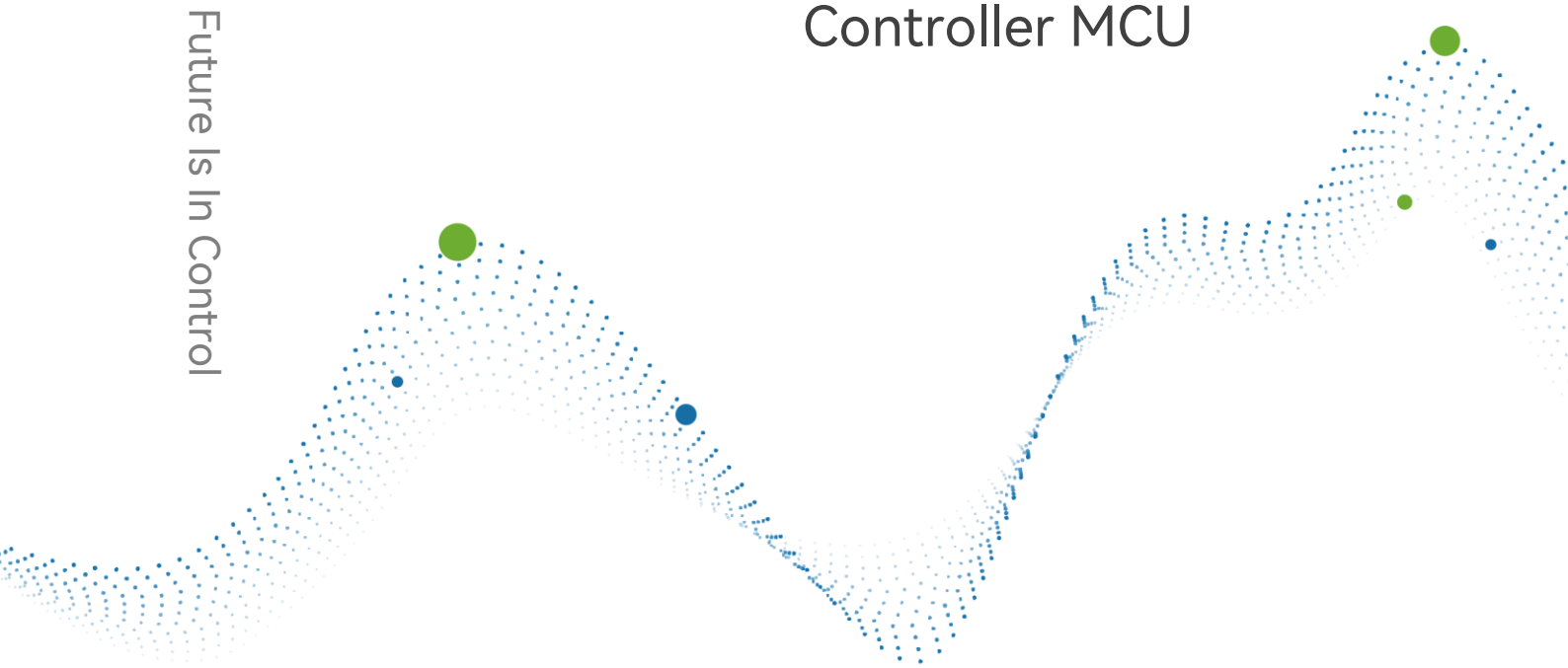
Fortior Tech

DATASHEET

FU6367Q

Three-phase Motor Controller MCU

Future Is In Control



Version No.: V1.0

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.

Disclaimer

This document is the property of Fortior Technology (Shenzhen) Co., Ltd. and its subsidiaries (hereinafter referred to as "Fortior Technology").

This document is provided on an "AS IS" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights of any third-party. Any and all information stated herein is for reference purpose only, and Fortior Technology hereby disclaims any and all warranties and liabilities of any kind. The customer has the full responsibility to design, program and test their applications, evaluate functionality and safety of their final products and accept all loss or damage, legal claim or lawsuit whatsoever arising from the use of the information, data or opinions contained herein, and agrees to defend, hold harmless and indemnify Fortior Technology from and against any and all claims, damages and liabilities. No license, express or implied, to any intellectual property rights of Fortior Technology is granted by this document. Fortior Technology reserves the rights to amend, revise or make changes to this document at any time without prior notice.

In the event of any inconsistency, conflict or ambiguity under this disclaimer, the final interpretation is at the sole discretion of Fortior Technology.

Contents

Copyright Notice 1
Disclaimer 2
Contents 3
Explanation of Symbols 28
Abbreviations 29
1 System Introduction 31
1.1 Features 31
1.2 Applications 33
1.3 Overview 33
1.4 Functional Block Diagram 34
1.4.1 FU6367Q 34
1.5 Memory Organization 35
1.5.1 Program Memory 35
1.5.2 Data Memory 35
1.5.3 SFR 36
1.5.4 XSFR 37
2 Pin Definitions 41
2.1 FU6367Q QFN56 Pins 41
2.2 FU6367Q QFN56 Pinout Diagram 46
3 Package Information 48
3.1 FU6367Q QFN56_7X7 48
4 Ordering Information 49
5 Electrical Characteristics 50
5.1 Absolute Maximum Ratings 50
5.2 MR Sensor Electrical Characteristics 50
5.3 Global Electrical Characteristics 51
5.4 GPIO Electrical Characteristics 52

5.5 6N Pre-driver IO Electrical Characteristics 52
5.6 ADC Electrical Characteristics 53
5.7 VREF and VHALF Electrical Characteristics 54
5.8 Operational Amplifier Electrical Characteristics 54
5.9 BEMF Electrical Characteristics 55
5.10 OSC Electrical Characteristics 55
5.11 Reset Electrical Characteristics 55
5.12 LDO Electrical Characteristics 55
5.13 Package Thermal Resistance 56
6 Reset Control 57
6.1 Reset Source (RST_SR) 57
6.2 Reset Enable 57
6.3 External Pin Reset and Power-on Reset 57
6.4 Low Voltage Detection Reset 57
6.5 Watchdog Timer Reset 58
6.6 Flash Error Detector Reset 58
6.7 Debug Reset 58
6.8 Soft Reset 58
6.9 Reset Registers 59
6.9.1 RST_SR (0xC9) 59
7 Interrupt 60
7.1 Interrupt Introduction 60
7.2 Interrupt Enable 60
7.3 External Interrupts 61
7.4 Interrupt Summary 61
7.5 Interrupt Registers 63
7.5.1 IE (0xA8) 63
7.5.2 IP0 (0x8A) 63
7.5.3 IP1 (0x8B) 64
7.5.4 IP2 (0x8C) 64
7.5.5 IP3 (0x8D) 65
7.5.6 TCON (0x88) 65

7.5.7 P1_IE (0xD1) 66
7.5.8 P1_IF (0xD2) 67
7.5.9 P4_IE (0xD3) 67
7.5.10 P4_IF (0xD4) 68
8 I²C 70
8.1 I²C Introduction 70
8.2 I²C Operations 71
8.2.1 Master Mode 71
8.2.2 Slave Mode 72
8.2.3 I²C Interrupt Sources 73
8.3 I²C Registers 73
8.3.1 I2C_CR (0x4028) 73
8.3.2 I2C_ID (0x4029) 74
8.3.3 I2C_DR (0x402A) 74
8.3.4 I2C_SR (0x402B) 74
9 SPI 77
9.1 SPI Introduction 77
9.2 SPI Operations 77
9.2.1 Signal Descriptions 77
9.2.1.1 Master Out, Slave In (MOSI) 77
9.2.1.2 Master In, Slave Out (MISO) 78
9.2.1.3 Serial Clock (SCLK) 78
9.2.1.4 Slave Select (NSS) 78
9.2.2 SPI Master Mode 80
9.2.2.1 Master Mode Configurations 80
9.2.3 SPI Slave Mode 80
9.2.3.1 Slave Mode Configurations 81
9.2.4 SPI Interrupt Sources 81
9.2.5 Serial Clock Timing 82
9.3 SPI Registers 83
9.3.1 SPI_CR0 (0x4030) 83
9.3.2 SPI_CR1 (0x4031) 84

9.3.3 SPI_CLK (0x4032) 85
9.3.4 SPI_DR (0x4033) 86
10 UART 87
10.1 UART Introduction 87
10.2 UART Operations 87
10.2.1 UART1 Operating Instructions 87
10.2.1.1 UART1 Mode0 87
10.2.1.2 UART1 Mode1 88
10.2.1.3 UART1 Mode2 88
10.2.1.4 UART1 Mode3 88
10.2.1.5 UART1 Interrupt Sources 89
10.2.2 UART2 Operating Instructions 89
10.2.2.1 UART2 Mode0 89
10.2.2.2 UART2 Mode1 89
10.2.2.3 UART2 Mode2 90
10.2.2.4 UART2 Mode3 90
10.2.2.5 UART2 Interrupt Sources 90
10.3 UART1 Registers 91
10.3.1 UT_CR (0x98) 91
10.3.2 UT_DR (0x99) 92
10.3.3 UT_BAUD (0x9A, 0x9B) 92
10.4 UART2 Registers 93
10.4.1 UT2_CR (0xD8) 93
10.4.2 UT2_DR (0x89) 94
10.4.3 UT2_BAUD (0x4042, 0x4043) 94
11 LIN 96
11.1 LIN Introduction 96
11.2 LIN Slave Mode Operations 97
11.3 Sleep and Wakeup 98
11.4 Error Detection and Handling 98
11.5 Other Matters 98
11.6 LIN Registers 99

11.6.1 LIN_CR (0xB9) 99
11.6.2 LIN_SR (0xBA) 100
11.6.3 LIN_CSR (0xBB) 101
11.6.4 LIN_ID (0xBC) 102
11.6.5 LIN_SIZE (0xBD) 102
11.6.6 LIN_BAUD (0xBF,0xBE) 103
12 CAN 104
12.1 CAN Introduction 104
12.1.1 Frame Structure 105
12.2 General Descriptions on CAN Controller 108
12.2.1 Key Features 109
12.3 Operating Modes 109
12.3.1 Reset Mode 109
12.3.2 Normal Operation Mode 109
12.3.3 Sleep Mode 109
12.4 Feature Descriptions 110
12.4.1 Bit-time Characteristics 110
12.4.2 Sending Management 111
12.4.3 Receiving Management 111
12.4.4 Receive Filtering 112
12.4.4.1 Standard Frame Single Filter Mode 113
12.4.4.2 Extended Frame Single Filter Mode 113
12.4.4.3 Standard Frame Double Filter Mode 114
12.4.4.4 Extended Frame Double Filter Mode 114
12.4.5 Error Management 115
12.5 CAN Interrupt Sources 115
12.6 Test Mode 117
12.6.1 Self-test Mode 117
12.6.2 Monitoring Mode 118
12.6.3 Self-test Mode & Monitoring Mode 118
12.7 CAN Registers 119
12.7.1 CAN_CR0 (0xB1) 119

12.7.2 CAN_CR1 (0xB2) 120
12.7.3 CAN_SR (0xB3) 120
12.7.4 CAN_IER (0xB4) 121
12.7.5 CAN_CR2 (0xB5) 122
12.7.6 CAN_IFR (0xB6) 123
12.7.7 CAN_BTR0 (0xB7) 124
12.7.8 CAN_BTR1 (0xC2) 125
12.7.9 CAN_BTR2 (0xC3) 125
12.7.10 CAN_FIR0 (0xC4) 125
12.7.11 CAN_FIR1 (0xC5) 125
12.7.12 CAN_FIR2 (0xC6) 126
12.7.13 CAN_FIR3 (0xC7) 126
12.7.14 CAN_FMR0 (0xCA) 126
12.7.15 CAN_FMR1 (0xCB) 126
12.7.16 CAN_FMR2 (0xCC) 127
12.7.17 CAN_FMR3 (0xCD) 127
12.7.18 CAN_ALC (0xCE) 127
12.7.19 CAN_ESR (0xCF) 127
12.7.20 CAN_LIM (0xDD) 128
12.7.21 CAN_TEC (0xDE) 128
12.7.22 CAN_REC (0xDF) 129
12.7.23 CAN_TXCR (0x0ED7) 129
12.7.24 CAN_TXID0 (0x0ED6) 129
12.7.25 CAN_TXID1 (0x0ED5) 130
12.7.26 CAN_TXID2 (0x0ED4) 130
12.7.27 CAN_TXID3 (0x0ED3) 130
12.7.28 CAN_TXDR0 (0x0ED2) 130
12.7.29 CAN_TXDR1 (0x0ED1) 131
12.7.30 CAN_TXDR2 (0x0ED0) 131
12.7.31 CAN_TXDR3 (0x0ECF) 131
12.7.32 CAN_TXDR4 (0x0ECE) 131
12.7.33 CAN_TXDR5 (0x0ECD) 132

12.7.34 CAN_TXDR6 (0x0ECC) 132
12.7.35 CAN_TXDR7 (0x0ECB) 132
12.7.36 CAN_RX0CR (0x0ECA) 132
12.7.37 CAN_RX0ID0 (0x0EC9) 133
12.7.38 CAN_RX0ID1 (0x0EC8) 133
12.7.39 CAN_RX0ID2 (0x0EC7) 133
12.7.40 CAN_RX0ID3 (0x0EC6) 133
12.7.41 CAN_RX0DR0 (0x0EC5) 134
12.7.42 CAN_RX0DR1 (0x0EC4) 134
12.7.43 CAN_RX0DR2 (0x0EC3) 134
12.7.44 CAN_RX0DR3 (0x0EC2) 134
12.7.45 CAN_RX0DR4 (0x0EC1) 135
12.7.46 CAN_RX0DR5 (0x0EC0) 135
12.7.47 CAN_RX0DR6 (0x0EBF) 135
12.7.48 CAN_RX0DR7 (0x0EBE) 135
12.7.49 CAN_RX1CR (0x0EBD) 136
12.7.50 CAN_RX1ID0 (0x0EBC) 136
12.7.51 CAN_RX1ID1 (0x0EBB) 136
12.7.52 CAN_RX1ID2 (0x0EBA) 137
12.7.53 CAN_RX1ID3 (0x0EB9) 137
12.7.54 CAN_RX1DR0 (0x0EB8) 137
12.7.55 CAN_RX1DR1 (0x0EB7) 137
12.7.56 CAN_RX1DR2 (0x0EB6) 138
12.7.57 CAN_RX1DR3 (0x0EB5) 138
12.7.58 CAN_RX1DR4 (0x0EB4) 138
12.7.59 CAN_RX1DR5 (0x0EB3) 138
12.7.60 CAN_RX1DR6 (0x0EB2) 139
12.7.61 CAN_RX1DR7 (0x0EB1) 139
12.7.62 CAN_RX2CR (0x0EB0) 139
12.7.63 CAN_RX2ID0 (0x0EAF) 139
12.7.64 CAN_RX2ID1 (0x0EAE) 140
12.7.65 CAN_RX2ID2 (0x0EAD) 140

12.7.66 CAN_RX2ID3 (0x0EAC) 140
12.7.67 CAN_RX2DR0 (0x0EAB) 141
12.7.68 CAN_RX2DR1 (0x0EAA) 141
12.7.69 CAN_RX2DR2 (0x0EA9) 141
12.7.70 CAN_RX2DR3 (0x0EA8) 141
12.7.71 CAN_RX2DR4 (0x0EA7) 142
12.7.72 CAN_RX2DR5 (0x0EA6) 142
12.7.73 CAN_RX2DR6 (0x0EA5) 142
12.7.74 CAN_RX2DR7 (0x0EA4) 142
13 MDU 143
13.1 MDU Introduction 143
13.2 MDU Features 143
13.3 MDU Instructions 143
13.3.1 MDU Operations 143
13.3.2 16-bit Signed Multiplication with the Result Shifted Left by One-bit 144
13.3.3 16-bit Signed Multiplication 144
13.3.4 16-bit Unsigned Multiplication 144
13.3.5 32-bit/16-bit Unsigned Division 145
13.3.6 LPF 145
13.3.7 Coordinate Transformation 146
13.3.8 Arctangent 147
13.3.9 PI/PID 148
13.3.9.1 PI/PID Introduction 148
13.3.9.2 PI/PID Features 148
13.3.9.3 PI/PID Operations 148
13.4 MDU Registers 150
13.4.1 MDU_CR (0xC1) 150
13.4.2 MUL0_MA (0x0FA0, 0x0FA1) 150
13.4.3 MUL0_MB (0x0FA2, 0x0FA3) 151
13.4.4 MUL0_MC (0x0FA4, 0x0FA5, 0x0FA6, 0x0FA7) 151
13.4.5 MUL1_MA (0x0F98, 0x0F99) 152
13.4.6 MUL1_MB (0x0F9A, 0x0F9B) 152

13.4.7 MUL1_MC (0x0F9C, 0x0F9D, 0x0F9E, 0x0F9F) 152
13.4.8 MUL2_MA (0x0F40, 0x0F41) 153
13.4.9 MUL2_MB (0x0F42, 0x0F43) 153
13.4.10 MUL2_MC (0x0F44, 0x0F45, 0x0F46, 0x0F47) 154
13.4.11 MUL3_MA (0x0F38, 0x0F39) 154
13.4.12 MUL3_MB (0x0F3A, 0x0F3B) 155
13.4.13 MUL3_MC (0x0F3C, 0x0F3D, 0x0F3E, 0x0F3F) 155
13.4.14 DIV0_DA (0x0F8C, 0x0F8D, 0x0F8E, 0x0F8F) 156
13.4.15 DIV0_DB (0x0F90, 0x0F91) 156
13.4.16 DIV0_DQ (0x0F92, 0x0F93, 0x0F94, 0x0F95) 157
13.4.17 DIV0_DR (0x0F96, 0x0F97) 158
13.4.18 DIV1_DA (0x0F80, 0x0F81, 0x0F82, 0x0F83) 158
13.4.19 DIV1_DB (0x0F84, 0x0F85) 159
13.4.20 DIV1_DQ (0x0F86, 0x0F87, 0x0F88, 0x0F89) 159
13.4.21 DIV1_DR (0x0F8A, 0x0F8B) 160
13.4.22 DIV2_DA (0x0F2C, 0x0F2D, 0x0F2E, 0x0F2F) 160
13.4.23 DIV2_DB (0x0F30, 0x0F31) 161
13.4.24 DIV2_DQ (0x0F32, 0x0F33, 0x0F34, 0x0F35) 161
13.4.25 DIV2_DR (0x0F36, 0x0F37) 162
13.4.26 DIV3_DA (0x0F20, 0x0F21, 0x0F22, 0x0F23) 162
13.4.27 DIV3_DB (0x0F24, 0x0F25) 163
13.4.28 DIV3_DQ (0x0F26, 0x0F27, 0x0F28, 0x0F29) 163
13.4.29 DIV3_DR (0x0F2A, 0x0F2B) 164
13.4.30 SCAT0_COS (0x0F16, 0x0F17) 164
13.4.31 SCAT0_SIN (0x0F18, 0x0F19) 164
13.4.32 SCAT0_THE (0x0F1A, 0x0F1B) 165
13.4.33 SCAT0_RES1 (0x0F1C, 0x0F1D) 165
13.4.34 SCAT0_RES2 (0x0F1E, 0x0F1F) 166
13.4.35 SCAT1_COS (0x0F0C, 0x0F0D) 166
13.4.36 SCAT1_SIN (0x0F0E, 0x0F0F) 166
13.4.37 SCAT1_THE (0x0F10, 0x0F11) 167
13.4.38 SCAT1_RES1 (0x0F12, 0x0F13) 167

