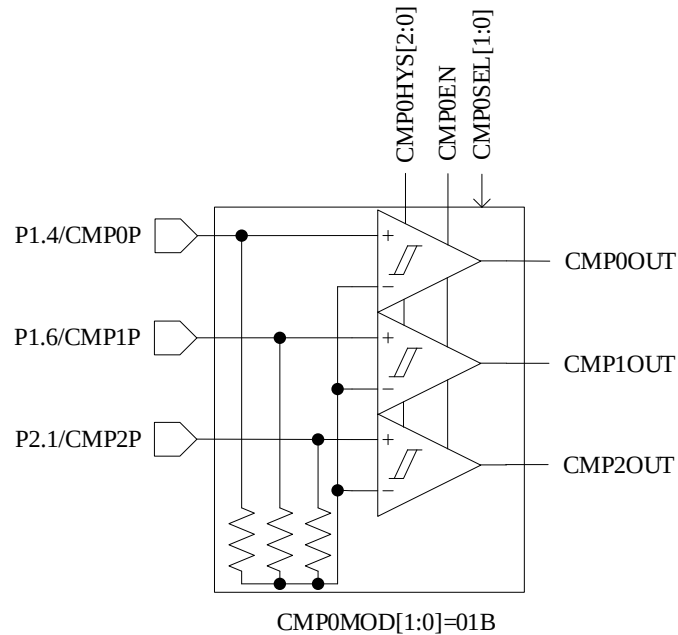


Figure 30-10 CMP0 with Built-in Three Comparators and Resistor

When $\text{CMP_CR2}[\text{CMP0MOD}] = 10\text{B}$, CMP0 works in triple-differential-comparator mode, where the motor rotor position is detected by differential Hall signals. The input and output pins are shown in Figure 30-11. The negative inputs of the three comparators are respectively connected to P1.5/CMP0M, P1.7/CMP1M and P2.1/CMP2M, and the positive inputs are respectively connected to P1.4/CMP0P, P1.6/CMP1P and P2.1/CMP2P. The outputs are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.

Configurations are as follows:

1. Set $\text{TIM1_CR3}[\text{T1TIS}] = 01\text{B}$ to select the comparator as the input;
2. Configure $\text{CMP_CR2}[\text{CMP0MOD}] = 10\text{B}$ to select triple-differential-comparator mode;
3. Set $\text{P1_AN}[7:4] = 1111\text{B}$ and $\text{P2_AN}[2:1] = 11\text{B}$ to configure the port pins as analog mode;
4. Set $\text{P1_PU}[4] = 0$ and Reset as 0. Skip this step if no modification is required;
5. Configure $\text{CMP_CR1}[\text{CMP0HYS}]$ select the hysteresis voltage and Reset as 000B;
6. Configure $\text{CMP_CR2}[\text{CMP0EN}] = 1$ to enable the comparator;
7. Configure $\text{CMP_CR2}[\text{CMP0SEL}]$ to select one-way or multiple-way output(s). See descriptions on $\text{CMP_CR2}[\text{CMP0SEL}]$ in section 30.2.3;
8. The outputs results are sent to registers CMP2OUT ~ CMP0OUT.

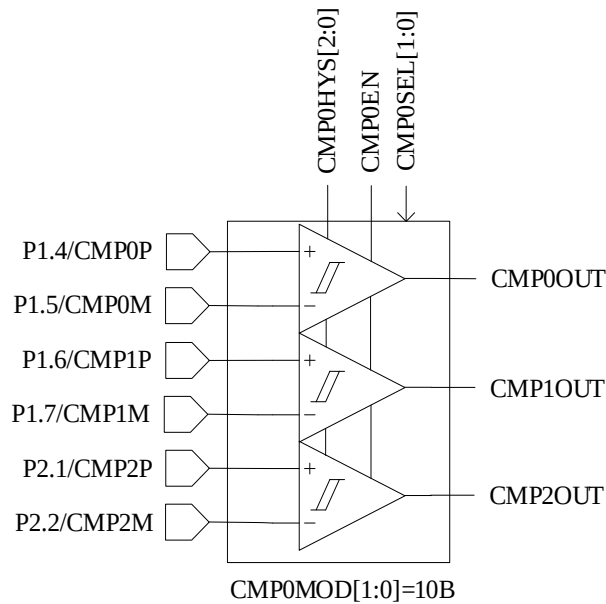


Figure 30-11 CMP0 in Triple-differential-comparator Mode

When $\text{CMP_CR2}[\text{CMP0MOD}] = 11\text{B}$, CMP0 works in dual-comparator mode. The I/O pins are shown in Figure 30-12. The negative inputs of the two comparators are connected together to P1.5/CMP0M, and the positive inputs are connected to P1.4/CMP0P and P1.3/CMP1PS respectively. The outputs are transferred to CMP0OUT and CMP1OUT respectively.

Configurations are as follows:

1. Set $\text{TIM1_CR3}[\text{T1TIS}] = 01\text{B}$ to select the comparator as the input;
2. Configure $\text{CMP_CR2}[\text{CMP0MOD}] = 11\text{B}$ to select dual-comparator mode;
3. Set $\text{P1_AN}[5:3] = 111\text{B}$ and $\text{P1_OE}[3] = 0\text{B}$ to configure the port pins as analog mode;
4. Pull-up resistor of P1[5:3] can be enabled ($\text{P1_PU}[5:3] = 111\text{B}$) or disabled ($\text{P1_PU}[5:3] = 000\text{B}$) as required, and set Reset as 0. Skip this step if no modification is required. In dual-comparator mode, pull-up resistor of P1[5:3] is generally disabled and used for special applications.
5. Configure $\text{CMP_CR1}[\text{CMP0HYS}]$ select the hysteresis voltage and Reset as 000B;
6. Configure $\text{CMP_CR2}[\text{CMP0EN}] = 1$ to enable the comparator;
7. Configure $\text{CMP_CR2}[\text{CMP0SEL}]$ to select one-way or multiple-way output(s). See descriptions on $\text{CMP_CR2}[\text{CMP0SEL}]$ in section 30.2.3;
8. The outputs results are sent to registers CMP1OUT and CMP0OUT.