13.4.39 SCAT1_RES2 (0x0F14, 0x0F15) 168
13.4.40 SCAT2_COS (0x0F02, 0x0F03) 168
13.4.41 SCAT2_SIN (0x0F04, 0x0F05) 168
13.4.42 SCAT2_THE (0x0F06, 0x0F07) 169
13.4.43 SCAT2_RES1 (0x0F08, 0x0F09) 169
13.4.44 SCAT2_RES2 (0x0F0A, 0x0F0B) 170
13.4.45 SCAT3_COS (0x0EF8, 0x0EF9) 170
13.4.46 SCAT3_SIN (0x0EFA, 0x0EFB) 170
13.4.47 SCAT3_THE (0x0EFC, 0x0EFD) 171
13.4.48 SCAT3_RES1 (0x0EFE, 0x0EFF) 171
13.4.49 SCAT3_RES2 (0x0F00, 0x0F01) 172
13.4.50 LPF0_K (0x0FD0, 0x0FD1) 172
13.4.51 LPF0_X (0x0FD2, 0x0FD3) 172
13.4.52 LPF0_Y (0x0FD4, 0x0FD5, 0x0FD6, 0x0FD7) 173
13.4.53 LPF1_K (0x0FC8, 0x0FC9) 173
13.4.54 LPF1_X (0x0FCA, 0x0FCB) 174
13.4.55 LPF1_Y (0x0FCC, 0x0FCD, 0x0FCE, 0x0FCF) 174
13.4.56 LPF2_K (0x0F78, 0x0F79) 175
13.4.57 LPF2_X (0x0F7A, 0x0F7B) 175
13.4.58 LPF2_Y (0x0F7C, 0x0F7D, 0x0F7E, 0x0F7F) 176
13.4.59 LPF3_K (0x0F70, 0x0F71) 176
13.4.60 LPF3_X (0x0F72, 0x0F73) 177
13.4.61 LPF3_Y (0x0F74, 0x0F75, 0x0F76, 0x0F77) 177
13.4.62 PI0_KP (0x0FB8, 0x0FB9) 178
13.4.63 PI0_EK1 (0x0FBA, 0x0FBB) 178
13.4.64 PI0_EK (0x0FBC, 0x0FBD) 178
13.4.65 PI0_KI (0x0FBE, 0x0FBF) 179
13.4.66 PI0_UKH (0x0FC0, 0x0FC1) 179
13.4.67 PI0_UKL (0x0FC2, 0x0FC3) 180
13.4.68 PI0_UKMAX (0x0FC4, 0x0FC5) 180
13.4.69 PI0_UKMIN (0x0FC6, 0x0FC7) 180
13.4.70 PI1_KP (0x0FA8, 0x0FA9) 181

13.4.71 PI1_EK1 (0x0FAA, 0x0FAB) 181
13.4.72 PI1_EK (0x0FAC, 0x0FAD) 182
13.4.73 PI1_KI (0x0FAE, 0x0FAF) 182
13.4.74 PI1_UKH (0x0FB0, 0x0FB1) 182
13.4.75 PI1_UKL (0x0FB2, 0x0FB3) 183
13.4.76 PI1_UKMAX (0x0FB4, 0x0FB5) 183
13.4.77 PI1_UKMIN (0x0FB6, 0x0FB7) 184
13.4.78 PI2_KP (0x0F5C, 0x0F5D) 184
13.4.79 PI2_EK1 (0x0F5E, 0x0F5F) 184
13.4.80 PI2_EK (0x0F60, 0x0F61) 185
13.4.81 PI2_KI (0x0F62, 0x0F63) 185
13.4.82 PI2_UKH (0x0F64, 0x0F65) 186
13.4.83 PI2_UKL (0x0F66, 0x0F67) 186
13.4.84 PI2_UKMAX (0x0F68, 0x0F69) 186
13.4.85 PI2_UKMIN (0x0F6A, 0x0F6B) 187
13.4.86 PI2_KD (0x0F6C, 0x0F6D) 187
13.4.87 PI2_EK2 (0x0F6E, 0x0F6F) 188
13.4.88 PI3_KP (0x0F48, 0x0F49) 188
13.4.89 PI3_EK1 (0x0F4A, 0x0F4B) 188
13.4.90 PI3_EK (0x0F4C, 0x0F4D) 189
13.4.91 PI3_KI (0x0F4E, 0x0F4F) 189
13.4.92 PI3_UKH (0x0F50, 0x0F51) 190
13.4.93 PI3_UKL (0x0F52, 0x0F53) 190
13.4.94 PI3_UKMAX (0x0F54, 0x0F55) 190
13.4.95 PI3_UKMIN (0x0F56, 0x0F57) 191
13.4.96 PI3_KD (0x0F58, 0x0F59) 191
13.4.97 PI3_EK2 (0x0F5A, 0x0F5B) 192
14 FOC 193
14.1 FOC Overview 193
14.1.1 FOC Introduction 193
14.1.2 Reference Voltage (VREF) Input 194
14.1.3 PI Controller 194

14.1.4 Coordinate Transformation 194
14.1.4.1 Inverse Park Transformation 194
14.1.4.2 Inverse Clarke Transformation 195
14.1.4.3 Clarke Transformation 195
14.1.4.4 Park Transformation 195
14.1.5 SVPWM 196
14.1.5.1 Continuous SVPWM 197
14.1.5.2 Discontinuous SVPWM 197
14.1.6 Overmodulation 198
14.1.7 Deadtime Compensation 198
14.1.8 Current and Voltage Sampling 198
14.1.8.1 Single-shunt Current Sampling Mode 198
14.1.8.2 Dual/Triple-shunt Current Sampling Mode 201
14.1.8.3 Current Sampling Offset 202
14.1.9 Angle Mode 202
14.1.9.1 Forced Ramping Angle 203
14.1.9.2 Forced Pulling Angle 203
14.1.9.3 Estimator Output Angle 204
14.1.9.3.1 Estimated Angle of Estimator 204
14.1.9.3.2 Forced Angle of Estimator 205
14.1.9.3.3 Angle Smooth Switching 205
14.1.9.3.4 Angle Compensation 206
14.1.10 Motor Real-time Parameters 206
14.1.10.1 Tailwind/headwind Detection 207
14.1.10.2 BEMF Detection 207
14.1.10.3 Motor Power 207
14.1.11 FG Output Generation 207
14.2 FOC Registers 208
14.2.1 FOC_CR0 (0x409F) 208
14.2.2 FOC_CR1 (0x40A0) 209
14.2.3 FOC_CR2 (0x40A1) 210
14.2.4 FOC_CR3 (0x409E) 211

14.2.5 FOC_CR4 (0x409D) 212
14.2.6 FOC_CR5 (0x409C) 212
14.2.7 FOC_TSMIN (0x40A2) 213
14.2.8 FOC_TGLI (0x40A3) 213
14.2.9 FOC_TBLO (0x40A4) 214
14.2.10 FOC_TRGDLY (0x40A5) 214
14.2.11 FOC_CSO (0x40A6, 0x40A7) 215
14.2.12 FOC_RTHESTEP (0x40A8, 0x40A9) 215
14.2.13 FOC_RTHEACC (0x40AA, 0x40AB) 216
14.2.14 FOC_EOMELPF (0x40AA, 0x40AB) 216
14.2.15 FOC_RTHECNT (0x40AC) 217
14.2.16 FOC_THECOR (0x40AD) 217
14.2.17 FOC_EMF (0x40AE, 0x40AF) 217
14.2.18 FOC_THECOMP (0x40AE, 0x40AF) 218
14.2.19 FOC_DMAX (0x40B0, 0x40B1) 218
14.2.20 FOC_OMEEST (0x40B0, 0x40B1) 218
14.2.21 FOC_DMIN (0x40B2, 0x40B3) 219
14.2.22 FOC_QMAX (0x40B4, 0x40B5) 219
14.2.23 FOC_QMIN (0x40B6, 0x40B7) 220
14.2.24 FOC_UD (0x40B8, 0x40B9) 220
14.2.25 FOC_UQ (0x40BA, 0x40BB) 220
14.2.26 FOC_ID (0x40BC, 0x40BD) 221
14.2.27 FOC_IQ (0x40BE, 0x40BF) 221
14.2.28 FOC_IBET (0x40C0, 0x40C1) 222
14.2.29 FOC_IQ_LPFK (0x40C0) 222
14.2.30 FOC_ID_LPFK (0x40C1) 222
14.2.31 FOC_VBET (0x40C2, 0x40C3) 223
14.2.32 FOC_UDCPS (0x40C2, 0x40C3) 223
14.2.33 FOC_UQCPS (0x40C4, 0x40C5) 223
14.2.34 FOC_VALP (0x40C4, 0x40C5) 224
14.2.35 FOC_FLUX (0x40C6, 0x40C7) 224
14.2.36 FOC_IC (0x40C6, 0x40C7) 225

14.2.37 FOC_LQ (0x40C8, 0x40C9) 225
14.2.38 FOC_IB (0x40C8, 0x40C9) 225
14.2.39 FOC_LD (0x40CA, 0x40CB) 226
14.2.40 FOC_IA (0x40CA, 0x40CB) 226
14.2.41 FOC_THETA (0x40CC, 0x40CD) 227
14.2.42 FOC_ETHETA (0x40CE, 0x40CF) 227
14.2.43 FOC_EALP (0x40D0, 0x40D1) 228
14.2.44 FOC_EBET (0x40D2, 0x40D3) 228
14.2.45 FOC_EOME (0x40D4, 0x40D5) 228
14.2.46 FOC_UQEX (0x40D6, 0x40D7) 229
14.2.47 FOC_KFG (0x40D6, 0x40D7) 229
14.2.48 FOC_POW (0x40D8, 0x40D9) 230
14.2.49 FOC_EOMEKLPF (0x40D8) 230
14.2.50 FOC_IAMAX (0x40DA, 0x40DB) 231
14.2.51 FOC_IBMAX (0x40DC, 0x40DD) 231
14.2.52 FOC_ICMAX (0x40DE, 0x40DF) 232
14.2.53 FOC_EFREQMAX (0x406F) 232
14.2.54 FOC_DKP (0x4070, 0x4071) 233
14.2.55 FOC_EKP (0x4074, 0x4075) 233
14.2.56 FOC_EKI (0x4076, 0x4077) 233
14.2.57 FOC_KSLIDE (0x4078, 0x4079) 234
14.2.58 FOC_EKLPFMIN (0x407A, 0x407B) 234
14.2.59 FOC_DKI (0x407C, 0x407D) 235
14.2.60 FOC_OMEKLPF (0x407E, 0x407F) 235
14.2.61 FOC_FBASE (0x4080, 0x4081) 236
14.2.62 FOC_EFREQACC (0x4082, 0x4083) 236
14.2.63 FOC_EFREQMIN (0x4084, 0x4085) 237
14.2.64 FOC_EFREQHOLD (0x4086, 0x4087) 237
14.2.65 FOC_EK3 (0x4088, 0x4089) 238
14.2.66 FOC_EK4 (0x408A, 0x408B) 238
14.2.67 FOC_EK1 (0x408C, 0x408D) 239
14.2.68 FOC_EK2 (0x408E, 0x408F) 239

14.2.69 FOC_IDREF (0x4090, 0x4091) 239
14.2.70 FOC_IQREF (0x4092, 0x4093) 240
14.2.71 FOC_QKP (0x4094, 0x4095) 240
14.2.72 FOC_QKI (0x4096, 0x4097) 241
14.2.73 FOC_UDCFLT (0x4098, 0x4099) 241
14.2.74 HFI_KP (0x40E0, 0x40E1) 242
14.2.75 HFI_KI (0x40E2, 0x40E3) 242
14.2.76 OMEGA_HFI (0x40E4, 0x40E5) 242
14.2.77 HFI_ALPHAH (0x40E6, 0x40E7) 243
14.2.78 HFI_BETAH (0x40E8, 0x40E9) 243
14.2.79 FOC_ID0 (0x40EA, 0x40EB) 244
14.2.80 FOC_ID1 (0x40EC, 0x40ED) 244
14.2.81 DELTA_THETA (0x40EE, 0x40EF) 244
14.2.82 FOC_IQ0 (0x40F0, 0x40F1) 245
14.2.83 FOC_IQ1 (0x40F2, 0x40F3) 245
14.2.84 HFI_I1H (0x40F4, 0x40F5) 246
14.2.85 HFI_I2H (0x40F6, 0x40F7) 246
14.2.86 HFI_I3H (0x40F8, 0x40F9) 247
15 Timer1 248
15.1 Timer1 Operations 248
15.1.1 Timer1 Counter Module 249
15.1.1.1 Prescaler 250
15.1.1.2 Base Timer 250
15.1.1.3 Reload Timer 251
15.1.2 Position Detection 252
15.1.2.1 Position Detection Signal 252
15.1.2.2 CMP/GPIO Position Detection Event 252
15.1.2.3 ADC Position Detection Event 253
15.1.2.4 Sampling 254
15.1.2.5 Filtering 254
15.1.3 Write Timing Interrupt 255
15.1.4 Timer1 Interrupt 255

15.2 Square-wave Control Based BLDC Motor 256
15.2.1 Six-step Phase Commutation of Square Wave Control 257
15.2.2 Square Wave Control Working Principle 258
15.2.2.1 60° Commutation Base Time 258
15.2.2.2 Forced Commutation at 60° 259
15.2.2.3 Diode Freewheeling Masking 259
15.2.2.4 Angle of Position Detection to Commutation 260
15.2.2.5 Cycle-by-cycle Current Limiting 260
15.3 Timer1 Registers 261
15.3.1 TIM1_CR0 (0x4068) 261
15.3.2 TIM1_CR1 (0x4069) 262
15.3.3 TIM1_CR2 (0x406A) 263
15.3.4 TIM1_CR3 (0x406B) 263
15.3.5 TIM1_CR4 (0x406C) 264
15.3.6 TIM1_IER (0x406D) 264
15.3.7 TIM1_SR (0x406E) 265
15.3.8 TIM1_BCOR (0x4070, 0x4071) 267
15.3.9 TIM1_CR5 (0x4072) 267
15.3.10 TIM1_DBR1 (0x4074, 0x4075) 268
15.3.11 TIM1_DBR2 (0x4076, 0x4077) 270
15.3.12 TIM1_DBR3 (0x4078, 0x4079) 271
15.3.13 TIM1_DBR4 (0x407A, 0x407B) 272
15.3.14 TIM1_DBR5 (0x407C, 0x407D) 274
15.3.15 TIM1_DBR6 (0x407E, 0x407F) 275
15.3.16 TIM1_DBR7 (0x4080, 0x4081) 276
15.3.17 TIM1_BCNTR (0x4082, 0x4083) 278
15.3.18 TIM1_BCCR (0x4084, 0x4085) 278
15.3.19 TIM1_BARR (0x4086, 0x4087) 279
15.3.20 TIM1_RARR (0x4088, 0x4089) 279
15.3.21 TIM1_RCNTR (0x408A, 0x408B) 280
15.3.22 TIM1_UCOP (0x408C, 0x408D) 280
15.3.23 TIM1_UFLP (0x408E, 0x408F) 281

15.3.24 TIM1__URES (0x4090, 0x4091) 281
15.3.25 TIM1_KRMAX (0x4092) 281
15.3.26 TIM1_KFMIN (0x4093) 282
15.3.27 TIM1_KF (0x4094, 0x4095) 282
15.3.28 TIM1_KR (0x4096, 0x4097) 282
15.3.29 TIM1__ITRIP (0x4098, 0x4099) 283
16 Timer2 284
16.1 Timer2 Instructions 284
16.1.1 Prescaler 284
16.1.2 Reading, Writing and Counting of TIM2__CNTR 285
16.1.3 Output Mode 285
16.1.3.1 Reading and Writing of TIM2__ARR/TIM2__DR 285
16.1.3.2 High/Low Level Output Mode 286
16.1.3.3 PWM Generation 286
16.1.3.4 Interrupt Events 286
16.1.4 Input Signal Filtering and Edge Detection 287
16.1.5 Input Capture Mode 287
16.1.6 Input Counter Mode 289
16.1.7 QEP&RSD Mode 290
16.1.7.1 RSD Comparator Sampling 291
16.1.8 Step Mode 292
16.2 Timer2 Registers 294
16.2.1 TIM2_CR0 (0xA1) 294
16.2.2 TIM2_CR1 (0xA9) 295
16.2.3 TIM2__CNTR (0xAA, 0xAB) 297
16.2.4 TIM2__DR (0xAC, 0xAD) 297
16.2.5 TIM2__ARR (0xAE, 0xAF) 298
17 Timer3/Timer4 299
17.1 Timer3/Timer4 Instructions 299
17.1.1 Prescaler 299
17.1.2 Reading, Writing and Counting of TIMx__CNTR 300
17.1.3 Output Mode 300

17.1.3.1 High-/Low-level Output Mode 300
17.1.3.2 PWM Generation 300
17.1.3.3 Interrupt Event 301
17.1.4 Input Signal Filtering and Edge Detection 301
17.1.5 Input Capture Mode 302
17.1.6 Timer4 FG Output Mode 303
17.2 Timer3/Timer4 Registers 304
17.2.1 TIMx_CR0 (0x9C/0x9E) (x = 3/4) 304
17.2.2 TIMx_CR1 (0x9D/0x9F) (x = 3/4) 305
17.2.3 TIMx__CNTR (0xA2, 0xA3/0x92, 0x93) (x = 3/4) 306
17.2.4 TIMx__DR (0xA4, 0xA5/0x94, 0x95) (x = 3/4) 307
17.2.5 TIMx__ARR (0xA6, 0xA7/0x96, 0x97) (x = 3/4) 307
18 Systick 308
18.1 Systick Instructions 308
18.2 Systick Registers 308
18.2.1 DRV_SR (0x4061) 308
18.2.2 SYST_ARR (0x4064, 0x4065) 309
19 Driver 310
19.1 Driver Instructions 310
19.1.1 Driver Introduction 310
19.1.2 Output Control Module 310
19.1.2.1 Count and Compare Module 311
19.1.2.2 Deadtime Module 312
19.1.2.3 Enable and Polarity of Output Signals 312
19.1.2.4 Interrupt 313
19.1.2.4.1 Compare Match Interrupt 313
19.1.2.4.2 FG Interrupt 314
19.1.3 6N Pre-driver Mode 314
19.2 Driver Registers 315
19.2.1 PI_CR (0xF9) 315
19.2.2 DRV_CR (0x4062) 316
19.2.3 DRV_SR (0x4061) 317

19.2.4 DRV_OUT (0xF8) 318
19.2.5 DRV_CMR (0x405C, 0x405D) 319
19.2.6 DRV_ARR (0x405E, 0x405F) 320
19.2.7 DRV_COMR (0x405A, 0x405B) 321
19.2.8 DRV_DR (0x4058, 0x4059) 321
19.2.9 DRV_DTR (0x4060) 322
19.2.10 DRV_CNTR (0x4066, 0x4067) 322
20 WDT 324
20.1 WDT Notes 324
20.2 WDT Operations 324
20.3 WDT Registers 325
20.3.1 WDT_CR (0x4026) 325
20.3.2 WDT_ARR (0x4027) 325
20.3.3 CCFG1 (0x401E) 325
21 RTC 327
21.1 Functional Block Diagram 327
21.2 RTC Operations 327
21.3 RTC Registers 327
21.3.1 RTC_TM (0x402C, 0x402D) 327
21.3.2 RTC_STA (0x402E) 328
22 IO 329
22.1 IO Introduction 329
22.2 IO Operations 329
22.3 IO Registers 330
22.3.1 P0_OE (0xFC) 330
22.3.2 P1_OE (0xFD) 330
22.3.3 P2_OE (0xFE) 330
22.3.4 P3_OE (0xFF) 331
22.3.5 P4_OE (0xE9) 331
22.3.6 P1_AN (0x4050) 331
22.3.7 P2_AN (0x4051) 332
22.3.8 P3_AN (0x4052) 332