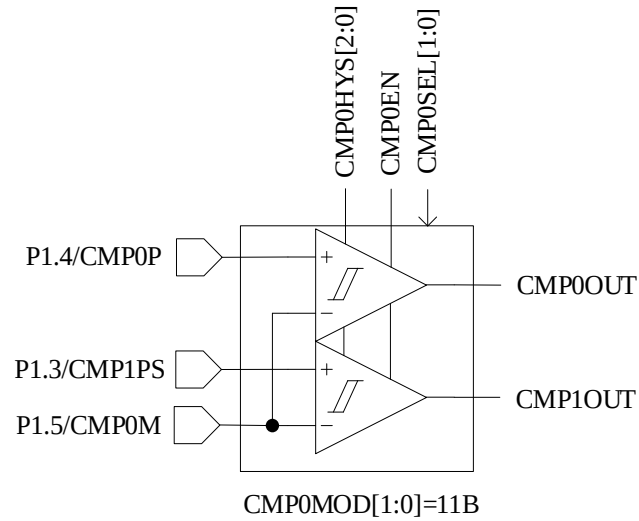


Figure 30-12 CMP0 in Dual-comparator Mode

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

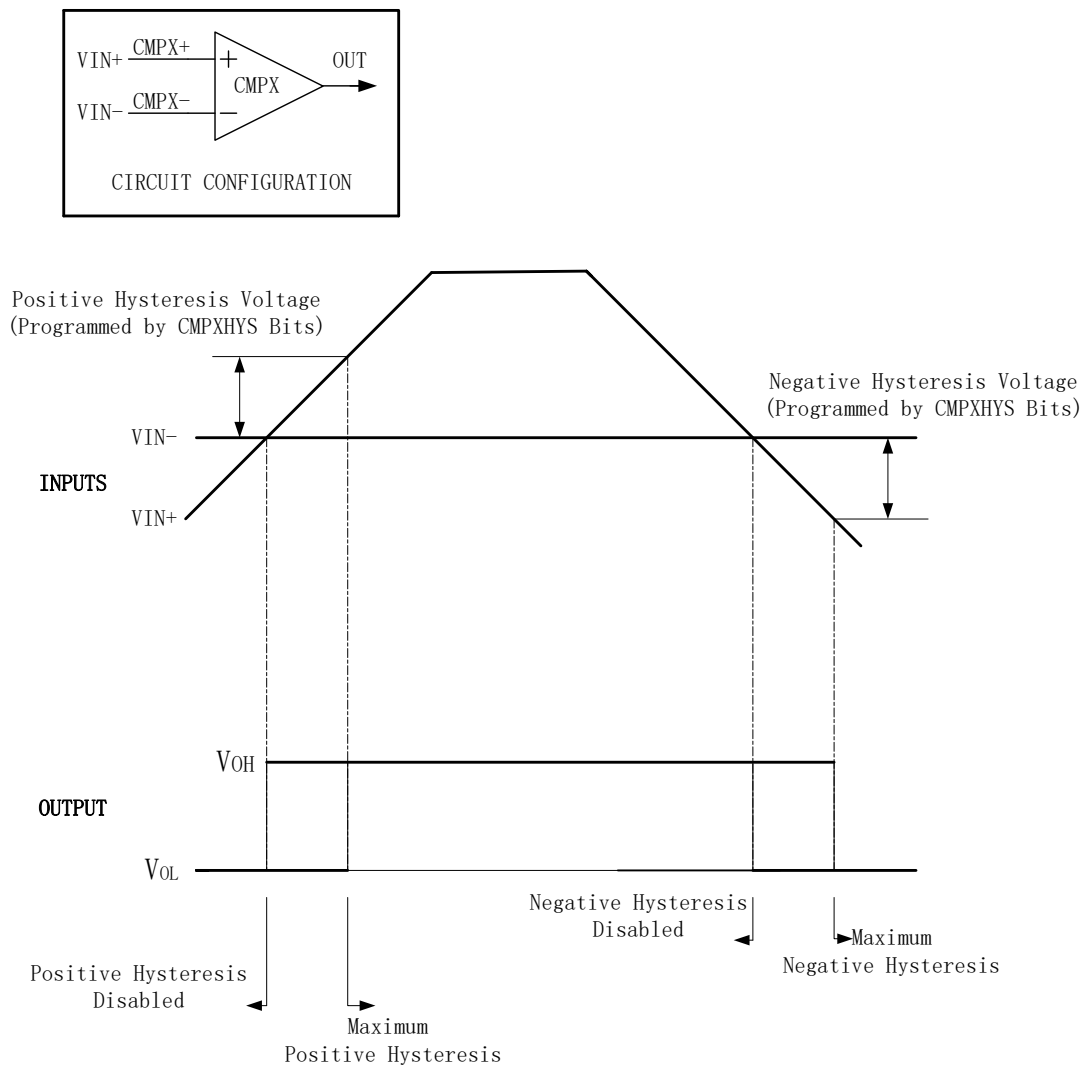


Figure 30-13 CMP0 Input and Output under Positive and Negative Hysteresis Voltage

30.1.4 Comparator Sampling

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

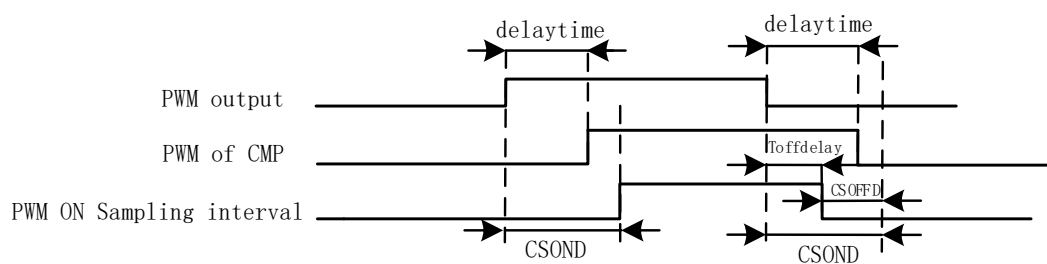


Figure 30-14 PWM ON Sampling Mode

There is a delay from the PWM output to the output of the comparator, which is mainly affected by the following factors: resistance value of drive resistor, MOS on-off speed, and input delay and hysteresis settings of the comparator. As shown in Figure 30-15, the delay time is from the chip output to the comparator output. When high-level sampling is performed, the sampling interval shall be enveloped by actual high-level output of the comparator. First, the sampling ON-delayed time $CMP_SAMR[CSOND]$ is set to overcome output delay and ringing due to MOS on/off. At the end of the sampling interval, $CMP_SAMR[CSOND]$ is delayed after the falling edge of PWM, at which time the actual sampling window has exceeded the corresponding high-level interval (PWM of CMP). The sampling OFF-lead time $CMP_SAMR[CSOFFD]$ is set to stop sampling $Toffdelay$ after PWM output falling edge, where $Toffdelay = CMP_SAMR[CSOND] - CMP_SAMR[CSOFFD]$.

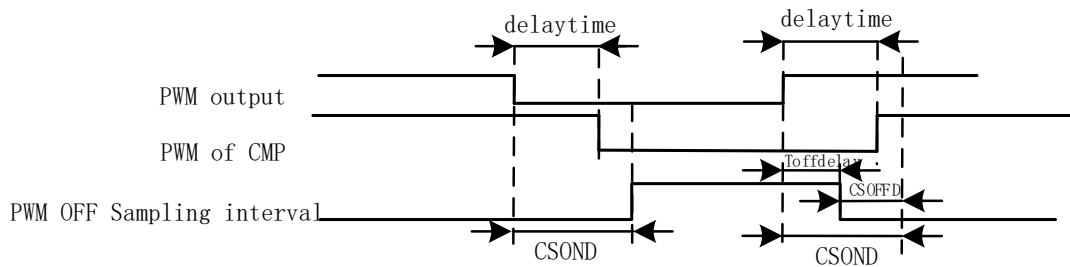


Figure 30-15 PWM OFF Sampling Mode

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

Method for measuring the delay of PWM output to comparator: Set $CMP_CR3[SAMSEL] = 00$ to disable the comparator sampling delay feature. Set $CMP_CR3[CMPSSEL]$ to select the corresponding comparator output. Enable the PWM output and comparator, manually rotate the motor to change the comparator value, and measure the delay between the PWM output and the comparator output.

30.1.5 Comparator Output

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

30.2 Comparator Registers

30.2.1 CMP_CR0 (0xD5)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IM		CMP2IM		CMP1IM		CMP0IM	
Type	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:6]	CMP3IM	CMP3 Interrupt Mode See descriptions on CMP_CR0[CMP0IM].						
[5:4]	CMP2IM	CMP2 Interrupt Mode See descriptions on CMP_CR0[CMP0IM].						
[3:2]	CMP1IM	CMP1 Interrupt Mode See descriptions on CMP_CR0[CMP0IM].						
[1:0]	CMP0IM	CMP0 Interrupt Mode 00: No interrupt is generated. 01: An interrupt is generated upon rising edge. 10: An interrupt is generated upon falling edge. 11: An interrupt is generated upon both rising/falling edges.						