22.3.9 P4_AN(0x4047) 333
22.3.10 P0_PU (0x4053) 333
22.3.11 P1_PU (0x4054) 333
22.3.12 P2_PU (0x4055) 334
22.3.13 P3_PU (0x4056) 334
22.3.14 P4_PU (0x4057) 334
22.3.15 PH_SEL (0x404C) 335
22.3.16 PH_SEL1 (0x404D) 336
22.3.17 PH_SEL2 (0x4049) 336
22.3.18 P0 (0x80) 337
22.3.19 P1 (0x90) 337
22.3.20 P2 (0xA0) 338
22.3.21 P3 (0xB0) 338
22.3.22 P4 (0xB8) 338
23 System Clock 340
23.1 Introduction 340
23.2 Clock Operations 340
23.2.1 Internal Fast Clock 340
23.2.2 External Fast Clock 340
23.2.2.1 External Clock Input Mode 341
23.2.3 Internal Slow Clock 341
23.2.4 External Slow Clock 342
23.3 MCD Module and Two-speed Mode 342
23.4 Clock Registers 344
23.4.1 OSC_CFG (0xF1) 344
23.4.2 CCFG3 (0x401C) 344
23.5 Clock Calibration 345
23.5.1 Introduction 345
23.5.2 Clock Calibration Registers 346
23.5.2.1 CAL_CR0 (0x4044) CAL_CR1 (0x4045) 346
24 Temperature Sensor 347
24.1 Temperature Sensor Registers 348

24.1.1 TSD_CR (0x402F) 348
24.1.2 TSEN_DR (0x4048) 348
25 ADC 349
25.1 ADC Introduction 349
25.2 ADC Block Diagram 350
25.3 ADC Operations 351
25.3.1 Sequential Sampling Mode 351
25.3.2 Triggered Sampling Mode 351
25.3.3 Output Data Format 352
25.4 ADC Registers 353
25.4.1 ADC_CR (0x4039) 353
25.4.2 ADC_MASK (0x4036, 0x4037) 354
25.4.3 PH_SEL2 (0x4049) 355
25.4.4 DAC_CR (0x4035) 355
25.4.5 ADC_SCYC (0x4038) 355
25.4.6 ADC0_DR (0x0FD8, 0x0FD9) 356
25.4.7 ADC1_DR (0x0FDA, 0x0FDB) 356
25.4.8 ADC2_DR (0x0FDC, 0x0FDD) 357
25.4.9 ADC3_DR (0x0FDE, 0x0FDF) 357
25.4.10 ADC4_DR (0x0FE0, 0x0FE1) 358
25.4.11 ADC5_DR (0x0FE2, 0x0FE3) 358
25.4.12 ADC6_DR (0x0FE4, 0x0FE5) 359
25.4.13 ADC7_DR (0x0FE6, 0x0FE7) 359
25.4.14 ADC8_DR (0x0FE8, 0x0FE9) 360
25.4.15 ADC9_DR (0x0FEA, 0x0FEB) 360
25.4.16 ADC10_DR (0x0FEC, 0x0FED) 361
25.4.17 ADC11_DR (0x0FEE, 0x0FEF) 361
25.4.18 ADC12_DR (0x0FF0, 0x0FF1) 362
25.4.19 ADC13_DR (0x0FF2, 0x0FF3) 362
25.4.20 ADC14_DR (0x0FF4, 0x0FF5) 363
25.4.21 ADC15_DR (0x0FF6, 0x0FF7) 363
26 DAC 365

26.1 DAC Introduction 365
26.2 DAC0 365
26.3 DAC1 366
26.4 DAC Registers 367
26.4.1 DAC_CR (0x4035) 367
26.4.2 DAC0_DR (0x404B) 367
26.4.3 DAC1_DR (0x404A) 368
27 DMA 369
27.1 DMA Instructions 369
27.2 DMA Registers 370
27.2.1 DMA0_CR0 (0x403A) 370
27.2.2 DMA1_CR0 (0x403B) 371
27.2.3 DMA0_LEN (0x403C) 372
27.2.4 DMA0_BA (0x403E, 0x403F) 373
27.2.5 DMA1_LEN (0x403D) 373
27.2.6 DMA1_BA (0x4040, 0x4041) 374
28 VREF 375
28.1 VREF Instructions 375
28.2 VREF Register 376
28.2.1 VREF_VHALF_CR (0x404F) 376
29 VHALF 377
29.1 VHALF Instructions 377
29.2 VHALF Register 377
30 Operational Amplifiers 378
30.1 Operational Amplifier Introduction 378
30.2 Operational Amplifier Instructions 378
30.2.1 Bus Current Sampling Operational Amplifier (AMP0) 378
30.2.1.1 AMP0 Normal Mode 378
30.2.1.2 AMP0 PGA Differential Input Mode 379
30.2.1.3 AMP0 PGA Single-ended Input Mode 379
30.2.2 Phase Current Operational Amplifier (AMP1/AMP2) 380

30.2.2.1 AMP1 Normal Mode 380
30.2.2.2 AMP2 Normal Mode 380
30.2.2.3 AMP1 PGA Differential Input Mode 381
30.2.2.4 AMP2 PGA Differential Input Mode 381
30.2.2.5 AMP1 PGA Single-ended Input Mode 382
30.2.2.6 AMP2 PGA Single-ended Input Mode 382
30.2.3 Operational Amplifier AMP3 383
30.2.3.1 AMP3 Normal Mode 383
30.2.3.2 AMP3 PGA Differential Input Mode 383
30.3 Operational Amplifier Registers 384
30.3.1 AMP_CR0 (0x404E) 384
30.3.2 AMP_CR1 (0x4034) 385
30.3.3 AMP_CR2 (0x4046) 386
31 Comparators 387
31.1 Comparator Operations 387
31.1.1 CMP3 387
31.1.1.1 Overcurrent Protection (OCP) 389
31.1.1.2 Cycle-by-cycle Current Limiting 390
31.1.2 CMP4 391
31.1.3 Comparator Group (CMPG) 392
31.1.4 Comparator Sampling 395
31.1.5 Comparator Output 396
31.2 Comparator Registers 397
31.2.1 CMP_CR0 (0xD5) 397
31.2.2 CMP_CR1 (0xD6) 397
31.2.3 CMP_CR2 (0xDA) 398
31.2.4 CMP_CR3 (0xDC) 400
31.2.5 CMP_CR4 (0xE1) 401
31.2.6 CMP_SAMR (0x40AD) 402
31.2.7 CMP_SR (0xD7) 403
31.2.8 HALL_CR (0xE2) 404
31.2.9 EVT_FILT (0xD9) 404

32 Power Supply 406
32.1 LDO 406
32.1.1 LDO Operations 406
32.2 Low Voltage Detection (LVD) 407
32.2.1 LVD Introduction 407
32.2.2 LVD Operations 407
32.2.3 LVD Registers 409
32.2.3.1 LVSR (0xDB) 409
32.2.3.2 CCFG2 (0x401D) 410
33 Flash 411
33.1 Flash Introduction 411
33.2 Flash Operations 411
33.3 Flash Registers 412
33.3.1 FLA_CR (0x85) 412
33.3.2 FLA_KEY (0x84) 413
34 CRC 414
34.1 Functional Block Diagram 414
34.2 CRC16 Polynomial 414
34.3 CRC16 Logic Diagram 415
34.4 CRC Operations 415
34.4.1 CRC Calculation of Single Byte 415
34.4.2 CRC Calculation of ROM Sector 415
34.5 CRC Registers 417
34.5.1 CRC_CR (0x4022) 417
34.5.2 CRC_DIN (0x4021) 418
34.5.3 CRC_DR (0x4023) 418
34.5.4 CRC_BEG (0x4024) 418
34.5.5 CRC_CNT (0x4025) 419
35 Sleep Mode 420
35.1 Introduction 420
35.2 Sleep Mode Register 421

35.2.1 PCON(0x87) 421
36 Code Protection 422
36.1 Introduction 422
36.2 Operating Instructions 422
37 Magnetoresistive (MR) Sensor 424
37.1 Introduction 424
37.2 Instructions 424
37.3 Functional Block Diagram 426
37.4 Application Recommendation 426
38 Revision History 427

Explanation of Symbols

- The symbol "[ ]" following a register indicates a bit in the register. For example, ABCD[XY] indicates the XY bit in ABCD register.
- The symbol "x" in a register name indicates similar registers. For example, TIMx_CR0 indicates TIM3_CR0 and TIM4_CR0.
- [m:n] indicates a range of bits. For example, [3:0] means the bits from bit3 to bit0.
- Pm.n indicates the n^{th} port of the Portm. P0.0 indicates the 0^{th} port of Port0.
- Register read and write symbols:
  - R: Read only
  - W: Write only
  - R/W: Read/write
  - W0: Only 0 can be written
  - W1: Only 1 can be written
- The symbol "-" indicates an 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
AMR	Anisotropic Magneto Resistance
BEMF	Back Electromotive Force
BLDC	Brushless Direct Current
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
DAC	Digital to Analog Convertor
DMA	Direct Memory Access
FG	Frequency Generator
FICE	Fortior Interactive Connectivity Establishment
FOC	Field Oriented Control
FOSC	Fast Oscillator
GPIO	General Purpose Input Output
IC	Integrated Circuit
I^2C	Inter Integrated Circuit
IDE	Integrated Development Environment
IRAM	Internal RAM
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:
  - High-voltage single-power supply mode: When VCC_MODE = 0, external power supply 6.5V ~ 28V is connected to VCC pin, and internal LDO supplies VDD5 voltage
- Dual core: 8051 core and ME core
- An instruction cycle mostly takes 1 or 2 system clock cycle(s)
- 32kB Flash ROM with CRC, self-program and code protection
- 256 bytes IRAM and 3.75k bytes XRAM
- ME: Core integrating PID module, FOC module, MDU auxiliary computing module and LPF module
- 16 interrupt sources with 4 configurable priority levels
- 31*GPIOs
- Timers:
  - Timer1: Timer supporting square-wave drive timing control, automatic commutation, cycle-by-cycle current limiting and Hall/BEMF-based position sensing
  - Timer2: Timer supporting PWM output, measurement of duty cycle and period of input PWM wave, measurement of the time of set PWM wave numbers, QEP decoding, tailwind/headwind detection, and rotation direction and speed detection of step motor
  - Timer3/Timer4: Timers supporting PWM output, and measurement of duty cycle and period of input PWM wave. Timer4 supports FG generation and Timer3 supports up to 48MHz input
  - Systick Timer
  - RTC
- Communication interfaces:
  - 1*SPI
  - 1*I^2C

- 2*UARTs, supporting single-wire mode
- 1*LIN, supporting single-wire mode
- 1*CAN
- Dual-channel DMA: supporting data transmission via I^2C/SPI/UART/LIN

- Analog peripherals:
  - 12-bit ADC, operating with 1μs conversion time and internal VREF or external VREF as reference voltage
  - Number of ADC channels: 16. Among them, AD15 is used for internal sampling
  - Internal VREF. 3V, 4V, 4.5V or VDD5 can be selected as the internal reference
  - Internal VHALF, with VREF/2, 1/4 VREF, 1/8 VREF or 25/64 VREF as the internal reference
  - Four standalone operational amplifier(s), where PGA is configurable
  - 3-channel analog comparator
  - DAC: Single-channel 9-bit, single-channel 6-bit
- Drive Type:
  - Built-in MOSFET driver: 6N pre-driver output
- FOC module supports single/dual/triple-shunt current sampling
- System clock
  - Built-in 24MHz high-speed RC oscillator
  - Built-in 32.8kHz low-speed RC oscillator
  - Support external 24MHz crystal clock
  - Support external 32768Hz crystal clock
- WDT
- LVD
- Temperature sensor
- Two-wire FICE protocol based in-circuit emulation

1.2 Applications

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

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

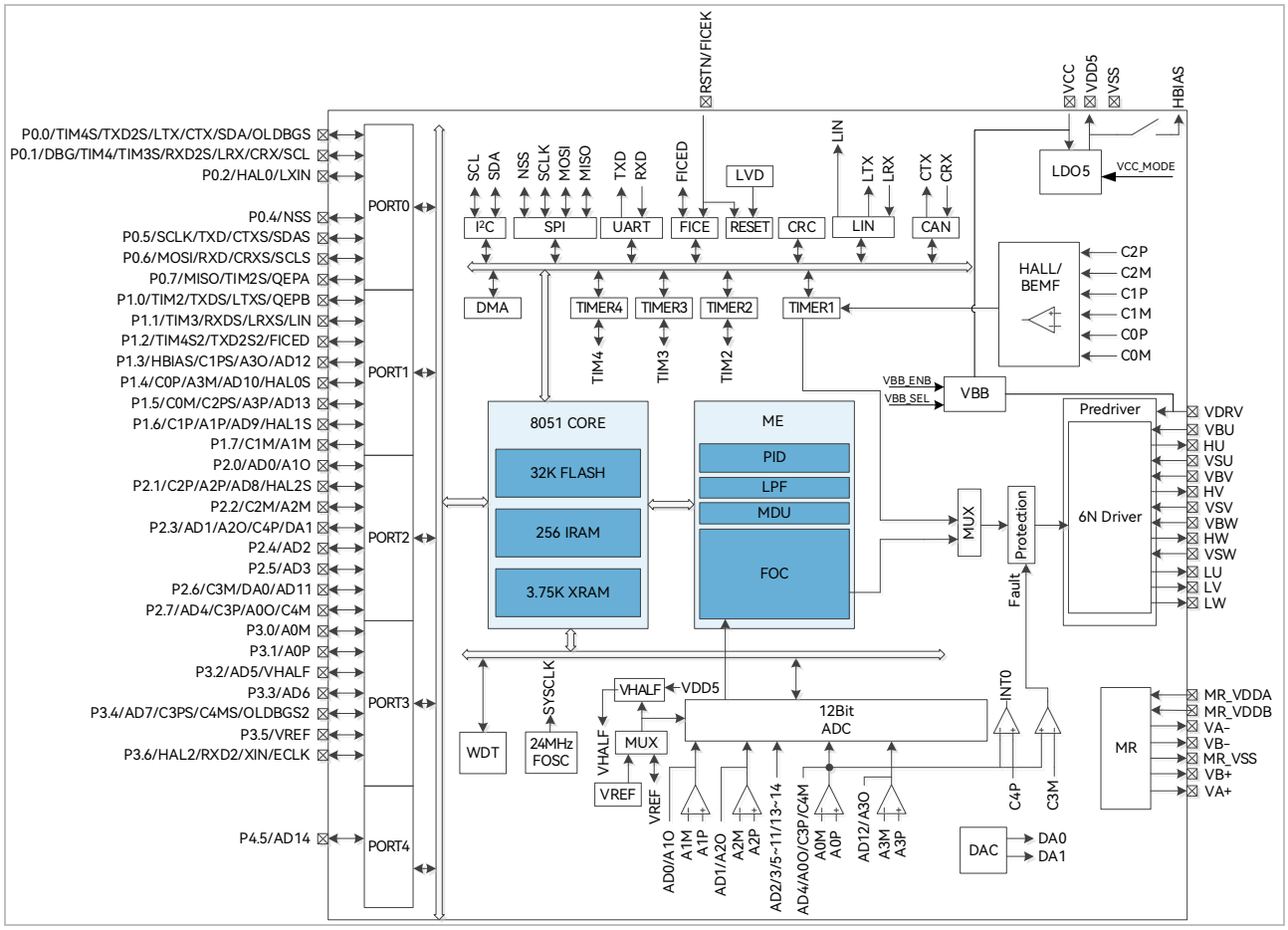
1.3 Overview

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

1.4 Functional Block Diagram

1.4.1 FU6367Q

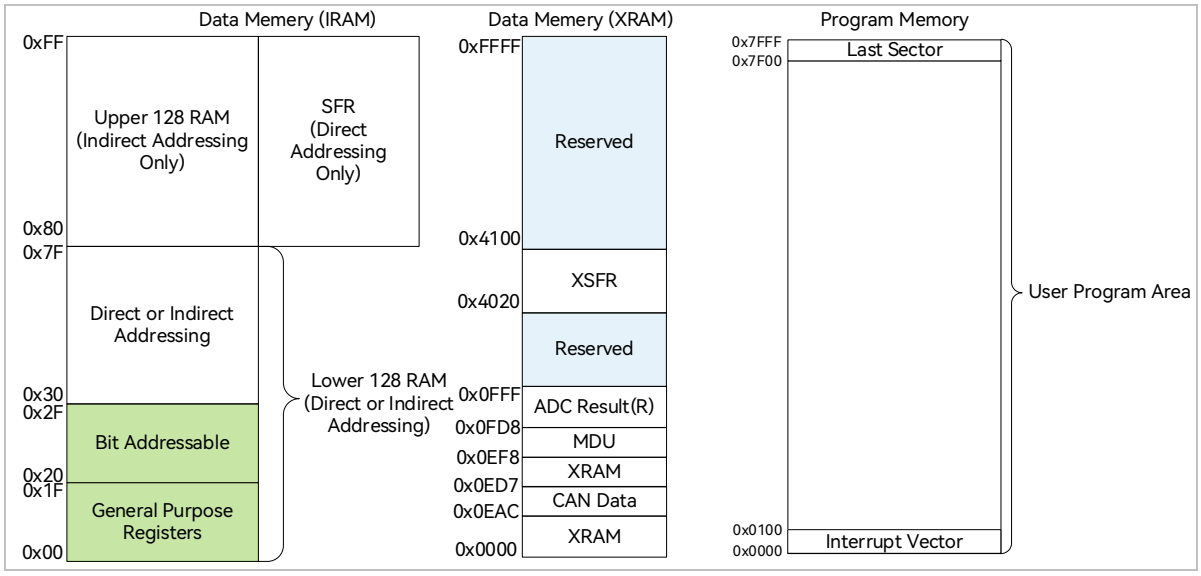
Figure 1-1 Functional Block Diagram of FU6367Q



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 0x7FFF to store control programs to store control programs.

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

1.5.2 Data Memory

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

The External Data Memory is addressed from 0x0000 to 0xFFFF, which can be accessed only with MOVX instructions. It comprises XRAM (0x0000 ~ 0x0EF7), extended control register space (0x4020 ~ 0x40FF), MDU register space (0x0EF8 ~ 0x0FD7), CAN message data store (0x0EAC ~ 0x0ED7) and ADC result memory area (0x0FD8 ~ 0x0FFF).

The Internal Data Memory is addressed from 0x00 to 0xFF. Locations 0x00 ~ 0x1F are addressable as 4 banks of general purpose registers, each bank consisting of 8 registers, adding up to 32 registers. Locations 0x20 ~

0x7F are used for general purpose RAM memory, supporting direct and indirect addressing. Locations (0x20 ~ 0x2F) are 16-bit addressable. When locations 0x80 ~ 0xFF are accessed by indirect addressing, it points to RAM. When locations 0x80 ~ 0xFF are accessed by direct addressing, it points to SFR.

1.5.3 SFR

Table 1-1 SFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xF8	DRV_OUT	PI_CR			P0_OE	P1_OE	P2_OE	P3_OE
0xF0	B	OSC_CFG						
0xE8		P4_OE						
0xE0	ACC	CMP_CR4	HALL_CR					
0xD8	UT2_CR	EVT_FILT	CMP_CR2	LVSR	CMP_CR3	CAN_ER_LIM	CAN_TEC	CAN_REC
0xD0	PSW	P1_IE	P1_IF	P4_IE	P4_IF	CMP_CR0	CMP_CR1	CMP_SR
0xC8		RST_SR	CAN_FMR0	CAN_FMR1	CAN_FMR2	CAN_FMR3	CAN_ALC	CAN_ESR
0xC0		MDU_CR	CAN_BTR1	CAN_BTR2	CAN_FIR0	CAN_FIR1	CAN_FIR2	CAN_FIR3
0xB8	P4	LIN_CR	LIN_SR	LIN_CSR	LIN_ID	LIN_SIZE	LIN_BAUDL	LIN_BAUDH
0xB0	P3	CAN_CR0	CAN_CR1	CAN_SR	CAN_IER	CAN_CR2	CAN_IFR	CAN_BTR0
0xA8	IE	TIM2_CR1	TIM2__CNTRL	TIM2__CNTRH	TIM2__DRL	TIM2__DRH	TIM2__ARRL	TIM2__ARRH
0xA0	P2	TIM2_CR0	TIM3__CNTRL	TIM3__CNTRH	TIM3__DRL	TIM3__DRH	TIM3__ARRL	TIM3__ARRH
0x98	UT_CR	UT_DR	UT_BAUDL	UT_BAUDH	TIM3_CR0	TIM3_CR1	TIM4_CR0	TIM4_CR1
0x90	P1		TIM4__CNTRL	TIM4__CNTRH	TIM4__DRL	TIM4__DRH	TIM4__ARRL	TIM4__ARRH
0x88	TCON	UT2_DR	IP0	IP1	IP2	IP3		
0x80	P0	SP	DPL	DPH	FLA_KEY	FLA_CR		PCON

Note:

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

1.5.4 XSFR

Table 1-2 XSFR Address Mapping

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x40F8	HFI_I3H	HFI_I3L						
0x40F0	FOC_IQ0H	FOC_IQ0L	FOC_IQ1H	FOC_IQ1L	HFI_I1H	HFI_I1L	HFI_I2H	HFI_I2L
0x40E8	HFI_BETAH	HFI_BETAL	FOC_ID0H	FOC_IDOL	FOC_ID1H	FOC_ID1L	DELTA_THETAH	DELTA_THETAL
0x40E0	HFI_KPH	HFI_KPL	HFI_KIH	HFI_KIL	OMEGA_HFIH	OMEGA_HFIL	HFI_ALPHAH	HFI_ALPHAL
0x40D8	FOC_POWH FOC_EOMEKLPF	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_UQEXH FOC_KFGH	FOC_UQEXL FOC_KFGL
0x40C8	FOC_IBH FOC_LQH	FOC_IBL FOC_LQL	FOC_IAH FOC_LDH	FOC_IAL FOC_LDL	FOC_THETAH	FOC_THETAL	FOC_ETHETAH	FOC_ETHETAL
0x40C0	FOC_IBETH FOC_IQ_LPFK	FOC_IBETL FOC_ID_LPFK	FOC_VBETH FOC_UDCPSH	FOC_VBETL FOC_UDCPSL	FOC_VALPH FOC_UQCPSH	FOC_VALPL FOC_UQCPSL	FOC_ICH FOC_FLUXH	FOC_ICL FOC_FLUXL
0x40B8	FOC_UDH	FOC_UDL	FOC_UQH	FOC_UQL	FOC_IDH	FOC_IDL	FOC_IQH	FOC_IQL
0x40B0	FOC_DMAXH FOC_OMEESTH	FOC_DMAXL FOC_OMEESTL	FOC_DMINH	FOC_DMINL	FOC_QMAXH	FOC_QMAXL	FOC_QMINH	FOC_QMINL
0x40A8	FOC_RTHESTEPH	FOC_RTHESTEPL	FOC_RTHEACCH FOC_EOMELPFH	FOC_RTHEACCL FOC_EOMELPFL	FOC_RTHECNT	FOC_THECOR CMP_SAMR	FOC_THECOMPH FOC_EMFH	FOC_THECOMPL FOC_EMFL
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			FOC_CR5	FOC_CR4	FOC_CR3	FOC_CR0
0x4090	FOC_IDREFH	FOC_IDREFL	FOC_IQREFH	FOC_IQREFL	FOC_QKPH	FOC_QKPL	FOC_QKIH	FOC_QKIL

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
	TIM1__URESH	TIM1__URESL	TIM1_KRMAX	TIM1_KFMIN	TIM1_KFH	TIM1_KFL	TIM1_KRH	TIM1_KRL
0x4088	FOC_EK3H	FOC_EK3L	FOC_EK4H	FOC_EK4L	FOC_EK1H	FOC_EK1L	FOC_EK2H	FOC_EK2L
	TIM1__RARRH	TIM1__RARRL	TIM1__RCNTRH	TIM1__RCNTRL	TIM1__UCOPH	TIM1__UCOPL	TIM1__UFLPH	TIM1__UFLPL
0x4080	FOC_FBASEH	FOC_FBASEL	FOC_EFREQACCH	FOC_EFREQACCL	FOC_EFREQMINH	FOC_EFRQMINL	FOC_EFREQHOLDH	FOC_EFREQHOLDL
	TIM1_DBR7H	TIM1_DBR7L	TIM1__BCNTRH	TIM1__BCNTRL	TIM1__BCCRH	TIM1__BCCRL	TIM1__BARRH	TIM1__BARRL
0x4078	FOC_KSLIDEH	FOC_KSLIDEL	FOC_EKLPFMINH	FOC_EKLPFMINL	FOC_DKIH	FOC_DKIL	FOC_OMEKLPFH	FOC_OMEKLPFL
	TIM1_DBR3H	TIM1_DBR3L	TIM1_DBR4H	TIM1_DBR4L	TIM1_DBR5H	TIM1_DBR5L	TIM1_DBR6H	TIM1_DBR6L
0x4070	TIM1_BCORH	TIM1_BCORL	TIM1_CR5		FOC_EKPH	FOC_EKPL	FOC_EKIH	FOC_EKIL
	FOC_DKPH	FOC_DKPL			TIM1_DBR1H	TIM1_DBR1L	TIM1_DBR2H	TIM1_DBR2L
0x4068	TIM1_CR0	TIM1_CR1	TIM1_CR2	TIM1_CR3	TIM1_CR4	TIM1_IER	TIM1_SR	FOC_EFREQMAX
0x4060	DRV_DTR	DRV_SR	DRV_CR		SYST_ARRH	SYST_ARRL	DRV__CNTRH	DRV__CNTRL
0x4058	DRV_DRH	DRV_DRL	DRV_COMRH	DRV_COMRL	DRV_CMRH	DRV_CMRL	DRV_ARRH	DRV_ARRL
0x4050	P1_AN	P2_AN	P3_AN	P0_PU	P1_PU	P2_PU	P3_PU	P4_PU
0x4048	TSEN_DR	PH_SEL2	DAC1_DR	DAC_DR	PH_SEL	PH_SEL1	AMP_CR0	VREF_VHALF_CR
0x4040	DMA1_BAH	DMA1_BAL	UT2_BAUDH	UT2_BAUDL	CAL_CR0	CAL_CR1	AMP_CR2	P4_AN
0x4038	ADC_SCYC	ADC_CR	DMA0_CR0	DMA1_CR0	DMA0_LEN	DMA1_LEN	DMA0_BAH	DMA0_BAL
0x4030	SPI_CR0	SPI_CR1	SPI_CLK	SPI_DR	AMP_CR1	DAC_CR	ADC_MASKH	ADC_MASKL
0x4028	I2C_CR	I2C_ID	I2C_DR	I2C_SR	RTC_TMH	RTC_TML	RTC_STA	TSD_CR
0x4020		CRC_DIN	CRC_CR	CRC_DR	CRC_BEG	CRC_CNT	WDT_CR	WDT_ARR
0x0FF8								
0x0FF0	AD12_DRH	AD12_DRL	AD13_DRH	AD13_DRL	AD14_DRH	AD14_DRL	AD15_DRH	AD15_DRL
0x0FE8	AD8_DRH	AD8_DRL	AD9_DRH	AD9_DRL	AD10_DRH	AD10_DRL	AD11_DRH	AD11_DRL
0x0FE0	AD4_DRH	AD4_DRL	AD5_DRH	AD5_DRL	AD6_DRH	AD6_DRL	AD7_DRH	AD7_DRL
0x0FD8	AD0_DRH	AD0_DRL	AD1_DRH	AD1_DRL	AD2_DRH	AD2_DRL	AD3_DRH	AD3_DRL
0x0FD0	LPF0_K		LPF0_X		LPF0_YH		LPF0_YL	

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x0FC8	LPF1_K		LPF1_X		LPF1_YH		LPF1_YL	
0x0FC0	PI0_UKH		PI0_UKL		PI0_UKMAX		PI0_UKMIN	
0x0FB8	PI0_KP		PI0_EK1		PI0_EK		PI0_KI	
0x0FB0	PI1_UKH		PI1_UKL		PI1_UKMAX		PI1_UKMIN	
0x0FA8	PI1_KP		PI1_EK1		PI1_EK		PI1_KI	
0x0FA0	MUL0_MA		MUL0_MB		MUL0_MCH		MUL0_MCL	
0x0F98	MUL1_MA		MUL1_MB		MUL1_MCH		MUL1_MCL	
0x0F90	DIV0_DB		DIV0_DQH		DIV0_DQL		DIV0_DR	
0x0F88	DIV1_DQL		DIV1_DR		DIV0_DAH		DIV0_DAL	
0x0F80	DIV1_DAH		DIV1_DAL		DIV1_DB		DIV1_DQH	
0x0F78	LPF2_K		LPF2_X		LPF2_YH		LPF2_YL	
0x0F70	LPF3_K		LPF3_X		LPF3_YH		LPF3_YL	
0x0F68	PI2_UKMAX		PI2_UKMIN		PI2_KD		PI2_EK2	
0x0F60	PI2_EK		PI2_KI		PI2_UKH		PI2_UKL	
0x0F58	PI3_KD		PI3_EK2		PI2_KP		PI2_EK1	
0x0F50	PI3_UKH		PI3_UKL		PI3_UKMAX		PI3_UKMIN	
0x0F48	PI3_KP		PI3_EK1		PI3_EK		PI3_KI	
0x0F40	MUL2_MA		MUL2_MB		MUL2_MCH		MUL2_MCL	
0x0F38	MUL3_MA		MUL3_MB		MUL3_MCH		MUL3_MCL	
0x0F30	DIV2_DB		DIV2_DQH		DIV2_DQL		DIV2_DR	
0x0F28	DIV3_DQL		DIV3_DR		DIV2_DAH		DIV2_DAL	
0x0F20	DIV3_DAH		DIV3_DAL		DIV3_DB		DIV3_DQH	
0x0F18	SCAT0_SIN		SCAT0_THE		SCAT0_RES1		SCAT0_RES2	
0x0F10	SCAT1_THE		SCAT1_RES1		SCAT1_RES2		SCAT0_COS	
0x0F08	SCAT2_RES1		SCAT2_RES2		SCAT1_COS		SCAT1_SIN	

Addr	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0x0F00	SCAT3_RES2		SCAT2_COS		SCAT2_SIN		SCAT2_THE	
0x0EF8	SCAT3_COS		SCAT3_SIN		SCAT3_THE		SCAT3_RES1	
0x0ED0	CAN_TXDR2	CAN_TXDR1	CAN_TXDR0	CAN_TXID3	CAN_TXID2	CAN_TXID1	CAN_TXID0	CAN_TXCR
0x0EC8	CAN_RX0ID1	CAN_RX0ID0	CAN_RX0CR	CAN_TXDR7	CAN_TXDR6	CAN_TXDR5	CAN_TXDR4	CAN_TXDR3
0x0EC0	CAN_RX0DR5	CAN_RX0DR4	CAN_RX0DR3	CAN_RX0DR2	CAN_RX0DR1	CAN_RX0DR0	CAN_RX0ID3	CAN_RX0ID2
0x0EB8	CAN_RX1DR0	CAN_RX1ID3	CAN_RX1ID2	CAN_RX1ID1	CAN_RX1ID0	CAN_RX1CR	CAN_RX0DR7	CAN_RX0DR6
0x0EB0	CAN_RX2CR	CAN_RX1DR7	CAN_RX1DR6	CAN_RX1DR5	CAN_RX1DR4	CAN_RX1DR3	CAN_RX1DR2	CAN_RX1DR1
0x0EA8	CAN_RX2DR3	CAN_RX2DR2	CAN_RX2DR1	CAN_RX2DR0	CAN_RX2ID3	CAN_RX2ID2	CAN_RX2ID1	CAN_RX2ID0
0x0EA0					CAN_RX2DR7	CAN_RX2DR6	CAN_RX2DR5	CAN_RX2DR4

Note:

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

2 Pin Definitions

The IO types are defined as follows:

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

2.1 FU6367Q QFN56 Pins

Table 2-1 FU6367Q QFN56 Pin Descriptions

Pin	FU6367Q QFN56	IO Type	Description
VSU	1	P	6N pre-driver U-phase input, as GND reference for U-phase high-side bootstrap
HU	2	DO	6N pre-driver high-side U-phase PWM output
VBU	3	P	6N pre-driver high-side U-phase bootstrap power supply
VSV	4	P	6N pre-driver V-phase input, as GND reference for V-phase high-side bootstrap
HV	5	DO	6N pre-driver high-side V-phase PWM output
VBV	6	P	6N pre-driver high-side V-phase bootstrap power supply
VSW	7	P	6N pre-driver W-phase input, as GND reference for W-phase high-side bootstrap
HW	8	DO	6N pre-driver high-side W-phase PWM output
VBW	9	P	6N pre-driver high-side W-phase bootstrap power supply
MR_VDDA	10	P	Power supply of angle sensor
MR_VDDB	11	P	Power supply of angle sensor
VA+	12	AO	Positive differential sine output of angle sensor
VB-	13	AO	Negative differential cosine output of angle sensor
MR_VSS	14	P	Ground for angle sensor

Pin	FU6367Q QFN56	IO Type	Description
VB+	15	AO	Positive differential cosine output of angle sensor
VA-	16	AO	Negative 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 4.7μF or above. › High-voltage single-power supply mode: When VCC_MODE = 0, external power supply 6.5V ~ 28V is connected to VCC pin, and internal LDO supplies VDD5 voltage.
VSS	18	P	Ground
VDD5	19	P	Internal LDO outputs 5V power supply, with an external capacitor of 1μF or above Power input or 5V internal LDO output, which is determined by VCC_MODE, with an external filter capacitor of 1μF or above. See descriptions on VCC pin for details.
RSTN/ FICEK	20	DI/ DI	Input of external reset, with built-in pull-up resistor FICE SCL
P4.5/ AD14	21	DB/ AI	GPIO, configurable as INT1 input Input of ADC channel 14
P1.2/ TIM4S2/ TXD2S2 FICED	22	DB/ DB/ DO/ DB	GPIO, configurable as INT1 input Timer4 input/output after function switching UART2 TXD output after function switching FICE SDA
P1.3/ HBIAS/ C1PS/ A3O/ AD12	23	DB/ DO/ AI/ DO/ AI	GPIO, configurable as INT1 input Hall bias power supply, internally connected to VDD5 via a switch to output large current CMP1 positive input after function switching AMP3 output Input of ADC channel 12
P1.4/ C0P/ A3M/ AD10/ HAL0S	24	DB/ AI/ AI/ AI/ DI	GPIO, configurable as INT1 input CMP0 positive input AMP3 negative input Input of ADC channel 10 Hall-IC0 logic level input after function switching
P1.5/ C0M/ C2PS/ A3P/ AD13	25	DB/ AI/ AI/ AI/ AI/	GPIO, configurable as INT1 input CMP0 negative input CMP2 positive input after function switching AMP3 positive input Input of ADC channel 13
P1.6/ C1P/ A1P/ AD9/	26	DB/ AI/ AI/ AI/	GPIO, configurable as INT1 input CMP1 positive input AMP1 positive input Input of ADC channel 9

Pin	FU6367Q QFN56	IO Type	Description
HAL1S		DI	Hall-IC1 logic level input after function switching
P1.7/ C1M/ A1M	27	DB/ AI/ AI	GPIO, configurable as INT1 input CMP1 negative input AMP1 negative input
P2.0/ AD0/ A1O	28	DB/ AI/ AO	GPIO Input of ADC channel 0 AMP1 output
P2.1/ C2P/ A2P/ AD8/ HAL2S	29	DB/ AI/ AI/ AI/ DI	GPIO CMP2 positive input AMP2 positive input Input of ADC channel 8 Hall-IC2 logic level input after function switching
P2.2/ C2M/ A2M	30	DB/ AI/ AI	GPIO CMP2 negative input AMP2 negative input
P2.3/ AD1/ A2O/ C4P/ DA1	31	DB/ AI/ AO/ AI/ DO	GPIO Input of ADC channel 1 AMP2 output CMP4 positive input DAC1 output, without Buffer output
P2.4/ AD2	32	DB/ AI	GPIO Input of ADC channel 2 for bus voltage sampling
P2.5/ AD3	33	DB/ AI	GPIO Input of ADC channel 3
P2.6/ C3M/ DA0/ AD11	34	DB/ AI/ AO/ AI	GPIO CMP3 negative input DAC0 output, without Buffer output Input of ADC channel 11
P2.7/ AD4/ C3P/ A0O/ C4M	35	DB/ AI/ AI/ AO/ AI	GPIO Input of ADC channel 4 for bus current sampling CMP3 positive input AMP0 output CMP4 negative input
P3.0/ A0M	36	DB/ AI	GPIO AMP0 negative input
P3.1/ A0P	37	DB/ AI	GPIO AMP0 positive input
P3.2/ AD5/ VHALF	38	DB/ AI/ AO	GPIO Input of ADC channel 5 VREF/2 output with a 1μF external capacitor

Pin	FU6367Q QFN56	IO Type	Description
P3.3/ AD6	39	DB/ AI	GPIO Input of ADC channel 6. When AMP_CR0[CP_EN] = 1 and AMP_CR0[CP_SEL] = 1, AMP0 output (P2.7) is delivered to P3.3 via a 16kΩ internal resistor. P3.3 is connected with a 1μF external capacitor for bus average current sampling.
P3.4/ AD7/ C3PS/ C4MS/ OLDBGS2	40	DB/ AI/ AI/ AI/ DO	GPIO Input of ADC channel 7. When AMP_CR0[CP_EN] = 1, AMP0 output (P2.7) is delivered to P3.4 via a 16kΩ internal resistor. P3.4 is connected with a 1μF external capacitor for bus average current sampling. CMP3 positive input after function switching CMP4 negative input after function switching SPI debug output after the second function switching
P3.5/ VREF	41	DB/ AB	GPIO ADC external VREF input or internal VREF output, with a 1μF ~ 4.7μF external capacitor
P3.6/ HAL2/ RXD2/ XIN/ ECLK	42	DB/ DI/ DB/ AI/ DI	GPIO, configurable as INT1 input Hall-IC2 logic level input UART2 RXD input in two-wire mode or TXD output/RXD input in single-wire mode Crystal clock input, connected to a 24MHz external crystal clock External fast clock input
P0.0/ TIM4S/ TXD2S/ LTX/ CTX/ SDA/ OLDBGS	43	DO/ DB/ DO/ DO/ DO/ DO/ DO	GPIO, configurable as INT0 input Timer4 input/output after function switching UART2 TXD output after function switching LIN TXD output CAN TXD output I²C SDA, configured as open-drain output SPI debug output after function switching
P0.1/ DBG/ TIM4/ TIM3S/ RXD2S/ LRX/ CRX/ SCL	44	DB/ DO/ DB/ DB/ DB/ DI/ DI/ DB	GPIO, configurable as INT0 input Debug port Timer4 input/output Timer3 input/output after function switching UART2 RXD input in two-wire mode or TXD output/RXD input in single-wire mode after function switching LIN RXD input CAN RXD input I²C SCL, configured as open-drain output
P0.2/ HAL0/ LXIN	45	DB/ AI/ DI	GPIO, configurable as INT0 input Hall-IC0 logic level input 32768Hz crystal clock input