30.2.2 CMP_CR1 (0xD6)

Bit	7	6	5	4	3	2	1	0
Name	HALLSEL	CMP3MOD		CMP3EN	CMP3HYS	CMP0HYS		
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	HALLSEL	Hall Input Selection 0: P0.2/P3.7/P3.6 1: P1.4/P1.6/P2.1						
[6:5]	CMP3MOD	Positive Input of CMP3. See Figure 30-1 for details. The negative input is connected to P2.6 or DAC output. 00: Single-comparator mode, where P2.7 is connected to the positive input, as shown in Figure 30-2. 01: Dual-comparator mode, where P2.0 and P2.3 are connected to the positive input, as shown in Figure 30-3. 1X: Triple-comparator mode, where P2.0, P2.3 and P2.7 are connected to the positive input, as shown in Figure 30-4.						
[4]	CMP3EN	CMP3 Enable 0: Disable 1: Enable						
[3]	CMP3HYS	CMP3 Hysteresis Voltage Selection 0: No hysteresis 1: Hysteresis voltage is selected						
[2:0]	CMP0HYS	CMP0 Hysteresis Voltage Selection: 000: No hysteresis 001: $\pm 2.5\text{mV}$ 010: -5mV 011: $+5\text{mV}$ 100: $\pm 5\text{mV}$ 101: -10mV 110: $+10\text{mV}$ 111: $\pm 10\text{mV}$						

30.2.3 CMP_CR2 (0xDA)

Bit	7	6	5	4	3	2	1	0
Name	CMP4EN	CMP0MOD		CMP0SEL		RSV		CMP0EN
Type	R/W	R/W		R/W	R/W	-	-	R/W
Reset	0	0	0	0	0	-	-	0
Bit	Name	Description						
[7]	CMP4EN	CMP4 Enable 0: Disable 1: Enable						
[6:5]	CMP0MOD	CMP0/1/2 Mode Setting 00: CMP0 with built-in three comparators (without built-in resistor), as shown in Figure 30-9. 01: CMP0 with built-in three comparators and resistors, as shown in Figure 30-10. 10: CMP0 in triple-differential-comparator mode, as shown in Figure 30-11 11: CMP0 in dual-comparator mode, where only CMP0 and CMP1 work, as shown in Figure 30-12.						
[4:3]	CMP0SEL	CMP0 Pin Combination Selection, used with CMP_CR2[CMP0MOD] bit. It is set to 00 by default. In square-wave drive application, TIM1_DBRx[T1CPE] automatically controls CMP_CR2[CMP0SEL] to enable or disable each comparator.						
		CMP0MOD	CMP0SEL	Function Description				
		00	00	CMP0/1/2 work simultaneously. The negative input of these comparators are connected to CMP0M. The hardware automatically compares the positive inputs CMP0P, CMP1P and CMP2P with CMP0M, and the output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.				
			01	Only CMP0 works. The positive input is connected to CMP0P, and the negative input to CMP0M. The output results are transferred to CMP0OUT.				
			10	Only CMP1 works. The positive input is connected to CMP1P, and the negative input to CMP0M. The output results are transferred to CMP1OUT.				
			11	Only CMP2 works. The positive input is connected to CMP2P, and the negative input to CMP0M. The output results are transferred to CMP2OUT.				
		01	00	CMP0/1/2 work simultaneously. The negative inputs of these comparators are connected to the center of built-in resistor. The hardware automatically compares the positive inputs CMP0P, CMP1P and CMP2P with CMP0M. The output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.				
			01	Only CMP0 works. The positive input is connected to CMP0P, and the negative input to the center of BEMF built-in resistor. The output results are transferred to CMP0OUT.				

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

30.2.4 CMP_CR3 (0xDC)

Bit	7	6	5	4	3	2	1	0
Name	CMPDTEN	DBGSEL		SAMSEL		CMPSEL		
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	CMPDTEN	Comparator Deadtime Sampling Enable 0: Disable 1: Enable						
[6:5]	DBGSEL	Debug Output Selection, connected to P0.1 pin 00: Debug Output Disable 01: Freewheeling shielding is completed and ZCP signal is detected 10: ADC Trigger Signal 11: Comparator Sampling Interval						
[4:3]	SAMSEL	Sampling delay enable of CMP0, CMP1, CMP2 and ADC in PWM ON/OFF modes 00: Sampling at both PWM ON and OFF modes without time delay 01: Sampling at PWM OFF mode, with time delay according to CMP_SAMR 10: Sampling at PWM ON mode, with time delay according to CMP_SAMR 11: Sampling at both PWM ON and OFF, with time delay according to CMP_SAMR						
[2:0]	CMPSEL	Comparator Output Selection Signals of one selected comparator is output to the pin. See Comparator Debugging for details. 000: No output 001: CMP0 010: CMP1 011: CMP2 100: CMP3 101: CMP4 110: Reserved 111: ADC result comparison						

30.2.5 CMP_SAMR(0x40AD)

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

[3:0]	CSOFFD	CMP0/CMP1/CMP2 OFF-delayed Sampling Time When PWM module switches from OFF to ON or from ON to OFF, turn-on/off of the power IC affects input signal of the comparator. In this case, CMP_SAMR[CSOFFD] is configured to reduce comparator interference. If MCU clock runs at 24MHz(41.67ns) OFF-delayed Sampling Time = CSOFFD x 41.67 x 8ns Notes: ■ CMP_SAMR[CSOND] must be greater than or equal to CMP_SAMR[CSOFFD]. ■ See section Sampling for square-wave control and section RSD Comparator Sampling for RSD.
-------	--------	--

30.2.6 CMP_SR (0xD7)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IF	CMP2IF	CMP1IF	CMP0IF	CMP3OUT	CMP2OUT	CMP1OUT	CMP0OUT
Type	R/W0	R/W0	R/W0	R/W0	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	CMP3IF	CMP3 Interrupt Flag This bit is set to “1” when CMP3 interrupt is generated. It is cleared to “0” by software. 0: No Interrupt Pending 1: Interrupt Pending
[6]	CMP2IF	CMP2 Interrupt Flag This bit is set to “1” when CMP2 interrupt is generated. It is cleared to “0” by software. TIM1_CR3[T1TIS] must be configured to “01b”, otherwise, it is the Hall interrupt flag. 0: No Interrupt Pending 1: Interrupt Pending
[5]	CMP1IF	CMP1 Interrupt Flag This bit is set to “1” when CMP1 interrupt is generated. It is cleared to “0” by software. TIM1_CR3[T1TIS] must be configured to “01b”, otherwise, it is the Hall interrupt flag. 0: No Interrupt Pending 1: Interrupt Pending
[4]	CMP0IF	CMP0 Interrupt Flag This bit is set to “1” when CMP0 interrupt is generated. It is cleared to “0” by software. TIM1_CR3[T1TIS] must be configured to “01b”, otherwise, it is the Hall interrupt flag. 0: No Interrupt Pending 1: Interrupt Pending
[3]	CMP3OUT	CMP3 Comparison Result 0: CMP3 comparison result is 0. 1: CMP3 comparison result is 1.
[2]	CMP2OUT	CMP2 Comparison Result TIM1_CR3[T1TIS] must be configured to “01”, otherwise, the results are generated by Hall signals. 0: CMP2 comparison result is 0. 1: CMP2 comparison result is 1.
[1]	CMP1OUT	CMP1 Comparison Result TIM1_CR3[T1TIS] must be configured to “01”, otherwise, the results are generated by Hall signals. 0: CMP1 comparison result is 0. 1: CMP1 comparison result is 1.