Pin	FU6367Q QFN56	IO Type	Description
P0.4/	46	DB/	GPIO, configurable as INT1 input
NSS		DB	SPI NSS
P0.5/	47	DB/	GPIO, configurable as INT0 input
SCLK/		DO/	SPI SCLK
TXD/		DO/	UART1 TXD output
CTXS/		DO/	CAN TXD output after function switching
SDAS		DB	I²C SDA after function switching, configured as open-drain output
P0.6/	48	DB/	GPIO, configurable as INT0 input
MOSI		DI/	SPI MOSI, master output or slave input
RXD/		DB/	UART1 RXD input in two-wire mode or TXD output/RXD input in single-wire mode
CRXS/		DI	CAN RXD input after function switching
SCLS		DB	I²C SCL after function switching, configured as open-drain output
P0.7/	49	DB/	GPIO, configurable as INT1 input
MISO/		DB/	SPI MISO, master input or slave output
TIM2S/		DB/	Timer2 input/output after function switching
QEPA		DI	QEP encode A input
P1.0/	50	DB/	GPIO, configurable as INT1 input and open-drain output
TIM2/		DB/	Timer2 input/output, configured as open-drain output
TXDS/		DO/	UART1 TXD output after function switching, configured as open-drain output
LTXS/		DO/	LIN TXD output after function switching, configured as open-drain output
QEPB		DI	QEP encode B input
P1.1/	51	DB/	GPIO, configurable as INT0/INT1 input and open-drain output
TIM3/		DB/	Timer3 input/output, configured as open-drain output
RXDS/		DB/	UART1 RXD input in two-wire mode or TXD output/RXD input in single-wire mode after function switching
LRXS/		DI/	LIN RXD input after function switching, configured as open-drain output
LIN		DB	LIN RXD/TXD, configured as open-drain output
VDRV	52	P	6N pre-driver power supply with a 1μF ~ 10μF external capacitor
COM	53	P	DRV ground
LU	54	DO	6N pre-driver low-side U-phase PWM output
LV	55	DO	6N pre-driver low-side V-phase PWM output
LW	56	DO	6N pre-driver low-side W-phase PWM output

2.2 FU6367Q QFN56 Pinout Diagram

Figure 2-1 FU6367Q QFN56 Pinout Diagram

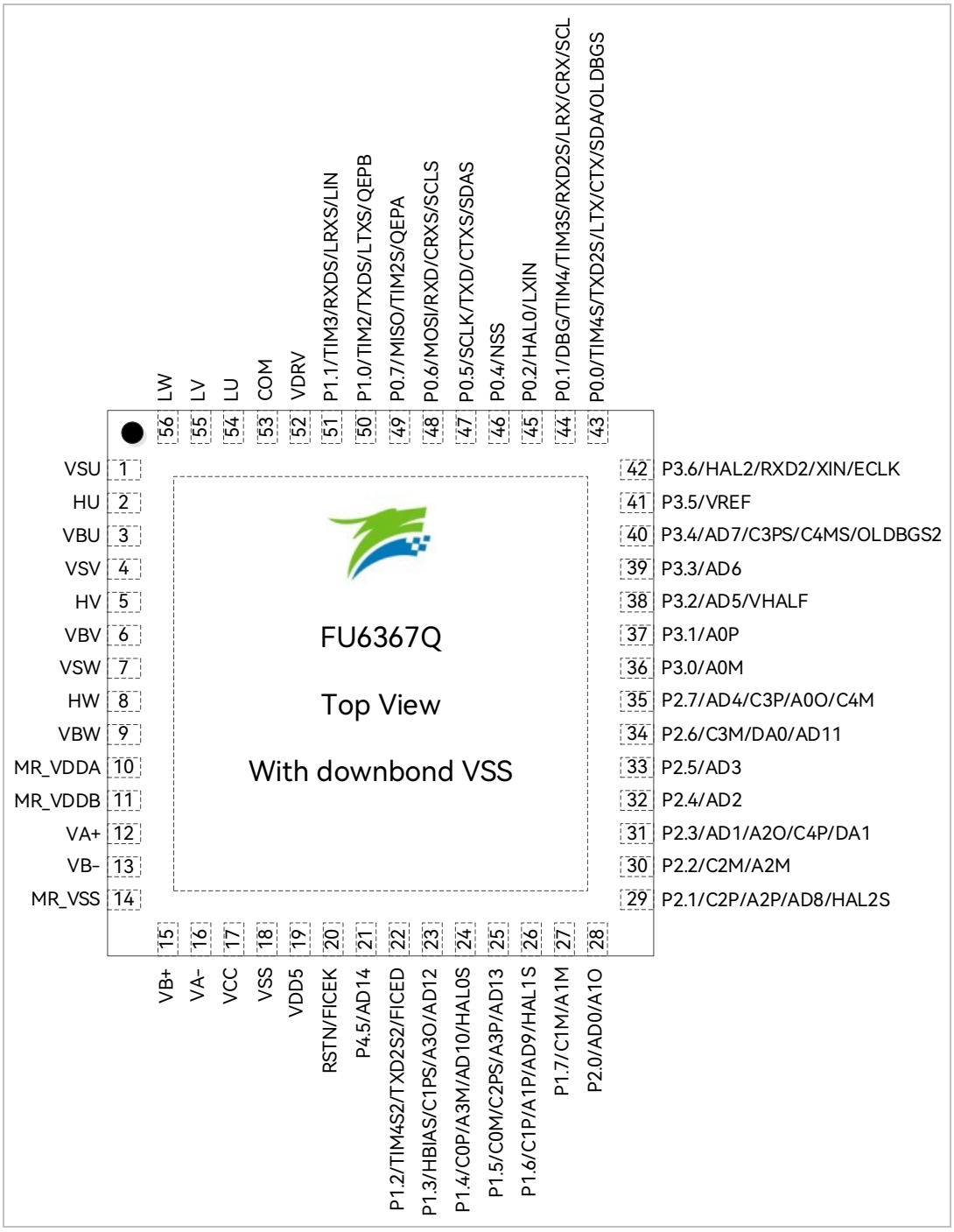
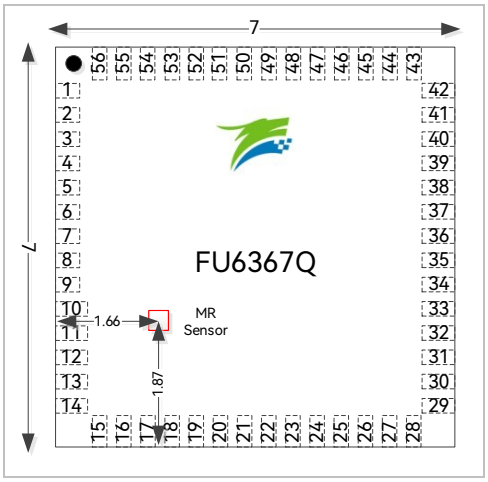


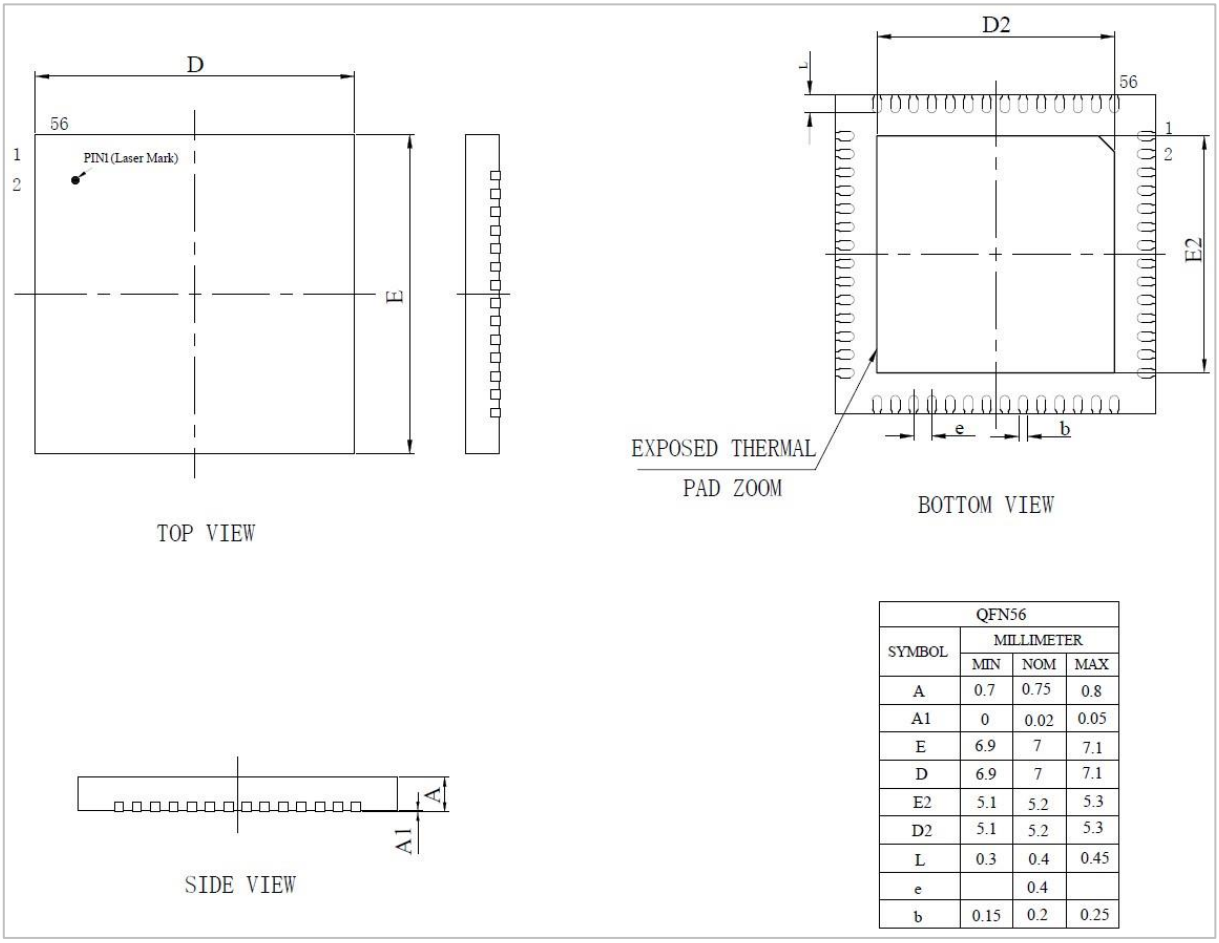
Figure 2-2 Location of the Sensor (unit: mm)



3 Package Information

3.1 FU6367Q QFN56_7X7

Figure 3-1 FU6367Q QFN56_7X7 Package Drawings and Dimensions



QFN56			
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.7	0.75	0.8
A1	0	0.02	0.05
E	6.9	7	7.1
D	6.9	7	7.1
E2	5.1	5.2	5.3
D2	5.1	5.2	5.3
L	0.3	0.4	0.45
e		0.4	
b	0.15	0.2	0.25

4 Ordering Information

Table 4-1 Model Selections

Model	Clock Frequency (MHz)	Flash (kByte)	XRAM (kByte)	Clock Circuit				Driver Interface		Drive Type		I²C/UART/SPI/LIN/CAN	DMA	GPIO	Timer	Analog Peripherals								Lead-free	Package
																ADC			DAC						
				Internal Fast Clock	External Fast Clock	Internal Slow Clock	External Slow Clock	6N Pre-driver	PWM	Square-wave	FOC					Number	Channel	Bits	Number	Bits	VREF	Operational Amplifier	Comparator		
FU6367Q	24	32	3.75	√	√	√	√	√	-	√	√	√	√	31	6	1	16	12	2	9\6	√	4	3	√	QFN56 (7x7mm)

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	℃
Operating Ambient Temperature T_A	VCC ≤ 12V, I_{vcc} ≤ 30mA	-	-	105	℃
Storage Temperature T_{stg}		-55	-	150	℃
VCC to VSS Spike Voltage	t < 500ms	-0.3	-	40	V
VCC to VSS Spike Voltage	t < 60s	-0.3	-	36	V
VCC to VSS Voltage		-0.3	-	20	V
VDRV to VSS Voltage		-0.3	-	20	V
High-side Floating Voltage $V_{BU,BV,BW}$		-0.3	-	205	V
High-side Floating Offset Voltage $V_{SU,SV,SW}$		$V_{BU,BV,BW}$ - 20	-	$V_{BU,BV,BW}$ + 0.3	V
High-side Output Voltage $V_{HU,HV,HW}$		$V_{SU,SV,SW}$ - 0.3	-	$V_{BU,BV,BW}$ + 0.3	V
Low-side Output Voltage $V_{LU,LV,LW}$		-0.3	-	VDRV + 0.3	V
VDD5 to VSS Voltage		-0.3	-	6.5	V
P10/P11		-0.3	-	VCC + 0.3	V
RSTN/GPIO to VSS Voltage	Exclude P1[1:0]	-0.3	-	VDD5 + 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℃ unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Power Supply VDDA/VDDB		1.0	5.0	10.0	V
Resistance (R)	Bridge Current = 1mA The two bridges are connected in parallel	888	1110	1332	Ω

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
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	135	140	165	mV
Zero Offset of Bridge	Magnetic Field =80G	-1	0	+1	mV/V
Temperature Coefficient of Bridge Resistance (TCR)	VDDA = 5V, T_A = -40℃ ~ +125℃	-	3.085	-	Ω/℃
Temperature Coefficient of Sensitivity (TCS)	VDDA = 5V, T_A = -40℃ ~ +125℃	-	-4.94	-	uV/V/℃
Temperature Coefficient of Bridge Offset (TCO)	T_A = -40℃ ~ +125℃	-	±7.74	-	uV/V
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 = -40℃ ~ 105℃ and VCC = 6.5V ~ 28V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VCC Operating Voltage[1][2]	High-voltage Single-power Supply Mode	6.5	-	28	V
I_{VCC} Operating Current[3]		-	20	-	mA
I_{VCC} Standby Current[3]		-	6	-	mA
I_{VCC} Sleep-mode Current	VCC = 12V, T_A = 25℃	-	30	50	μA
Operating Ambient Temperature T_A	VCC ≤ 15V & I_{VCC} ≤ 30mA	-	-	105[4]	℃

Note:

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

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

[3] Characteristics may vary with different configurations.

[4] The chip can works at the maximum T_A only when it is ensured that Operating Junction Temperature T_J does not exceed the maximum specified in any case.

5.4 GPIO Electrical Characteristics

Table 5-4 GPIO Electrical Characteristics

(T_A = 25℃ and VCC = 6.5V ~ 28V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
Turn-on Rise Time	50pF load, from 10% to 90%, T_A = 25℃	-	15	-	ns
Turn-off Fall Time	50pF load, from 90% to 10%, T_A = 25℃	-	13	-	ns
V_{OH} High-level Output Voltage	I_{OH} = 4mA, T_A = -40℃ ~ 105℃	VDD5 - 0.7	-	-	V
V_{OL} Low-level Output Voltage	I_{OL} = 4mA, T_A = -40℃ ~ 105℃	-	-	VSS + 0.7	V
V_{IH} High-level Input Voltage[1]		0.7*VDD5	-	-	V
V_{IL} Low-level Input Voltage		-	-	0.2*VDD5	V
P10/P11 High-level Input Voltage	High-voltage Mode	0.7*VCC	-	-	V
P10/P11 Low-level Input Voltage	High-voltage Mode	-	-	0.2*VCC	V
Pull-up Resistor[2]		-	33	-	kΩ
Pull-up Resistor[3]		-	5.6	-	kΩ
Pull-down Resistor[4]		-	30	-	kΩ
Pull-down Resistor[5]		-	50	-	kΩ

Note:

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

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

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

[4] P0[1]

[5] P1[1]

5.5 6N Pre-driver IO Electrical Characteristics

Table 5-5 6N Pre-driver IO Electrical Characteristics

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

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
High-level Peak Output Current		-	0.9	-	A
Low-level Peak Output Current		-	1.1	-	A
VDRV Supply Voltage		-	11	-	V

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
High-side Floating Voltage $V_{BU,BV,BW}$		-	-	180	V
High-side Floating Offset Voltage $V_{SU,SV,SW}$		$V_{BU,BV,BW}$ - 12	-	$V_{BU,BV,BW}$ - 6.5	V
VDRV UVLO Threshold Voltage		4.2	4.6	5	V
VDRV UVLO Release Voltage		3.9	4.3	4.7	V
VDRV UVLO Hysteresis Voltage		0.2	0.3	-	V
Turn-on Rise Time	1nF load, from 10% to 90%	-	12	25	ns
Turn-off Fall Time	1nF load, from 90% to 10%	-	12	25	ns
Deadtime	DT	-	200	-	ns

5.6 ADC Electrical Characteristics

Table 5-6 ADC Electrical Characteristics

(T_A = 25℃ and VCC = 6.5V ~ 28V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
INL (Integral Nonlinearity)	12-bit	-	2	-	LSB
DNL (Differential Nonlinearity)	12-bit	-	1.5	-	LSB
OFFSET (Offset Error)	12-bit	-	6	-	LSB
SNR (Signal-to-noise Ratio)	f_{IN} = 350kHz	-	70.8	-	dB
ENOB (Effective Number of Bits)	f_{IN} = 350kHz	-	10.5	-	bit
SFDR (Spurious-free Dynamic Range)	f_{IN} = 350kHz	-	68.2	-	dB
THD (Total Harmonic Distortion)	f_{IN} = 350kHz	-	67	-	dB
R_{IN} Input Resistance		-	800	-	Ω
C_{IN} Input Capacitance		-	30	-	pF
Conversion Time		-	13	-	ADCLK[1]
Sampling Time		3	-	63	ADCLK2[2]

Note:

[1] ADCLK = 24MHz

[2] ADCLK2 = 12MHz

5.7 VREF and VHALF Electrical Characteristics

Table 5-7 VREF and VHALF Electrical Characteristics

(T_A = -40℃ ~ 105℃ and VCC = 6.5V ~ 28V)

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

5.8 Operational Amplifier Electrical Characteristics

Table 5-8 Operational Amplifier Electrical Characteristics

(T_A = 25℃ and VCC = 6.5V ~ 28V unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{ICMR} Common-mode Input Voltage Range		0	-	VDD5 - 1.5	V
V_{OS} Operational Amplifier Offset Voltage	T_A = 25℃	-	5	10	mV
A_{OL} Open-loop Gain	R_L = 100kΩ	-	80	-	dB
Unity-gain Bandwidth (UGBW)	C_L = 40pF	6	10	-	MHz
Slew Rate (SR)	C_L = 40pF	10	15	-	V/μs
Operational Amplifier Gain[1][2]	AMP_CR1[AMP0_GAIN] = 001	1.88	2	2.12	-
	AMP_CR1[AMP0_GAIN] = 010	3.76	4	4.24	-
	AMP_CR1[AMP0_GAIN] = 011	7.5	8	8.5	-
	AMP_CR1[AMP0_GAIN] = 100	15	16	17	-

Note:

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

[2] AMP0 is used as an example. See AMP_CR1 (0x4034) for configurations of other operational amplifiers.

5.9 BEMF Electrical Characteristics

Table 5-9 BEMF Electrical Characteristics

(T_A = 25℃, VCC = 6.5V ~ 28V and VCC_MODE = 0 unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
BEMF Built-in Resistor	T_A = -40℃ ~ 105℃	5.4	6.8	8.2	kΩ
Relative Accuracy between BEMF Built-in Resistors		-	1	-	%

5.10 OSC Electrical Characteristics

Table 5-10 OSC Electrical Characteristics

(T_A = -40℃ ~ 105℃, VCC = 6.5V ~ 28V and VCC_MODE = 0)

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, system clock rate of the chip is 24MHz and T = 1/SYSCLK.

5.11 Reset Electrical Characteristics

Table 5-11 Reset Electrical Characteristics

(T_A = 25℃, VCC = 6.5V ~ 28V and VCC_MODE = 0 unless otherwise specified)

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

5.12 LDO Electrical Characteristics

Table 5-12 LDO Electrical Characteristics

(T_A = 25℃, VCC = 6.5V ~ 28V and VCC_MODE = 0 unless otherwise specified)

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD5 Voltage	VCC = 6.5V ~ 28V, VCC_MODE = 0, T_A = -40℃ ~ 105℃	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 θ_{JA}[1]	JEDEC standard, 2S2P PCB	33	℃/W
	JEDEC standard, 1S0P PCB	55	℃/W
Junction-to-case Temperature Thermal Resistance θ_{JC}[1]	JEDEC standard, 2S2P PCB	9.2	℃/W

Note:

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

6 Reset Control

6.1 Reset Source (RST_SR)

The chip includes a reset circuitry with 7 reset sources:

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

The reset flag is queryable and recorded in register RST_SR. Following the last reset, the affected reset flag is set to “1” and all other reset flags are cleared to “0”. In order to clear a reset flag, you can set RST_SR[RSTCLR] flag to “1” so that RST_SR[7:3]&RST_SR[0] are cleared. After reset, MCU starts the program from address 0x0000.

6.2 Reset Enable

See the corresponding control registers.

6.3 External Pin Reset and Power-on Reset

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

The chip resets when the chip powers on and the voltage settles above the reset voltage threshold.

6.4 Low Voltage Detection Reset

The chip’s internal circuitry monitors VCC voltage. When VCC voltage drops to a level below the reset

voltage threshold, the internal monitor circuitry sends the reset signal to reset the chip.

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

6.5 Watchdog Timer Reset

After the watchdog timer (WDT) is enabled, the software periodically clears WDT to avoid timeout. If system error occurs, the timer generates an output pulse to reset the chip.

6.6 Flash Error Detector Reset

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

6.7 Debug Reset

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

6.8 Soft Reset

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

6.9 Reset Registers

6.9.1 RST_SR (0xC9)

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

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

7 Interrupt

7.1 Interrupt Introduction

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

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

7.2 Interrupt Enable

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

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

7.3 External Interrupts

The external interrupt has 2 interrupt sources: INT0 and INT1. They both can be configured as interrupt on rising edge, interrupt on falling edge or interrupt on edge changes (rise or fall).

The digital input signals from P0.0 ~ P0.2, P0.5 ~ P0.6, P1.1 and output signals from CMP4 can be used to trigger an INT0. The interrupt source is selected through LVSR[EXT0CFG] bit. These trigger sources share one interrupt entry point, one interrupt flag bit TCON[IF0] and one interrupt enable bit IE[EX0]. TCON[IT0] bit selects the interrupt edge. IP0[PX0] bit configures the priority level.

The digital input signals from P0.4, P0.7, P1.0 ~ P1.7, P3.6 and P4.5 can be used to trigger an INT1. P1_IF and P4_IF are interrupt flag bits, and P1_IE and P4_IE are interrupt enable bits. Each trigger source has a corresponding interrupt flag bit and an interrupt enable bit. INT1 can select multiple trigger sources that are recognized by P1_IF and P4_IF in the interrupt subroutine. These 16 interrupt sources share one interrupt entry and one interrupt enable bit IE[EX1]. To enable INT1, first set IE[EX1] to “1” and then configure the corresponding enable bit. The interrupt edge is configured by TCON[IT1] bit, and the priority level by IP0[PX1] bit. See 7.5.7 P1_IE (0xD1) ~ 7.5.10 P4_IF (0xD4) for INT1 interrupt flags and enable registers.

7.4 Interrupt Summary

Table 7-1 Interrupt Summary

Interrupt Source	Priority Order	Interrupt Vector	Interrupt Flag	Cleared by SW?	Enable Bit	Priority Control
Reset	Highest	0x0000	None	N	Always enabled	Highest
LVW Interrupt MCD Interrupt	0	0x0003	LVSR[0] TCON[6]	Y	CCFG1[6] IE[5]	IP0[1:0]
INT0	1	0x000B	TCON[2]	Y	IE[0]	IP0[3:2]
INT1	2	0x0013	P1_IF[7:0] P4_IF[7:0]	Y	IE[2]	IP0[5:4]
FG Interrupt DRV CM Interrupt	3	0x001B	DRV_SR[5:4]	Y	DRV_SR[3] DRV_SR[2:0]	IP0[7:6]
Timer2 Interrupt	4	0x0023	TIM2_CR1[7:5]	Y	TIM2_CR1[4:3] TIM2_CR0[3]	IP1[1:0]
Timer1 Interrupt	5	0x002B	TIM1_SR[5:0]	Y	TIM1_IER[5:0]	IP1[3:2]
ADC Interrupt	6	0x0033	ADC_CR[0]	Y	ADC_CR[1]	IP1[5:4]

Interrupt Source	Priority Order	Interrupt Vector	Interrupt Flag	Cleared by SW?	Enable Bit	Priority Control
CMP0/1/2 Interrupt Hall Interrupt	7	0x003B	CMP_SR[6:4] HALL_CR[7]	Y	CMP_CR0[5:0] HALL_CR[6]	IP1[7:6]
RTC/CAN Interrupt	8	0x0043	RTC_STA[6] CAN_IFR[6:0]	Y	IE[6] CAN_IER[7:0] CAN_CR2[2:0]	IP2[1:0]
Timer3 Interrupt	9	0x004B	TIM3_CR1[7:5]	Y	TIM3_CR1[4:3] TIM3_CR0[3]	IP2[3:2]
Systick Interrupt	10	0x0053	DRV_SR[7]	Y	DRV_SR[6]	IP2[5:4]
Timer4 Interrupt	11	0x005B	TIM4_CR1[7:5]	Y	TIM4_CR1[4:3] TIM4_CR0[3]	IP2[7:6]
CMP3 Interrupt	12	0x0063	CMP_SR[7]	Y	CMP_CR0[7:6]	IP3[1:0]
I^2C Interrupt UART1 Interrupt	13	0x006B	I2C_SR[0] UT_CR[1:0]	Y	I2C_CR[0] IE[4]	IP3[3:2]
SPI Interrupt UART2 Interrupt LIN Interrupt	14	0x0073	SPI_CR1[7:4] UT2_CR[1:0] LIN_SR[2:0] LIN_CSR[4:3]	Y	IE[3] UT2_BAUDH[5] LIN_CR[3]	IP3[5:4]
DMA Interrupt	15	0x007B	DMA0_CR0[0] DMA1_CR0[0]	Y	DMA0_CR0[2] DMA1_CRO[2:1]=10	IP3[7:6]

Note:

- UT_CR[RI], UT_CR[TI], UT2_CR[UT2RI], UT2_CR[UT2TI], DMA0_CR0[DMAIF] and DMA1_CR0[DMAIF] flags can be cleared to “0” or set to “1” by software. When these flags are set to “1”, an interrupt request is generated. Other interrupt flags can only be cleared to “0” by software, and setting them to “1” has no meaning.
- For a register containing multiple interrupt flags, you can write a “1” to the active interrupt flags in order to prevent clearing a interrupt flag to “0”. Take DRV_SR as an example. When DRV_SR[SYSTIF] is cleared to “0” by software, you can use the statement DRV_SR = (DRV_SR&0x7F) | 0x30 to prevent the software from clearing DRV_SR[FGIF] and DRV_SR[DCIF] to “0”.

7.5 Interrupt Registers

7.5.1 IE (0xA8)

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

7.5.2 IP0 (0x8A)

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

Bit	Name	Description
[7:6]	PDRV	FG/DRV Compare Match Interrupt Priority Setting
[5:4]	PX1	External Interrupt 1 (INT1) Priority Setting

[3:2]	PX0	External Interrupt 0 (INT0) Priority Setting
[1:0]	PLVW_MCD	LVW/MCD Match Interrupt Priority Setting

Note:

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

7.5.3 IP1 (0x8B)

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

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

Note:

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

7.5.4 IP2 (0x8C)

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

Bit	Name	Description
[7:6]	PTIM4	Timer4 Interrupt Priority Setting
[5:4]	PSYSTICK	Systick Interrupt Priority Setting
[3:2]	PTIM3	Timer3 Interrupt Priority Setting
[1:0]	PRTC_CAN	RTC/CAN CM Interrupt Priority Setting

Note:

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

7.5.5 IP3 (0x8D)

Bit	7	6	5	4	3	2	1	0
Name	PDMA		PSPI_UART2_LIN		PI2C_UART1		PCMP3	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	PDMA	DMA Interrupt Priority Setting
[5:4]	PSPI_UART2_LIN	SPI/UART2/LIN Interrupt Priority Setting
[3:2]	PI2C_UART1	I²C/UART1 Interrupt Priority Setting
[1:0]	PCMP3	CMP3 Interrupt Priority Setting

Note:

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

7.5.6 TCON (0x88)

Bit	7	6	5	4	3	2	1	0
Name	RSV	MCDIF	RSV	IT1		IF0	IT0	
Type	-	R/W0	-	R/W	R/W	R/W0	R/W	R/W
Reset	-	0	-	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	MCDIF	Missing Clock Detector Interrupt Flag When a missing clock event is detected, the flag is set to “1” by hardware. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[5]	RSV	Reserved
[4:3]	IT1	External Interrupt 1 (INT1) Edge Selection 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on edge changes (rise or fall)
[2]	IF0	External Interrupt 0 (INT0) Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write:

		0: This bit is cleared to “0” 1: No effect
[1:0]	IT0	External Interrupt 0 (INT0) Edge Selection 00: Interrupt on rising edge 01: Interrupt on falling edge 1X: Interrupt on edge changes (rise or fall)

7.5.7 P1_IE (0xD1)

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

Bit	Name	Description
[7]	P17_IE	P1.7 INT1 Enable 0: Disable 1: Enable
[6]	P16_IE	P1.6 INT1 Enable 0: Disable 1: Enable
[5]	P15_IE	P1.5 INT1 Enable 0: Disable 1: Enable
[4]	P14_IE	P1.4 INT1 Enable 0: Disable 1: Enable
[3]	P13_IE	P1.3 INT1 Enable 0: Disable 1: Enable
[2]	P12_IE	P1.2 INT1 Enable 0: Disable 1: Enable
[1]	P11_IE	P1.1 INT1 Enable 0: Disable 1: Enable
[0]	P10_IE	P1.0 INT1 Enable 0: Disable 1: Enable

7.5.8 P1_IF (0xD2)

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

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

7.5.9 P4_IE (0xD3)

Bit	7	6	5	4	3	2	1	0
Name	P04_IE	RSV	P45_IE	P44_IE	P36_IE	P42_IE	P41_IE	P07_IE
Type	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	0

Bit	Name	Description
[7]	P04_IE	P0.4 INT1 Enable 0: Disable 1: Enable

[6]	RSV	Reserved
[5]	P45_IE	P4.5 INT1 Enable 0: Disable 1: Enable
[4]	P44_IE	P4.4 INT1 Enable 0: Disable 1: Enable
[3]	P36_IE	P3.6 INT1 Enable 0: Disable 1: Enable
[2]	P42_IE	P4.2 INT1 Enable 0: Disable 1: Enable
[1]	P41_IE	P4.1 INT1 Enable 0: Disable 1: Enable
[0]	P07_IE	P0.7 INT1 Enable 0: Disable 1: Enable

7.5.10 P4_IF (0xD4)

Bit	7	6	5	4	3	2	1	0
Name	P04_IF	RSV	P45_IF	P44_IF	P36_IF	P42_IF	P41_IF	P07_IF
Type	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	0

Bit	Name	Description
[7]	P04_IF	P0.4 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[6]	RSV	Reserved
[5]	P45_IF	P4.5 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[4]	P44_IF	P4.4 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[3]	P36_IF	P3.6 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[2]	P42_IF	P4.2 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending

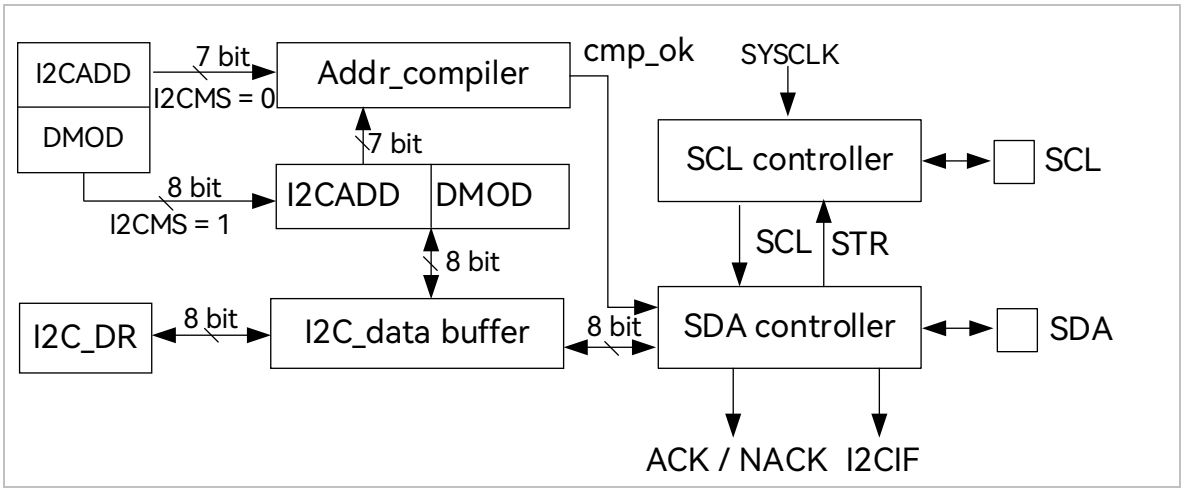
[1]	P41_IF	P4.1 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending
[0]	P07_IF	P0.7 INT1 Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending

8 I²C

8.1 I²C Introduction

The I²C module provides an industry standard two-wire serial interface and is a simple bi-directional synchronous serial bus for communication between MCU and external I²C devices, as shown in Figure 8-1. The bus consists of two serial lines: SDA (serial data line) and SCL (serial clock line). P0.0/P0.5 serves as SDA port and P0.1/P0.6 as SCL port. After I²C is enabled, P0.0/P0.5 and P0.1/P0.6 automatically shifts into open-drain outputs, and P0.5 and P0.6 are translated into SDA and SCL ports after I²C function switching.

Figure 8-1 Block Diagram of I²C Module



Features:

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

Both SDA and SCL lines are high level when the bus is idle, which is the only basis for detecting whether the bus is idle or not. Only one master device and at least one slave device are active on the bus during the transmission. When the bus is occupied, other devices must wait for the bus idle to start an I²C communication. The master starts the bus to transfer data. Clock signal is sent to all devices via SCL and

the slave address and read/write mode are sent via SDA. When a device on the bus matches the address, it acts as a slave. The relationships between masters and slaves or data transfer direction on the bus are not constant. The process for the master to send data to the slave is shown in Figure 8-2 Master Transmits Data to Slave. The master first addresses the slave device and waits for the slave response. And then, it sends data to the slave. Finally, the master terminates the data transmission. The process for the master to receive data from the slave is shown in Figure 8-3. The master first addresses the slave and waits for the slave response. And then, it receives the data from the slave. Finally, the master terminates data transmission. In this case, the master generates timing clock and stops data transmission.

Figure 8-2 Master Transmits Data to Slave

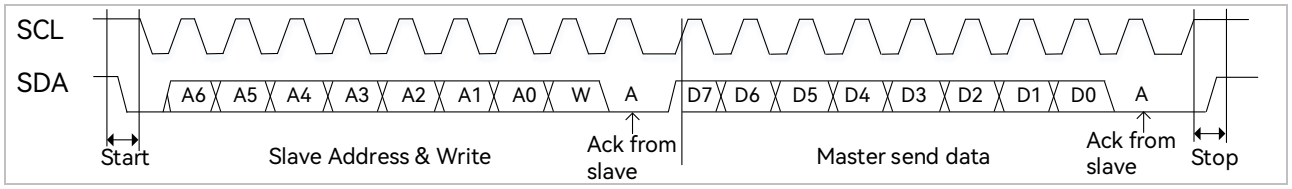
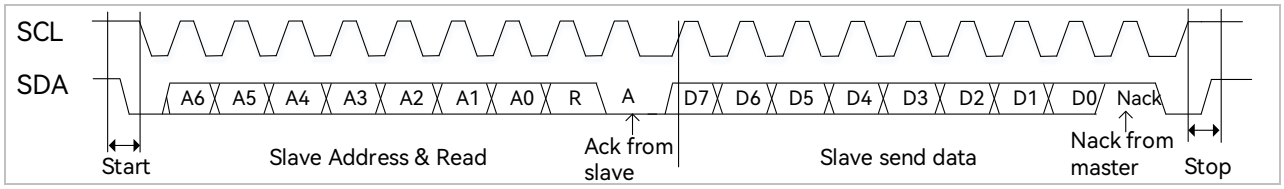


Figure 8-3 Master Receives Data from Slave



8.2 I²C Operations

8.2.1 Master Mode

1. Set I2C_CR[I2CMS] to “1” to select master mode;
2. Configure I2C_CR [I2CSPD] to set the clock rate of SCL;
3. Configure I2C_ID[I2CADD] to set the slave address;
4. Configure I2C_SR[DMOD] to set the read/write direction;
5. Set I2C_CR[I2CEN] to “1” to enable I²C;
6. Set I2C_SR[I2CSTA] to “1” to 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 cleared 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 cleared 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”. The stop signal is sent after I2C_SR[STR] is reset.

8.2.2 Slave Mode

1. Set I2C_CR[I2CMS] to “0” to select slave mode;
2. Configure I2C_ID[I2CADD] to set the slave address or set I2C_ID[GC] to “1” to enable general call mode;
3. Set I2C_CR[I2CEN] to “1” to enable I^2C;
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] and I2C_SR[STR] is cleared 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.

8.2.3 I²C Interrupt Sources

The interrupt sources of I²C include:

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

8.3 I²C Registers

8.3.1 I2C_CR (0x4028)

Bit	7	6	5	4	3	2	1	0
Name	I2CEN	I2CMS	RSV	I2CDMANAKINT	I2CDMAAUTO	I2CSPD		I2CIE
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]	I2CEN	I²C Enable The associated GPIOs are enabled to switch to I²C mode, with open-drain output. The pull-up setting decides whether to pull I²C HIGH. 0: Disable 1: Enable
[6]	I2CMS	Master/Slave Mode Selection 0: Slave 1: Master
[5]	RSV	Reserved
[4]	I2CDMANAKINT	DMA Transmission Interrupt Enable When NAK Reply Is Ignored 0: Disable 1: Enable
[3]	I2CDMAAUTO	Automatically Transfer the First Byte of Data during DMA Transmission 0: Disable 1: Enable
[2:1]	I2CSPD	I²C Transfer Rate Setting, valid only in Master Mode 00: 100kHz 01: 400kHz 10: 1MHz 11: Reserved
[0]	I2CIE	I²C Interrupt Enable 0: Disable 1: Enable

8.3.2 I2C_ID (0x4029)

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

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

8.3.3 I2C_DR (0x402A)

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

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

8.3.4 I2C_SR (0x402B)

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

Bit	Name	Description
[7]	I2CBSY	I^2C Busy Flag When I2C_CR[I2CEN] = 0, I2C_SR[I2CBSY] is cleared to “0” by hardware. Master Mode: After START is transmitted, I2C_SR[I2CBSY] is set to “1” by hardware; after STOP is transmitted, I2C_SR[I2CBSY] is cleared to “0” by hardware. Slave Mode: After START is received and address matches, I2C_SR[I2CBSY] is set to “1” by hardware; after STOP is received, I2C_SR[I2CBSY] is cleared to “0” by hardware.