[0]	CMP0OUT	CMP0 comparison result TIM1_CR3[T1TIS] must be configured to “01”, otherwise, the results are generated by Hall signals. 0: CMP0 comparison result is 0. 1: CMP0 comparison result is 1.
-----	---------	---

30.2.7 EVT_FILT (0xD9)

Bit	7	6	5	4	3	2	1	0
Name	TSDEN	TSDADJ		MOEMD		EFSRC	EFDIV	
Type	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	0	1	1	0	0	0	0	0
Bit	Name	Description						
[7]	TSDEN	TSD Enable 0: Disable 1: Enable						
[6:5]	TSDADJ	TSD Temperature Selection. See section 29.2.3.						
[4:3]	MOEMD	MOE Cleared and Enabled by Hardware MOE is cleared and enabled by hardware upon bus over-/under-current protection event. 00: MOE is not automatically cleared. 01: MOE is automatically cleared. 10: MOE is automatically cleared and enabled by hardware upon DRV timer overflow events (for square-wave drive). 11: MOE is automatically cleared and enabled automatically upon DRV timer overflow/underflow events or every 5 μ s (for square-wave drive).						
[2]	EFSRC	Input Source of Filtering Module Upon Bus Current Protection Event 0: CMP3 interrupt 1: INT0 interrupt						
[1:0]	EFDIV	Filter Width for Bus Current Protection 00: Not to filter 01: 4 system clock cycles 10: 8 system clock cycles 11: 16 system clock cycles						

31 Power Supply

31.1 LDO Introduction

31.1.1 LDO Operations

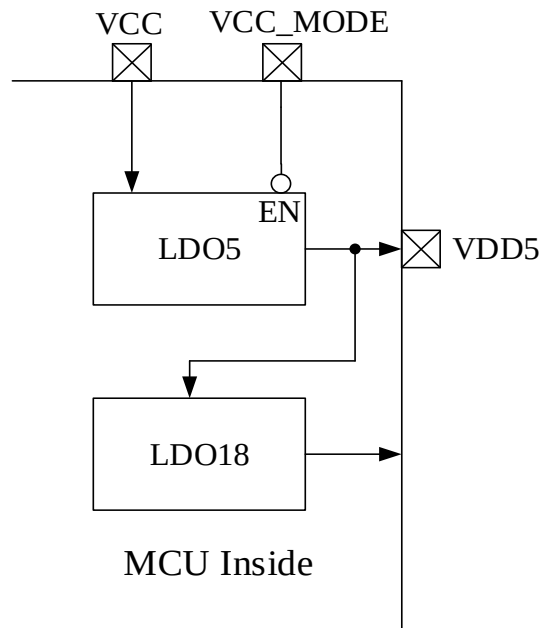


Figure 31-1 Functional Block Diagram of Power Supply

The I/O pins of LDO module is shown in Figure 31-1. The LDO module converts the input supply voltage to 5V VDD5 as the power supply for built-in analog modules. Internal LDO5 or external supply for VDD5 is selected by configuring VCC_MODE.

Single-power Supply Mode: VCC_MODE=0 and VCC= 5~24V. See Figure 31-2 for details.

Dual-power Supply Mode: VCC_MODE=1, VCC≥VDD5, VCC= 5~36V and VDD5=5V. See Figure 31-3 for details.

Note: When VCC_MODE = 1, VDD5 is selected as the power supply.

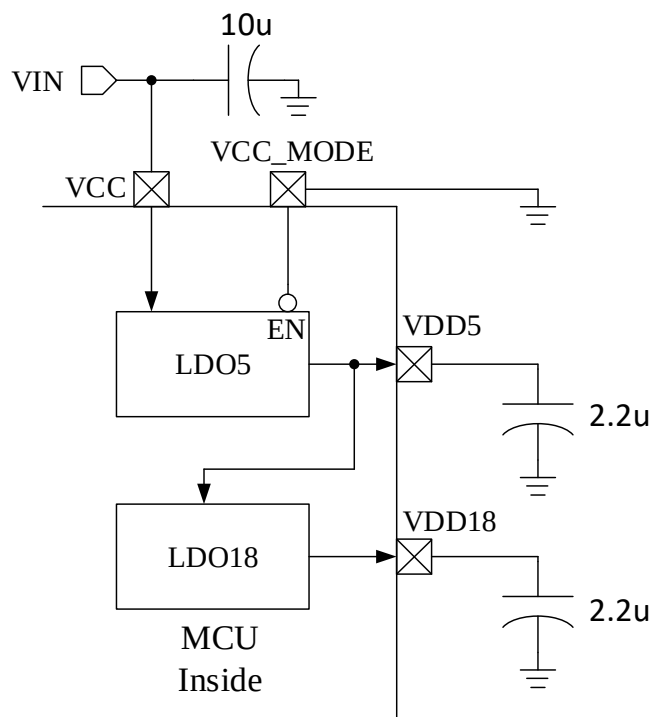


Figure 31-2 Single-power Supply Mode

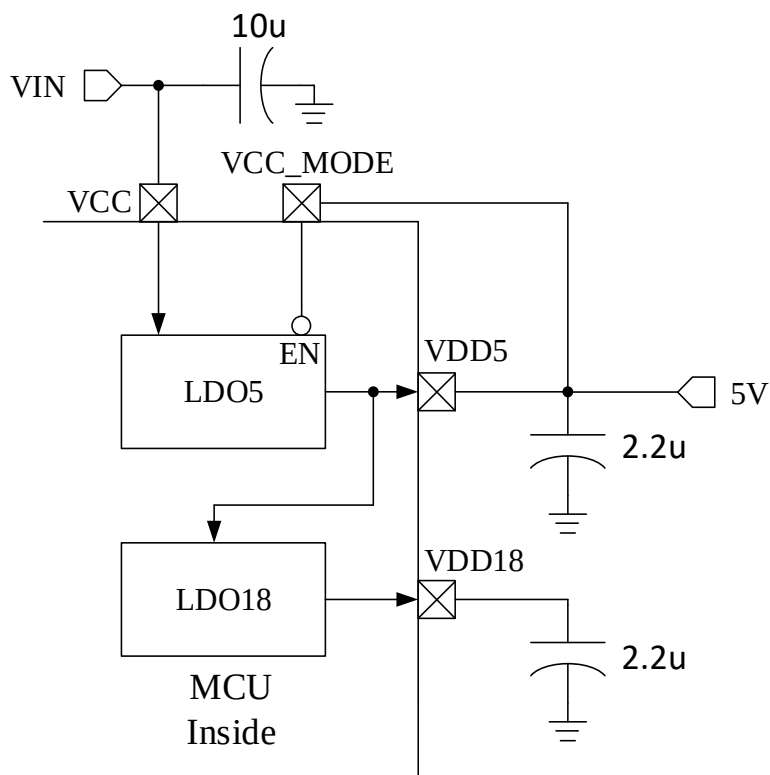


Figure 31-3 Dual-power Supply Mode

31.2 Low Voltage Detector (LVD)

31.2.1 LVD Introduction

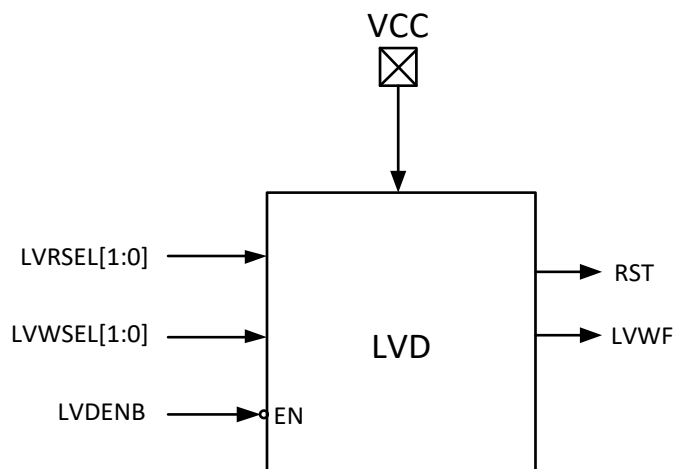


Figure 31-4 LV Detection Module

Configuring CCFG1[LVDENB] = 0 enables LVD module.

31.2.2 LVD Registers

31.2.2.1 CCFG2:RST_MOD (0x401D)

Bit	7	6	5	4	3	2	1	0
Name	LVRSEL		WDTBTEN	WDTRSTEN	RSV		LVWSEL	
Type	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Reset	0	0	0	0	-	-	0	0
Bit	Name		Description					
[7:6]	LVRSEL		Low Voltage Reset Threshold Voltage. VDD5 is detected for the low voltage reset. 00: Low voltage reset is enabled when VDD5 is lower than 2.8V. 01: Low voltage reset is enabled when VDD5 is lower than 3.0V. 10: Low voltage reset is enabled when VDD5 is lower than 3.5V. 11: Low voltage reset is enabled when VDD5 is lower than 3.8V.					
[5:2]			See section 37.1.2.					
[1:0]	LVWSEL		Low Voltage Warning Threshold Voltage. VCC is detected for the low voltage warning. 00: Low voltage warning is enabled when VCC is lower than 7V. 01: Low voltage warning is enabled when VCC is lower than 8V. 10: Low voltage warning is enabled when VCC is lower than 9V. 11: Low voltage warning is enabled when VCC is lower than 10V.					

31.2.2.2 CCFG1:CK_RST_CFG (0x401E)

Bit	7	6	5	4	3	2	1	0
Name	LVWENB	LVWIE	WDTEN	RSV		FCK_SEL		RSV
Type	R/W	R/W	R/W	-	-	R/W	R/W	-
Reset	0	0	0	-	-	0	0	-
Bit	Name	Description						
[7]	LVWENB	Low Voltage Warning Enable 0: Enable 1: Disable						
[6]	LVWIE	LVW Detection Interrupt Enable. CCFG2[LVWSEL] decides VCC LV warning voltage threshold. 0: Disable 1: Enable. LV reset detection circuit shall be enabled (CCFG1[LVDENB] = 0).						
[5:0]		See section 37.1.1.						

31.2.2.3 LVSR(0xDB)

Bit	7	6	5	4	3	2	1	0
Name	RSV		EXT0CFG			TSDF	LVWF	LVWIF
Type	-	-	R/W	R/W	R/W	R	R	R/W
Reset	-	-	0	0	0	0	0	0
Bit	Name	Description						
[7:6]	RSV	Reserved						
[5:3]	EXT0CFG	INT0 Pin Selection 000: P0.0 001: P0.1 010: P0.2 011: P0.3 100: P0.4 101: P0.5 110: P0.6 111: CMP4 Output						
[2]	TSDF	Over Temperature State Indicator 0: The current temperature does not exceed the threshold 1: The current temperature exceeds the threshold Note: This flag bit often works with TSD interrupt flag TCON[5]						
[1]	LVWF	VCC Low Voltage(LV) Flag This bit indicates whether the chip is in the low voltage state. 0: The chip is not in the LV warning state. 1: The chip is in the LV warning state.						
[0]	LVWIF	VCC LV Interrupt Flag This bit indicates whether a low voltage event occurs. It is set to "1" by hardware after low voltage detection interrupt is enabled and an low voltage detection interrupt is detected. A write of "0" in software clears this bit. This bit is not set to "1" by hardware when LVD interrupt is disabled. 0: No Interrupt Pending 1: Interrupt Pending						

32 Flash

32.1 Flash Introduction

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

32.2 FLA_CR (0x85)

Bit	7	6	5	4	3	2	1	0
Name	RSV			FLAERR	FLAACT	RSV	FLAERS	FLAWEN
Type	-	-	-	R/W	R/W	-	R/W	R/W
Reset	-	-	-	0	0	-	0	0

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

32.3 FLA_KEY (0x84)

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

Bit	Name	Description
[7:0]	FLA_KEY	Flash Erase/Write Release The Flash software programming feature is activated after “0x5A” and “0x1F” are written to register FLA_KEY in sequence. If the sequence is incorrect or other values are written, Flash space is frozen until the next reset. After unlocking, any write to the FLA_CR register causes the FLA_KEY to be locked again. Read: The lowest 2 bits reflect internal status, and 6 high-order bits obtains 0x00: 00: Locked 01: Write of 0x5A is done, waiting for 0x1F 10: Frozen 11: Released

32.4 Flash Operations

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

Notes:

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

33 CRC

33.1 CRC Functional Block Diagram

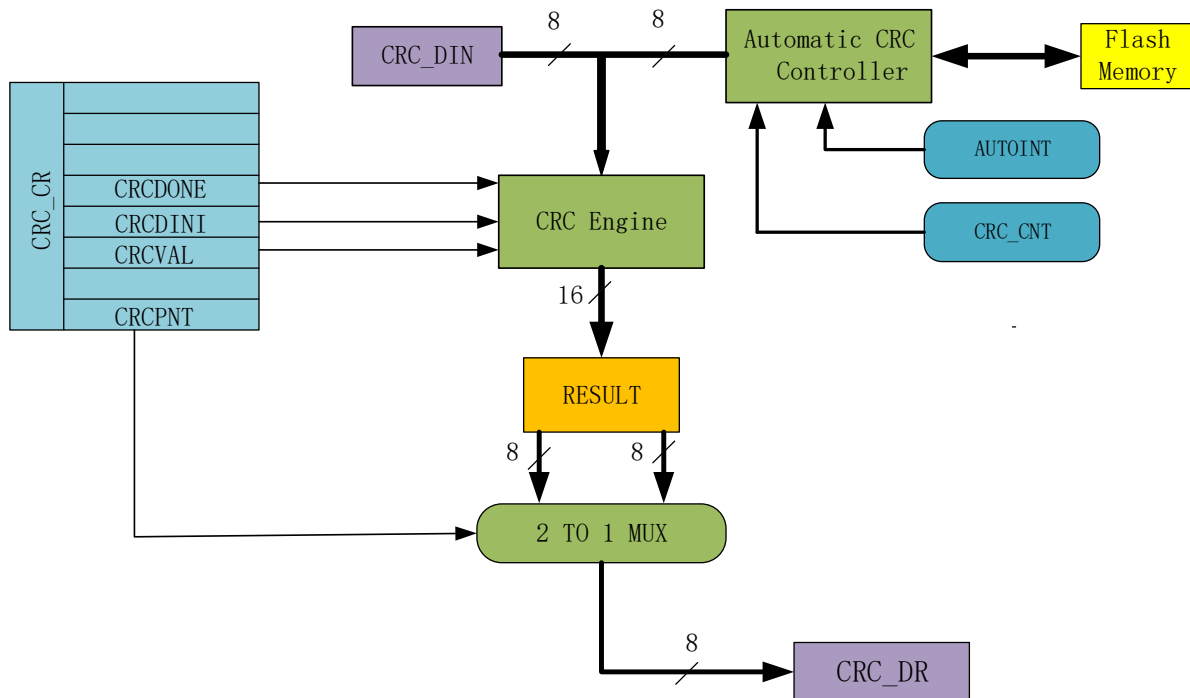


Figure 33-1 CRC Functional Block Diagram

CRC module outputs the result of CRC calculation for any 8-bit data based on a fixed polynomial. As shown in Figure 33-1, CRC receives the 8-bit data from CRC_DIN and sends the 16-bit result to the internal register after the calculation is completed. The result can be indirectly accessed through CRC_CR[CRCPNT] and CRC_DR.

Table 33-1 CRC Criteria and Polynomials

S/N.	CRC Criteria	Polynomial	Hexadecimal Representation
1	CRC12	$x^{12}+x^{11}+x^3+x^2+x+1$	80F
2	CRC16	$x^{16}+x^{15}+x^2+1$	8005
3	CRC16-CCITT-FALSE	$x^{16}+x^{12}+x^5+1$	1021
4	CRC32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^9+x^5+x^4+x+1$	04C11DB7

33.2 CRC16 Polynomial

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

33.3 CRC16 Logic Diagram

The schematic diagram of CRC16 is shown in Figure 33-2. The chip implementation is based on parallel algorithm. For each input byte, MCU calculates the results within one system clock cycle.

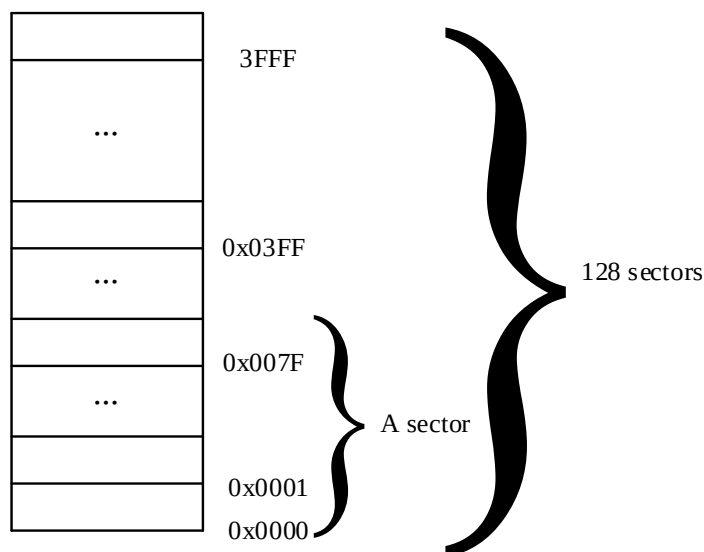


Figure 33-3 ROM Sectors

As shown in Figure 33-3, ROM contains 16k bytes and is divided into 128 sectors, numbered from sector0 to sector127. Each sector contains 128 bytes. For CRC calculation of sectors, the value of CRC_BEG (the starting sector) can be any value falling between 0x00 and 0x00 ~ 0x7F, including 0x00 and 0x7F. The CRC_CNT (total number of sectors to be calculated) can be any value between 0x00~0x7F, including 0x00 and 0x7F.