<table>
<tr><td>[6]</td><td>DMOD</td><td>I²C R/W Flag
0: WRITE (master sends the data, and slave receives the data)
1: READ (master receives the data, and slave sends the data)

Note:
Read only in Slave Mode</td></tr>
<tr><td>[5]</td><td>RSV</td><td>Reserved</td></tr>
<tr><td>[4]</td><td>I2CSTA</td><td>Master Mode:
When this bit is configured with “1”, START and address bytes are send after both SCL and SDA are HIGH confirmed by the hardware. This bit is cleared to “0” by hardware automatically when the transmission is completed, and I2C_SR[I2CSTA] writing is forbidden during data transmission. After the data is sent or received, I2C_SR[I2CSTA] is set to “1” to transmit RESTART.
0: Not START and address bytes
1: Transmit START or RESTART and address bytes
Slave Mode:
This bit is set to “1” after the hardware receives START and address matches, and cleared to “0” by software.
Table 8-1 Mapping between I²C Data Type and I2C_SR[I2CSTA] & I2C_SR[I2CSTP] in Slave Mode
<table>
<tr><th>I2CSTA</th><th>I2CSTP</th><th>I²C Data Type</th></tr>
<tr><td>0</td><td>0</td><td>Data byte</td></tr>
<tr><td>0</td><td>1</td><td>STOP</td></tr>
<tr><td>1</td><td>0</td><td>START + address bytes</td></tr>
<tr><td>1</td><td>1</td><td>STOP received first, then START + address bytes</td></tr>
</table>
Note:
When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTA] is automatically cleared to “0”.</td></tr>
<tr><td>[3]</td><td>I2CSTP</td><td>Master Mode:
This bit cannot be written to “1” by software unless I2C_SR[I2CBSY] = 1; STOP is transmitted after I2C_SR[STR] is cleared to release SCL. After the transmission, this bit is cleared to “0” automatically by hardware. If 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.</td></tr>
</table>

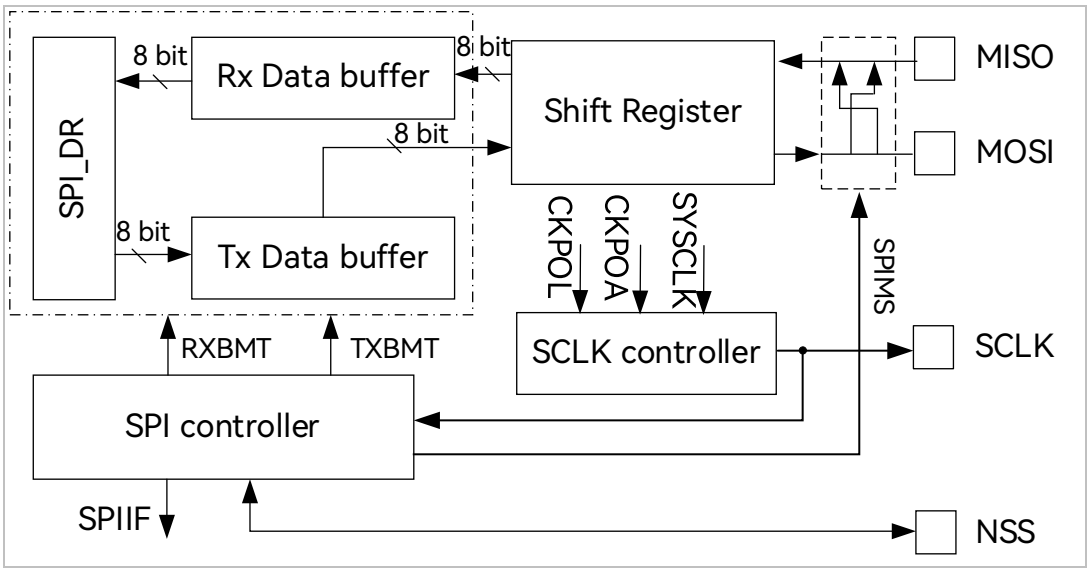
		See Table 8-1 for status flags. Note: When I2C_CR[I2CEN] = 0, I2C_SR[I2CSTP] is automatically cleared to “0” by hardware.
[2]	STR	I²C Bus Pending Flag Master Mode: After START and address or DATA byte are transmitted, I2C_SR[STR] are set to “1” by hardware and SCL is pulled LOW. SCL is released after I2C_SR[STR] is cleared by software. When I2C_SR[I2CSTA] and I2C_SR[I2CSTP] are both “1”, I2C_SR[STR] is set to “1” only after hardware sends STOP and START & address bytes. Slave Mode: When DATA byte is received or START is received and address matches, I2C_SR[STR] is set to “1” by hardware and SCL is pulled LOW. SCL is released after I2C_SR[STR] is cleared by software. Note: This bit is set to “1” by hardware and cleared to “0” by software. When I2C_CR[I2CEN] = 0, I2C_SR[STR] is automatically cleared to “0”.
[1]	NACK	This bit refers to the feedback from a receiver to a sender after a byte is transferred via I²C. It is automatically cleared to “0” when I2C_SR[I2CEN] = 0. 0: ACK, indicating that the receiver can continue to receive data. 1: NACK, indicating that the receiver attempts to stop data transmission. When the device is in READ mode, I2C_SR[NACK] is configured to transmit ACK/NACK after the 8th bit of data is received. 0: Bit9 transmits ACK 1: Bit9 transmits NACK When the device is in WRITE mode, I2C_SR[NACK] is read to receive ACK/NACK after the 8th bit of data is transmitted. 0: Bit9 receives ACK 1: Bit9 receives NACK
[0]	I2CIF	I²C Interrupt Flag 0: No Interrupt Pending 1: Interrupt Pending When I2C_SR[STR] = 1, an interrupt is generated in both Master and Slave modes. When I2C_SR[I2CSTP] = 1, an interrupt is generated only in Slave mode.

9 SPI

9.1 SPI Introduction

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

Figure 9-1 Block Diagram of SPI Module



9.2 SPI Operations

9.2.1 Signal Descriptions

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

9.2.1.1 Master Out, Slave In (MOSI)

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

9.2.1.2 Master In, Slave Out (MISO)

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

9.2.1.3 Serial Clock (SCLK)

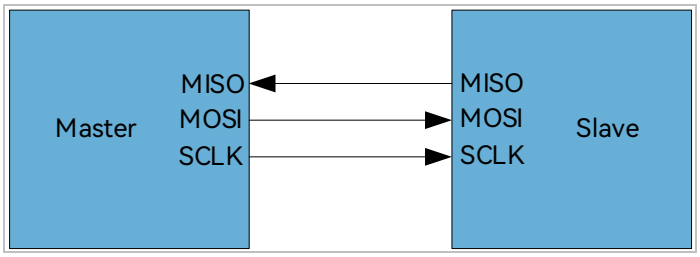
The serial clock (SCLK) signal is an output from the master device and an input to slave devices. It is used to synchronize serial data transmission between the master and slave. SCLK signal is generated by SPI operating as a master. In 4-wire slave mode, SCLK is ignored when the slave is not selected (NSS = 1).

9.2.1.4 Slave Select (NSS)

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

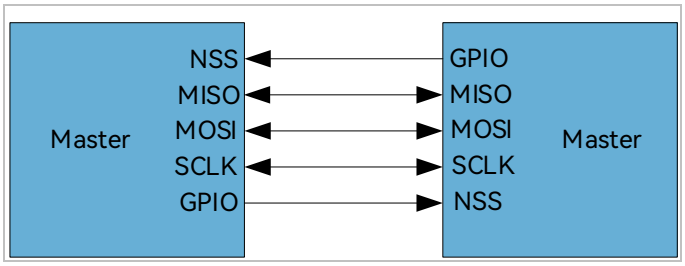
When SPI_CR1[NSSMOD] = 00, SPI operates in 3-wire Mode. NSS port is not necessary in this mode and there is only one master and one slave on the SPI bus. The connection diagram is shown in Figure 9-2.

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



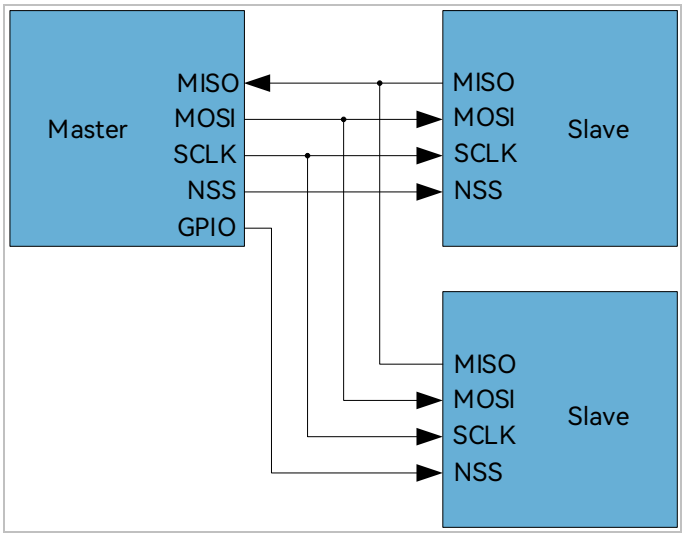
When SPI_CR1[NSSMOD] = 01, SPI operates in 4-wire Slave/Multi-Master Mode. In this mode, NSS pins on the SPI bus are configured as inputs, waiting to be addressed by the master. When SPI_CR0[SPIMS] = 0, SPI operates in 4-wire Slave Mode. If NSS is set to “0”, the slave is selected; while NSS is set to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

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



When SPI_CR1[NSSMOD] = 1X, SPI operates in 4-wire Single Master Mode. In this mode, NSS pin of the master on the bus is configured as an output, and NSS pin of the slave devices are configured as inputs. SPI_CR1[NSSMOD0] setting decides the output level of NSS pin serving as signal to select a slave. Other slaves can be selected using GPIO pins. The connection diagram is shown in Figure 9-4.

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



9.2.2 SPI Master Mode

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

9.2.2.1 Master Mode Configurations

1. Configure SPI_CR1[NSSMOD] to set the SPI operating mode;
2. Configure SPI_CR0[CPOL] to set the clock polarity;
3. Configure SPI_CR0[CPHA] to set the clock phase;
4. Set SPI_CR0[SPIMS] to “1” to select master mode;
5. Configure SPI_CLK to set the SCLK rate;
6. Set SPI_CR1[SPIEN] to “1” to enable SPI;
7. Write the data to SPI_DR. SPI transmits data for each write;
8. After SPI_CR1[SPIIF] is set to “1”, SPI_DR is read to receive the data.

9.2.3 SPI Slave Mode

When SPI_CR0[SPIMS] = 0, SPI operates in slave mode. In this mode, SCLK signal is sent by the master SPI. The data is shifted in from MOSI pin and shifted out from MISO pin. If no SCLK signal is input, shift register of the slave is in the stop state. If SCLK signal is input, the shift register of slave starts to receive and transmit

data through MOSI and MISO pins. The slave device cannot initiate transfers. The data sent to the master device is pre-loaded into the shift register by writing to SPI_DR. If the shift register is empty, the data in the transit buffer is transferred into the shift register. After the transmission is completed, SPI_CR1[SPIIF] and SPI_CR1[TXBMT] are set to “1”. The received data that is transferred into receive butter and receive buffer empty flag bit SPI_CR0[RXBMT] is cleared, indicating the new data has not been read. If SPI_CR0[RXBMT] is “0” and there is new data ready to be sent to the receive buffer, SPI_CR1[RXOVRN] is set to “1” and the data in the receive buffer remains unaffected. When data is written to SPI_DR, SPI_CR1[TXBMT] is cleared. If data is written in this case, the write conflict flag bit SPI_CR1[WCOL] is set to “1” and the data in the transmit buffer keeps unchanged.

9.2.3.1 Slave Mode Configurations

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

9.2.4 SPI Interrupt Sources

The interrupt sources of SPI include:

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

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

9.2.5 Serial Clock Timing

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

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

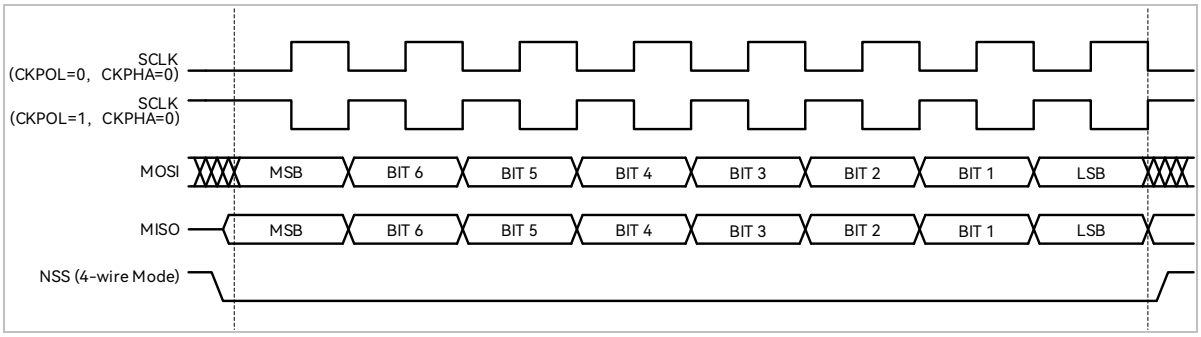
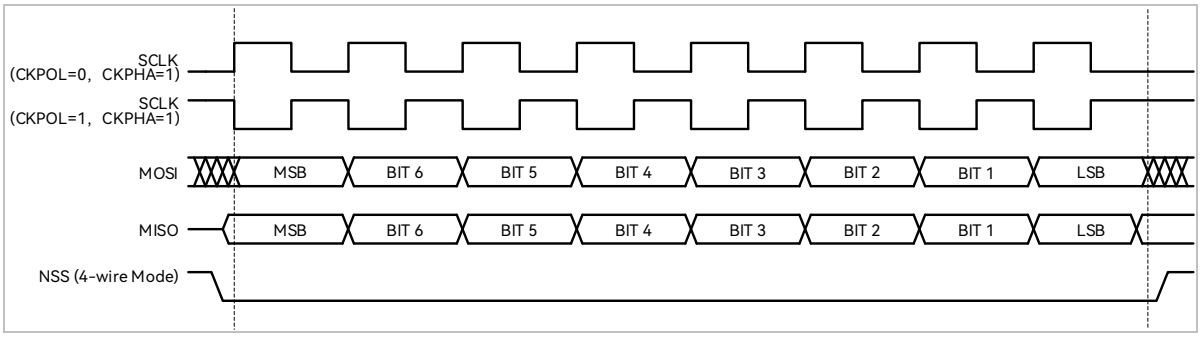


Figure 9-6 SDA/SCL Line Timing Diagram (SPI_CR0[CPHA] = 1)



9.3 SPI Registers

9.3.1 SPI_CR0 (0x4030)

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

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

Note:

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

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

9.3.2 SPI_CR1 (0x4031)

Bit	7	6	5	4	3	2	1	0
Name	SPIIF	WCOL	MODF	RXOVRN	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. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	WCOL	Write Conflict Interrupt Flag When SPI_CR1[TXBMT] is “0”, a write to SPI_DR sets this bit to “1”. This bit can be cleared to “0” by software only. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[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. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

[4]	RXOVRN	Receive Overflow Interrupt Flag (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 by software only. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0”. 1: No effect
[3:2]	NSSMOD	SPI Mode Selection 00: 3-wire Slave or 3-wire Master Mode. NSS signal is not routed to a port pin. 01: 4-wire Slave or Multi-Master Mode (Default). NSS pin is configured as an input. 1X: 4-wire Single-Master Mode. NSS pin is configured as output and outputs SPI_CR1[2] value.
[1]	TXBMT	Transmit Buffer Empty Flag This bit is cleared to “0” when new data is written to the Transit Buffer. It is set to “1” when the data in the Transit Buffer is transferred to the SPI shift register, indicating that it is safe to write a new byte to the transmit buffer. 0: A new byte is written to the transmit buffer. 1: Data in the transmit buffer has been transferred to the shift register.
[0]	SPIEN	SPI Enable 0: Disable 1: Enable

9.3.3 SPI_CLK (0x4032)

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

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

9.3.4 SPI_DR (0x4033)

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

Bit	Name	Description
[7:0]	SPI_DR	SPI Data Register SPI_DR Register is used to transmit and receive SPI data. Read: Receive the data of Receive Buffer Write: Write the data into Transit Buffer and initiate a transfer

10 UART

10.1 UART Introduction

UART is a full-duplex or half-duplex serial data exchange interface as shown in Figure 10-1. The baud rate is configurable and supports DMA transmission. Figure 10-2 depicts the UART timing.

Figure 10-1 Block Diagram of UART Module

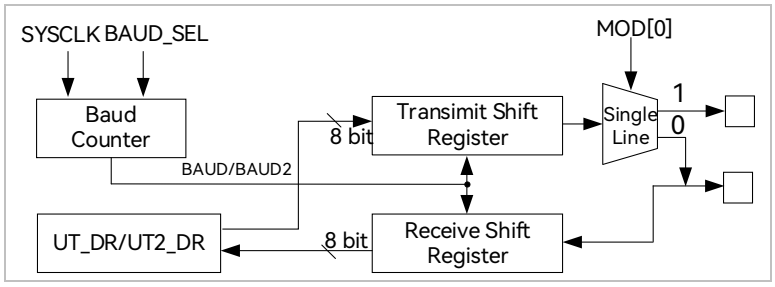
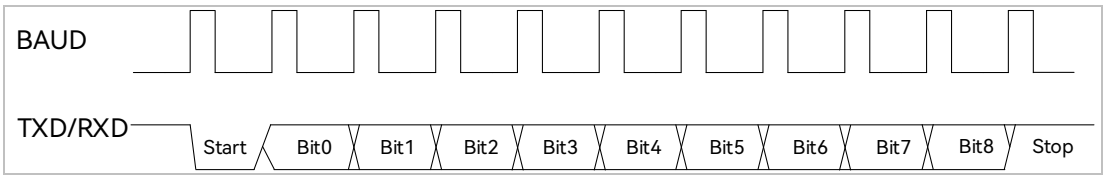


Figure 10-2 UART Timing Diagram



10.2 UART Operations

The corresponding registers shall be enabled before using UART feature. See 22.3.15 PH_SEL (0x404C) (bits [6] and [5]) ~ 22.3.16 PH_SEL1 (0x404D) (bit [7:6]) for details.

10.2.1 UART1 Operating Instructions

10.2.1.1 UART1 Mode0

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

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

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

10.2.1.2 UART1 Mode1

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

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

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

10.2.1.3 UART1 Mode2

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

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

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

10.2.1.4 UART1 Mode3

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

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

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

10.2.1.5 UART1 Interrupt Sources

UART1 interrupt sources include:

- After the data is transmitted via UART1, UT_CR[TI] is set to “1” by hardware.
- After the data and STOP are received via UART1, UT_CR[RI] is set to “1” by hardware.

10.2.2 UART2 Operating Instructions

10.2.2.1 UART2 Mode0

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

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

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

10.2.2.2 UART2 Mode1

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

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

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

10.2.2.3 UART2 Mode2

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

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

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

10.2.2.4 UART2 Mode3

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

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

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

10.2.2.5 UART2 Interrupt Sources

UART2 interrupt sources include:

- After the data is transmitted via UART2, UT2_CR[UT2TI] is set to "1" by hardware.
- After the data and STOP are received via UART2, UT2_CR[UT2RI] is set to "1" by hardware.