As CRC_BEG increases, CRC_CNT decreases accordingly. For example, if CRC_BEG is 0x7F, CRC_CNT can be 0x00 only, i.e., the CRC value of the data in the last sector is calculated. In this case, if CRC_CNT is large (0x01 or larger), CRC controller will automatically limits the number of sectors to be calculated. Finally, CRC module only calculates CRC value of the last sector.

33.5 CRC Registers

33.5.1 CRC_CR (0x4022)

Bit	7	6	5	4	3	2	1	0
Name	RSV			CRCDONE	CRCDINI	CRCVAL	AUTOINT	CRCPNT
Type	-	-	-	R	R/W	R/W	R/W	R/W
Reset	-	-	-	1	0	0	0	0
Bit	Name		Description					
[7:5]	RSV		Reserved					
[4]	CRCDONE		CRC Sector Calculation Completion Flag During the calculation, this bit is automatically set to “0” and the software program stops. In other cases, this bit is automatically set to “1” by the hardware, so the software always returns “1” when reading this bit.					
[3]	CRCDINI		CRC Result Initialization Trigger 0: No effect 1: CRC result initialization is triggered.					

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

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

33.5.2 CRC_DIN (0x4021)

Bit	7	6	5	4	3	2	1	0
Name	CRC_DIN							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	CRC_DIN	CRC Input Data Each time a data frame is written to this register, CRC module automatically calculates a new CRC result based on the existing CRC result, and overwrites the original one. Note: It is a virtual register, so the written data is not saved. 0x00 is returned when the address is accessed.						

33.5.3 CRC_DR (0x4023)

Bit	7	6	5	4	3	2	1	0
Name	CRC_DR							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	Name	Description						
[7:0]	CRC_DR	CRC Result Output Each time this register is read or written, the configuration of CRC_CR[CRCPNT] determines whether to access the high-order or low-order bits of the CRC result.						

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

33.5.4 CRC_BEG (0x4024)

Bit	7	6	5	4	3	2	1	0
Name	RSV				CRC_BEG			
Type	-	-	-	-	R/W	R/W	R/W	R/W
Reset	-	-	-	-	0	0	0	0
Bit	Name	Description						
[7]	RSV	Reserved						
[6:0]	CRC_BEG	First ROM Sector Pending Automatic CRC Calculation Example: If CRC_BEG is set to “1”, CRC calculation starts from location 1*128 = 128, or rather from the first byte of sector 2.						

33.5.5 CRC_CNT (0x4025)

Bit	7	6	5	4	3	2	1	0
Name	RSV	CRC_CNT						
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
Bit	Name	Description						
[7]	RSV	Reserved						
[6:0]	CRC_CNT	Offset of Sector Pending Automatic CRC Calculation This bit defines the offset of ROM sector for CRC calculation and determines the last sector pending CRC calculation.						

34 Sleep Mode

34.1 Introduction

The chip operates in three modes: normal mode, standby mode and sleep mode. These modes are selected by setting PCON[IDLE] and PCON[STOP]. The operating states of the module under different power modes are summarized in Table 34-1.

Table 34-1 Power Consumption Modes

Power Mode	Description	Wakeup Source	Power Consumption Performance
Normal	All modules work at full speed except for peripherals that are disabled	NA	High power consumption with best performance
Standby	CPU clock stops and other functional modules are either enabled or disabled depending on their control bit setting. WDT stops.	Any interrupt Reset/Debug on external interrupt	Low power performance with flexible performance
Sleep	The analog fast clock circuit is disconnected and software shall ensure that ADC, FOC, and driver modules are disabled before the chip enters the Sleep Mode. WDT is disabled.	External interrupt, RTC interrupt, Reset/Debug on external interrupt	Extremely low power performance with flexible performance

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

PCON = 0x02;

nop();

nop();

34.2 Sleep Mode Register

34.2.1 PCON (0x87)

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

35 Code Protection

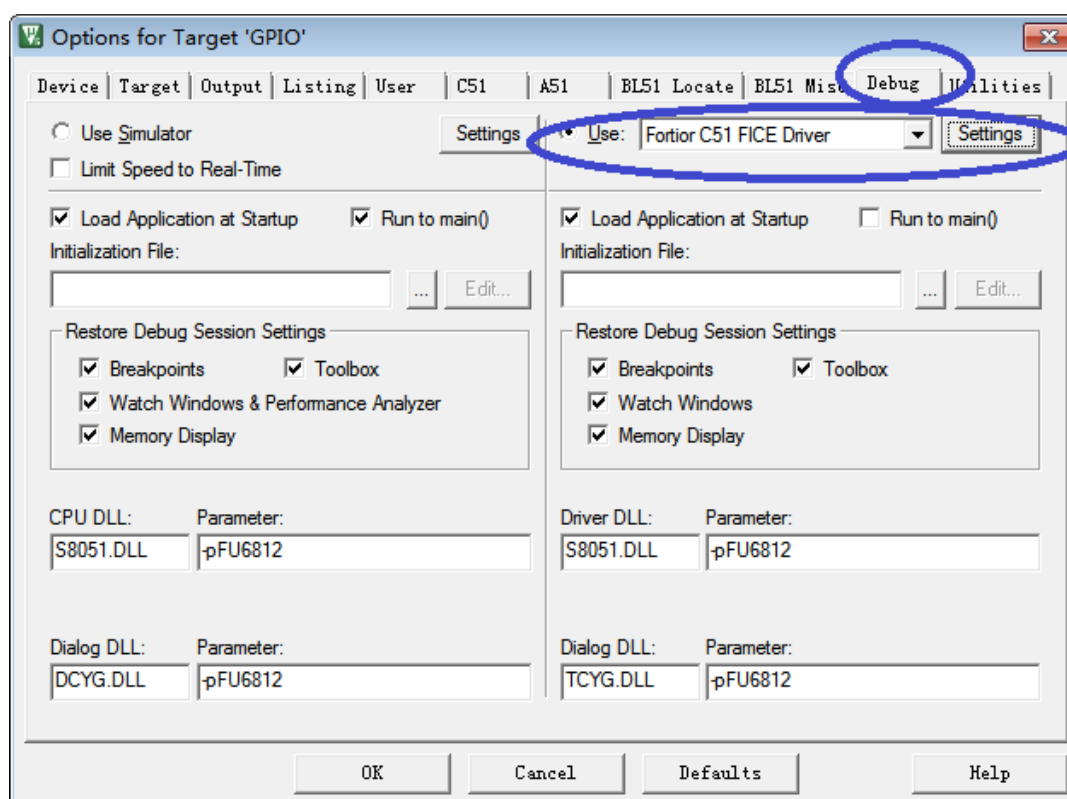


Figure 35-1 Code Protection Configurations

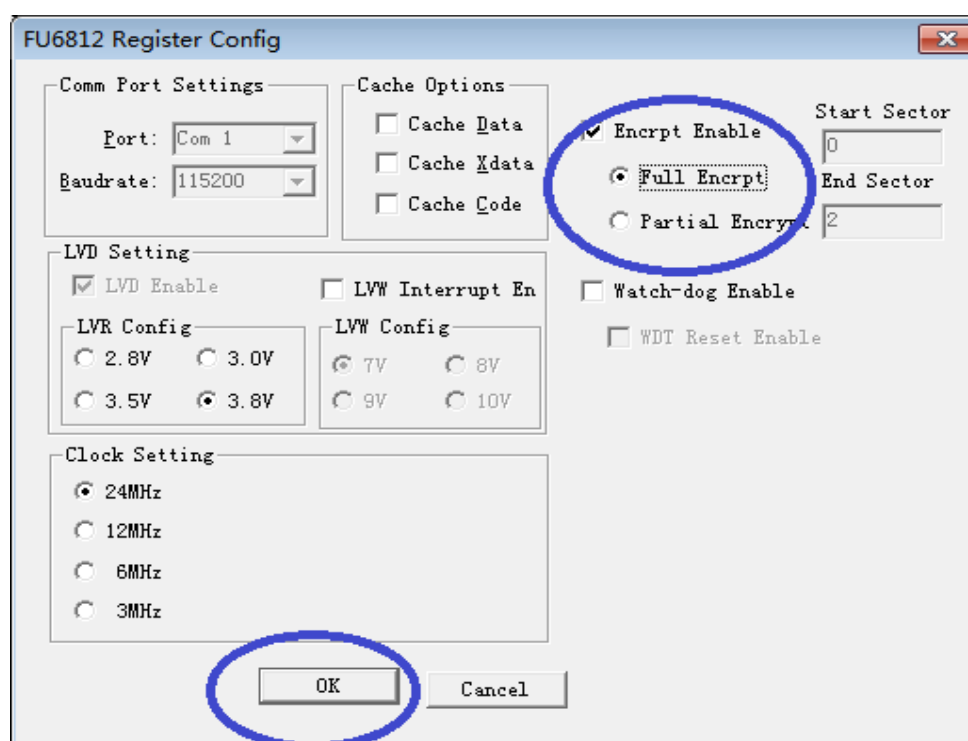


Figure 35-2 Full Code Protection Mode

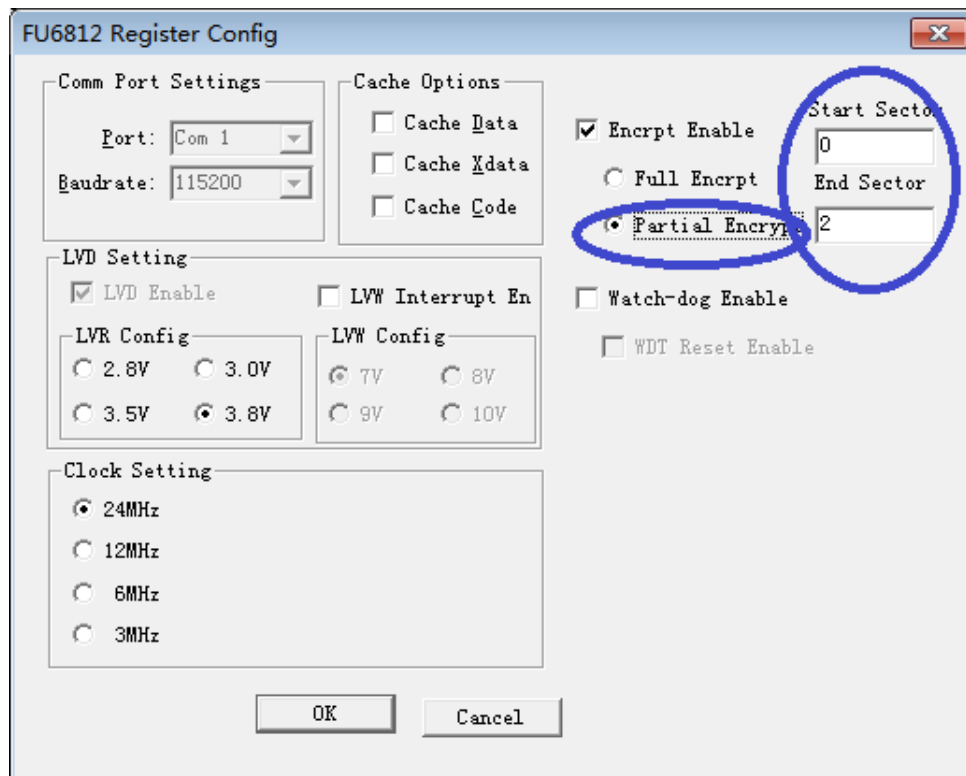


Figure 35-3 Partial Code Protection Mode

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

1. Start 8051 IDE, enter Target Options and select Debug tab. As shown in Figure 35-1, click Settings to proceed with the setting;
2. Select the options as shown in Figure 35-2, and click OK. Then compile the project and download it. Get the BIN file and program it to Flash.

The chip support full code protection mode and partial code protection mode. After Full Encrypt is selected as shown in Figure 35-2, all codes in Flash are protected. After Partial Encrypt is selected as shown in Figure 35-3, the codes from sector 0 to END SECTOR are protected. The last sector is always protected in any case.

The size of each sector is 128 bytes.

36 Magnetoresistance (MR) Sensor

36.1 Introduction

The chip has a built-in angle sensor based on AMR magnetoresistance technology. The MR sensor detects rotating magnetic fields to obtain two pairs of differential sinusoidal signals with an amplitude of around 120mV that are output to the pins of the chip and amplified by AMP1 and AMP2. The angle sensor supports 180°-degree angle measurement and provides a 360°-degree absolute position when used with Hall sensor or other technologies.

36.2 Instructions

Figure 36-1 shows the position relationship between sensor chip and magnet during basic angle measurement, i.e., when a magnetic field of 80G or greater is applied parallel to the sensor surface. For instance, placing a radially magnetized permanent magnet above the sensor, or on an end of the shaft of a motor or other equipment. When the magnet rotates with the motor, the direction of magnetic field sensed by the sensor changes accordingly. A corresponding direction of magnetic field θ is provided at the same time. Figure 36-2 presents the relationship between the voltage waveform of signal output and the angle. It shows that the output signals of Bridge A and Bridge B are both of 180° phase. The output waveforms are exactly identical at 0°-180° and 180°-360°. Therefore, based on the outputs from Bridge A and Bridge B, the angle θ between 0°-180° can be determined. With Hall sensor or other technologies, any angle within 0°-360° can be precisely measured. The output voltage is as follows, where the typical value of S is 12mV/V.

Output voltage of Bridge A: $V = VDD * S * \sin(2\theta)$

Output voltage of Bridge B: $V = VDD * S * \cos(2\theta)$

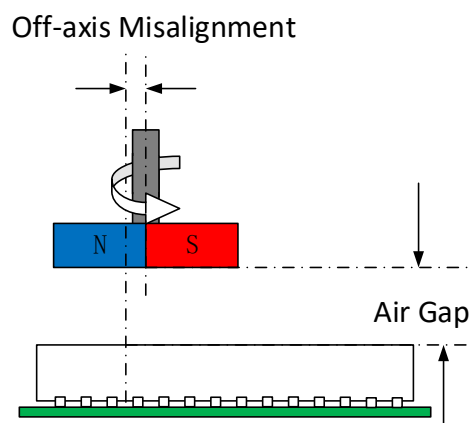


Figure 36-1 Position Relationship between Sensor Chip and Magnet

Table 36-1 Electrical and Magnetic Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit
Dmag	Magnet Diameter	-	6	-	mm
Tmag	Magnet Thickness	-	2.5	-	mm
Bpk	Peak Flux Density	80	-	-	Gs
AG	Air Gap	0.5	2	-	mm
DISP	Radial Displacement	-	-	0.3	mm
TCmag1	NdFeB	-	-0.12	-	%/° C
TCmag2	SmCo	-	-0.035	-	%/° C

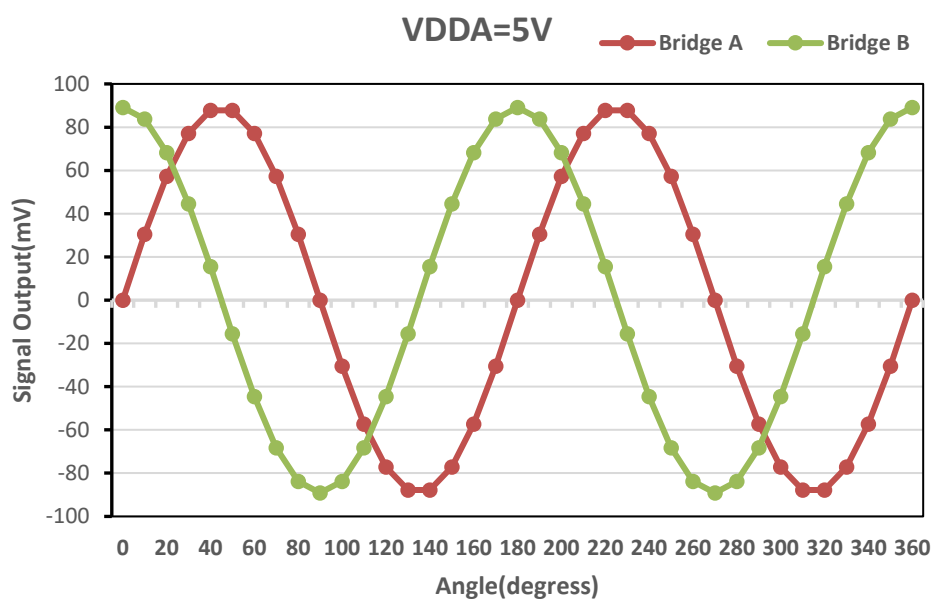


Figure 36-2 Voltage Waveform of Signal Output and Angle

36.3 Functional Block Diagram

A simplified block diagram of FU6361Q application is shown in Figure 36-3. Two interleaved Wheatstone bridges generate sine and cosine signals. The signals are sent to operational amplifier on the chip through externally configured resistors to adjust the magnification times of the operational amplifier. ADC and hardware decoder are used for angular decoding.

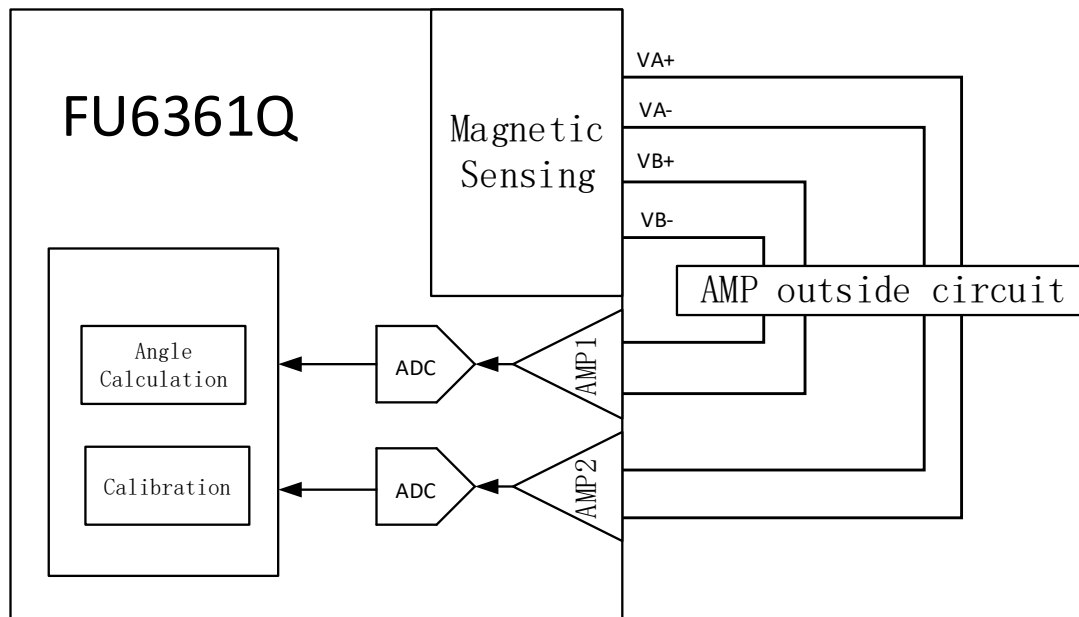


Figure 36-3 Functional Block Diagram of MR Sensor

36.4 Application Recommendation

As shown below, the output signals of VA and VB pins of the angle sensor are differentially amplified by AMP1 and AMP2. The recommended magnification is 15 to 40 times. Biasing chip is used in VHALF module for biasing voltage. A 100pF capacitor is placed in the feedback loop to reduce the bandwidth of the system and filter the noise from other than the sensor and amplifier circuit.

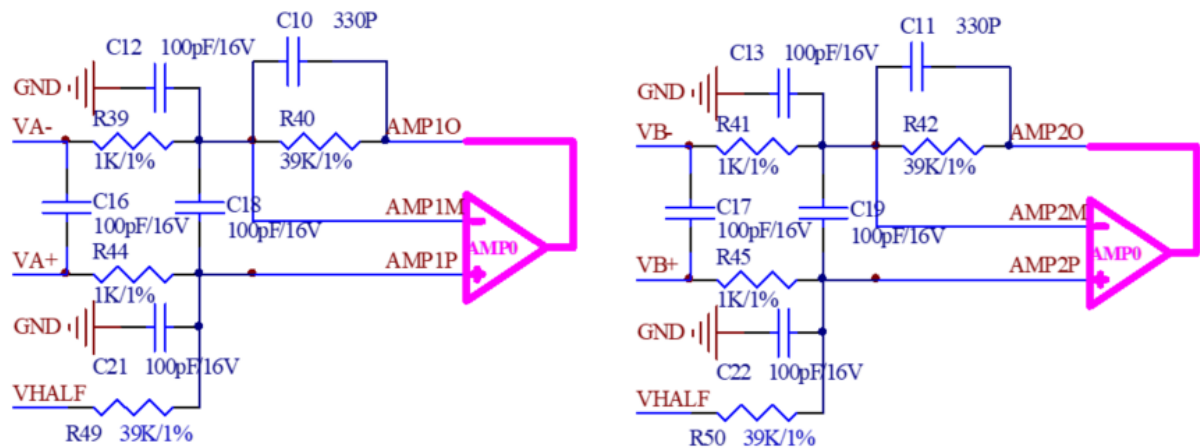


Figure 36-4 Recommended Circuit Diagram

37 Configuration Registers

37.1 CCFG

37.1.1 CCFG1: CK_RST_CFG

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

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

37.1.2 CCFG2: RST_MOD

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

Bit	7	6	5	4	3	2	1	0
Name	LVRSEL		WDTBTEN	WDTRSTEN	RSV		LVWSEL	
Type	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Reset	0	0	0	0	-	-	0	0

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

38 Revision History

Rev.	Description	Date	Prepared By
V1.0	First release, translated from Chinese version 1.0	2023/12/05	Eric Deng

Copyright Notice

Copyright by Fortior Technology (Shenzhen) Co., Ltd. All Rights Reserved.

Right to make changes — Fortior Technology (Shenzhen) Co., Ltd. reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. The information contained in this manual is provided for the general use by our customers. Our customers shall ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Fortior Technology (Shenzhen) Co., Ltd. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

This manual is copyrighted by Fortior Technology (Shenzhen) Co., Ltd. You may not reproduce, transmit, transcribe, store in a retrieval system, or translate into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, any part of this publication without the expressly written permission from Fortior Technology (Shenzhen) Co., Ltd. You may not alter or remove any copyright or other notice from copies of this content.

If there are any differences between the Chinese and the English contents, please take the Chinese version as the standard.

Fortior Technology (Shenzhen) Co., Ltd.

Room 203, 2/F, Building No.11, Keji Central Road 2,
Software Park, High-Tech Industrial Park, Shenzhen, P.R. China 518057
Tel: 0755-26867710
Fax: 0755-26867715
URL: <http://www.fortiortech.com>

Contained herein

Copyright by Fortior Technology (Shenzhen) Co., Ltd. All rights reserved.