10.3 UART1 Registers

10.3.1 UT_CR (0x98)

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

Bit	Name	Description
[7:6]	MOD	Mode Selection 00: Mode0 01: Mode1 10: Mode2 11: Mode3
[5]	SM2	Communication Mode Selection 0: Single-device Communication 1: Multi-device Communication
[4]	REN	Receive Enable 0: Disable 1: Enable
[3]	TB8	Bit9 of the sent data in Mode2 and Mode3
[2]	RB8	Bit9 of the received data in Mode2 and Mode3
[1]	TI	Data Sending Completed Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: The interrupt is generated
[0]	RI	Data Receiving Completed Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: The interrupt is generated

10.3.2 UT_DR (0x99)

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

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

10.3.3 UT_BAUD (0x9A, 0x9B)

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

Bit	Name	Description
[15]	BAUD_SEL	Frequency Multiplier Enable 0: Disable 1: Enable
[14]	UART_RX_INV	Receive Inverting Enable 0: Disable 1: Enable
[13]	UART_TX_INV	Send Inverting Enable 0: Disable 1: Enable

[12]	UART_CH	UART Function Switching 0: P0.5 is configured as TXD and P0.6 as RXD 1: P1.0 is configured as TXD and P1.1 as RXD
[11:0]	BAUD	Baud Rate Setting Baud rate = SYSCLK/(16/(1 + UT_BAUD[BAUD_SEL]))/(UT_BAUD[BAUD] + 1) Example: If baud rate = 9600 and UT_BAUD[BAUD_SEL] = 0, then UT_BAUD[BAUD] = (24M/16/9600/(1 + 0)) - 1 = 155, i.e., 0x9B

10.4 UART2 Registers

10.4.1 UT2_CR (0xD8)

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

Bit	Name	Description
[7:6]	UT2MOD	Mode Selection 00: Mode0 01: Mode1 10: Mode2 11: Mode3
[5]	UT2SM2	Communication Mode Selection 0: Single-device Communication 1: Multi-device Communication
[4]	UT2REN	Receive Enable 0: Disable 1: Enable
[3]	UT2TB8	Bit9 of the sent data in Mode2 and Mode3
[2]	UT2RB8	Bit9 of the received data in Mode2 and Mode3
[1]	UT2TI	Data Sending Completed Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: The interrupt is generated
[0]	UT2RI	Data Receiving Completed Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending

		Write: 0: This bit is cleared to “0” 1: The interrupt is generated

10.4.2 UT2_DR (0x89)

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

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

10.4.3 UT2_BAUD (0x4042, 0x4043)

UT2_BAUDH(0x4042)								
Bit	15	14	13	12	11	10	9	8
Name	BAUD2_SEL	UART2_RX_INV	UART2_TX_INV	UART2 IEN	BAUD2[11:8]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
UT2_BAUDL(0x4043)								
Bit	7	6	5	4	3	2	1	0
Name	BAUD2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	1	0	1	1

Bit	Name	Description
[15]	BAUD2_SEL	Frequency Multiplier Enable 0: Disable 1: Enable
[14]	UART2_RX_INV	Receive Inverting Enable 0: Disable 1: Enable

[13]	UART2_TX_INV	Send Inverting Enable 0: Disable 1: Enable
[12]	UART2IEN	UART2 Interrupt Enable 0: Disable 1: Enable
[11:0]	BAUD2	Baud Rate Setting Baud rate = SYSCLK/(16/(1 + UT2_BAUD[BAUD2_SEL]))/(UT2_BAUD[BAUD2] + 1) Example: If baud rate = 9600 and UT2_BAUD[BAUD_SEL] = 0, then UT2_BAUD[BAUD2] = (24M/16/9600/(1 + 0)) - 1 = 155, i.e., 0x9B

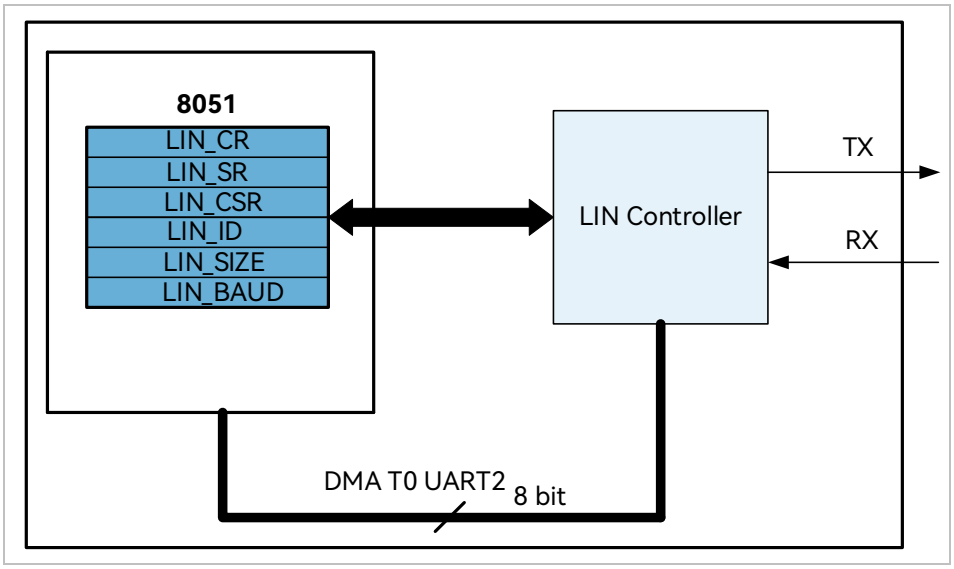
11 LIN

11.1 LIN Introduction

LIN is an asynchronous, serial communication interface mainly used in automotive network. LIN controller complies with the 2.2 Specification (backward compatible). As shown in Figure 11-1, the controller implements a complete LIN hardware interface, which works in slave mode and supports baud rate adaption. P0.0/P0.1 and P1.0/P1.1 (function switching) are translated into LIN pins, among which P1.1 is configured as RXD/TXD pin. Data transmission between LIN and DMA is possible by setting DMAx_CR0[DMACFG] = 11X, and LIN_CR[LINRW] bit determines the R/W direction.

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

Figure 11-1 Block Diagram of LIN Module



The LIN controller has three main components, as shown in Figure 11-1:

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

11.2 LIN Slave Mode Operations

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

When LIN Interrupt is enabled, an interrupt is generated in any of the following five cases as shown in Table 11-1.

Table 11-1 LIN Interrupt Sources and Descriptions

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

LIN slave transmits and receives the data as follows:

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

11.3 Sleep and Wakeup

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

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

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

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

11.4 Error Detection and Handling

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

11.5 Other Matters

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

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

11.6 LIN Registers

11.6.1 LIN_CR (0xB9)

Bit	7	6	5	4	3	2	1	0
Name	RSV	DMASEL	CHSEL		LINIE	CHKMOD	LINRW	AUTOSIZE
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	DMASEL	LIN DMA Channel Selection 0: DMA0 1: DMA1
[5:4]	CHSEL	LIN Mode and Port Pin Configuration 00: 2-wire mode, P0.0 configured as LTXD pin and P0.1 as LRXD pin 01: Reserved 10: 2-wire mode, P1.0 configured as LTXD pin and P1.1 as LRXD pin 11: 1-wire mode, P1.1 configured as LTXD/LRXD pin
[3]	LINIE	LIN Interrupt Enable 0: Disable 1: Enable
[2]	CHKMOD	Checksum Selection 0: Enhanced checksum 1: Classic checksum
[1]	LINRW	Send/Receive Selection 0: Current frame is an operation for receiving data 1: Current frame is an operation for sending data
[0]	AUTOSIZE	Data Length Dependent On (LIN_ID[5:4]) Enable 0: Disable 1: Enable Mapping between LIN_ID[5:4] and Data Length: 0X: 2 bytes 10: 4 bytes 11: 8 bytes

11.6.2 LIN_SR (0xBA)

Bit	7	6	5	4	3	2	1	0
Name	ERRSYNC	ERRCHK	ERRPRTY	ABORT	LINACT	LINIDLE	LINDONE	LINREQ
Type	R	R	R	R	R	R/W0	R/W0	R/W0
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	ERRSYNC	Sync Error (synchronization timeout or premature synchronization). This bit is cleared to “0” by hardware when a new frame arrives or when LIN[CLRERR] is cleared to “0”. 0: No sync error occurs 1: A sync error occurs
[6]	ERRCHK	Data Check Error. This bit is cleared to “0” by hardware when a new frame arrives or when LIN_CSR[CLRERR] is cleared to “0”. 0: No data check error occurs 1: A data check error occurs
[5]	ERRPRTY	ID Check Error. This bit is cleared to “0” by hardware when a new frame arrives or when LIN_CSR[CLRERR] is cleared to “0”. 0: No ID check error occurs 1: An ID check error occurs
[4]	ABORT	Aborted Transmission Flag This bit is set to “1” upon reception of a frame header or when LIN_CSR[LINSTOP] is set to “1” during transmission. 0: No aborted transmission occurs 1: An aborted transmission occurs
[3]	LINACT	LIN Bus Active Flag 0: No data is transmitting on LIN bus 1: Data is transmitting on LIN bus
[2]	LINIDLE	LIN Bus Idle Interrupt Flag This bit is set to “1” when the bus stays idle for more than 4s. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	LINDONE	Transmission Completion Interrupt Flag This bit is set to “1” after the slave receives or sends the data, and cleared to “0” when a new frame arrives or software writes “0” to LIN_SR[LINDONE]. Read: 0: No Interrupt Pending 1: Interrupt Pending

		Write: 0: This bit is cleared to “0” 1: No effect
[0]	LINREQ	Header Reception Interrupt Flag This bit is set to “1” after a frame header is received and its ID is correct. This bit is cleared to “0” when a new frame arrives, or software writes “1” to LIN_CSR[LINACK] or “0” to LIN_SR[LINREQ]. 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

11.6.3 LIN_CSR (0xBB)

Bit	7	6	5	4	3	2	1	0
Name	ERRBIT	RDBAKDIS	LINSLP	CLRERR	LINWAKEUP	LINACK	LINSTOP	LINEN
Type	R	R/W	R/W	R/W0	R/W	W1	W1	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	ERRBIT	Read-back Bit Error This bit is set to “1” when LIN read-back feature is enabled by setting LIN_CSR[RDBAKDIS] bit and the logic level of LIN TXD pin is different from LIN bus level. 0: No bit error occurs 1: A bit error occurs
[6]	RDBAKDIS	LIN Read-back Enable 0: Enable 1: Disable
[5]	LINSLP	LIN Sleep Mode Enable Read: 0: LIN is active 1: LIN in sleep mode Write: 0: LIN exits sleep mode 1: LIN enters sleep mode
[4]	CLRERR	LIN Error Clear Read: 0: No error occurs 1: An error occurs Write: 0: This error bit is cleared 1: No effect

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

11.6.4 LIN_ID (0xBC)

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

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

11.6.5 LIN_SIZE (0xBD)

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

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

11.6.6 LIN_BAUD (0xBF,0xBE)

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

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

12 CAN

12.1 CAN Introduction

CAN, known as Controller Area Network, implements as an ISO standard-compliant serial communication protocol, supporting safe, convenient and reliable data communication for automotive networks.

CAN controller reads the bus level according to voltage difference between the two bus conductors CAN_H and CAN_L. CAN bus voltage is either recessive or dominant. The transmitter sends messages to the receiver by changing the bus voltage.

CAN Protocol has the following features:

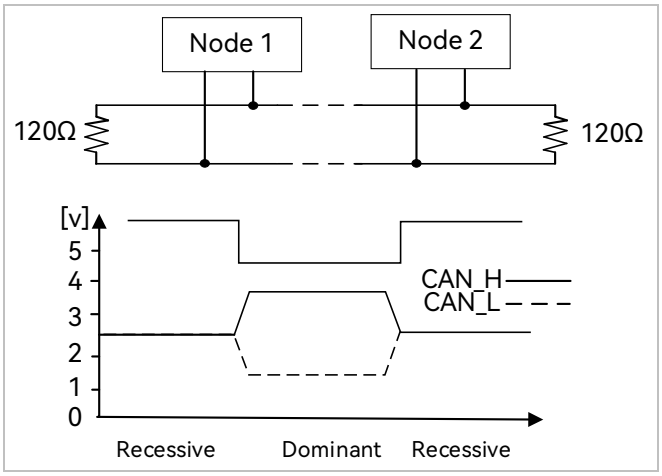
- Multi-master control: If there is no transmission activity on CAN bus, all devices (also known as nodes) can transmit messages (multi-master control). When two or more CAN nodes simultaneously attempt to transmit a message, the messages are transmitted in the order of their priority defined by Identifier ("ID"). The ID does not represent the destination address but the priority of messages accessing the CAN bus. In this case, an arbitration scheme is employed bit-by-bit for the ID of each frame. The node winning the arbitration is given higher priority and can continue transmitting, while the losing node stops transmitting and operates as a receiver.
- Flexibility: All nodes connected to CAN bus carry no address-alike message. In this case, the software/hardware and application layer of other nodes remain unchanged when a new node is added to the CAN bus. CAN supports bit rates up to 1Mbps (with a bus length of 40m) and bus lengths up to 10km (at a bit rate of lower than 5kbps).
- It is designed with error detection, notification and recovery features: All nodes can detect errors (error detection) and notify other nodes (error notification) immediately when an error is detected. A node with a message transfer underway holds the current frame whenever it detects an error, and attempts re-transmission repeatedly until it is delivered successfully (error recovery).
- Fault isolation: CAN determines whether an error is a transient data error (e.g., external noise) or a persistent data error (e.g., node-specific external fault, driver fault, disconnection) on the bus. With

this feature, if a persistent data error is detected over CAN bus, the node inducing the fault is isolated.

- Multiple nodes: Multiple nodes can be connected to CAN bus at the same time. The total number of nodes connected is theoretically not limited. However, it is limited by time delay and electrical load on CAN bus in practice. The number of nodes accessible on CAN bus decreases with the bit rate.

The figure below gives ISO11898-compliant physical layer characteristics of CAN.

Figure 12-1 ISO11898-compliant Physical Layer Characteristics



The above characteristics show that the dominant bus level corresponds to logic 0, with 2.5V voltage difference between CAN_H and CAN_L. The recessive bus level corresponds to logic 1, with 0V voltage difference between CAN_H and CAN_L. The dominant level has bus access priority, meaning that when any node sends a dominant bit, the result is dominant. The recessive level is somewhat inclusive so that the result is recessive only when all nodes send a recessive bit. In addition, an 120Ω termination resistor is mounted onto both start and stop points of CAN bus for impedance matching with an effort to minimize echoed/reflected signals.

12.1.1 Frame Structure

CAN Protocol is implemented with the following five types of frames:

- Data Frame
- Remote Frame
- Error Frame
- Overload Frame

Interframe Space

Both data frame and remote frame support standard and extended formats. Frames of standard format contain an 11-bit identifier, while frames of extended format contain a 29-bit identifier.

Table 12-1 CAN Protocol-supported Frame Types and Their Functions

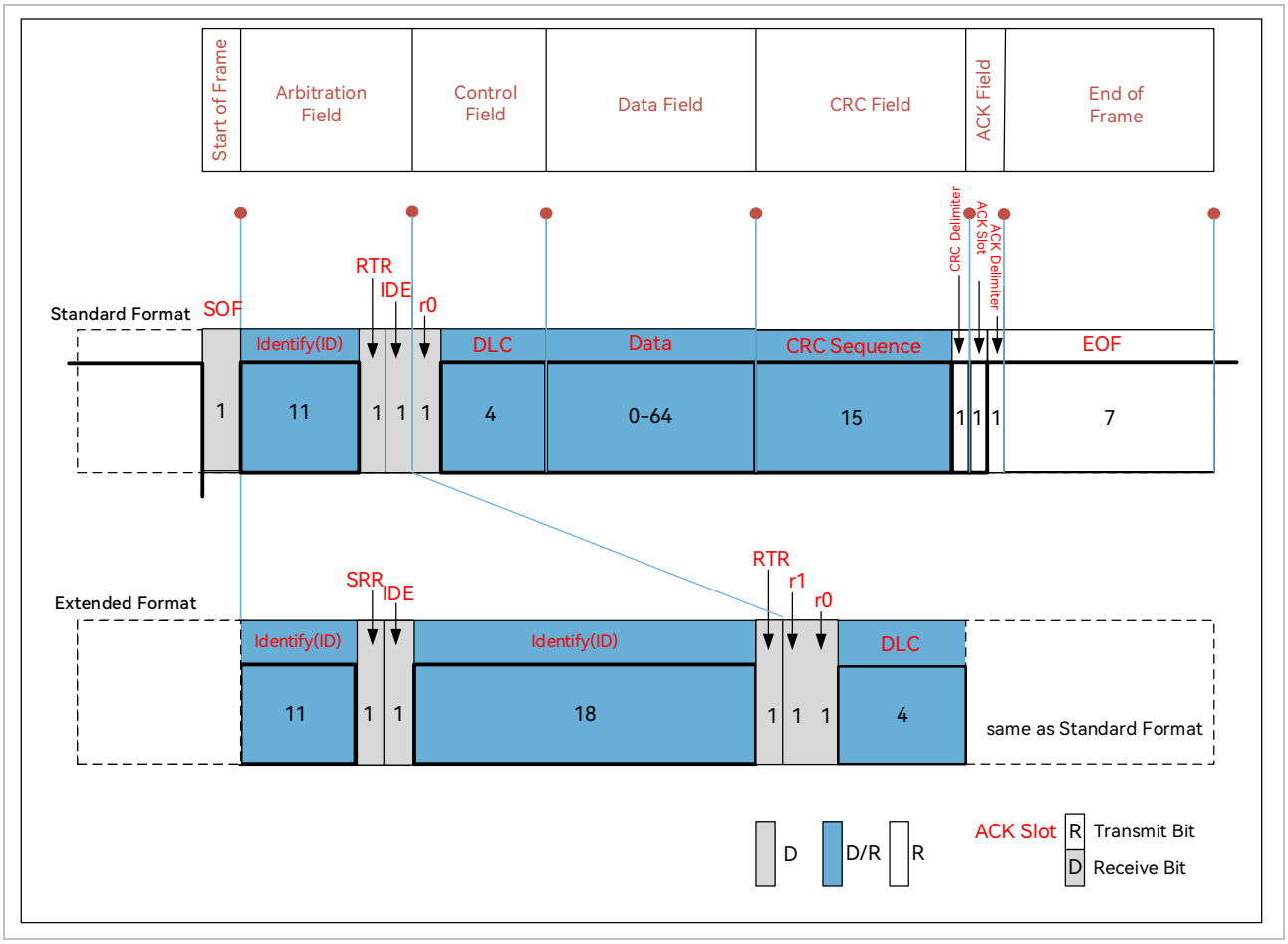
Type	Function
Data Frame	Transfer a frame from the transmitter to the receiver
Remote Frame	The receiver requests a transmission from the transmitter with same ID
Error Frame	Broadcast an error to all other nodes when a node detects a fault
Overload Frame	The receiver sends a frame to notify that it is busy
Interframe Space	Separate data frame and remote from the preceding frame

The following section provides a detailed introduction of data frame (other frames are omitted for the sake of brevity). Date frame typically comprises seven parts, including:

1. Start of Frame (SOF): signify the start of a frame
2. Arbitration Field: determine the priority of the current frame
3. Control Field: signify the bytes and reserved bit of a frame
4. Data Field: contain 0 ~ 8 bytes of data
5. CRC Field: detect errors
6. ACK Field: signify that a node has received the frame correctly
7. End of Frame (EOF): signify the end of a frame

The figure below presents the structure of data frame.

Figure 12-2 Structure of Data Frame



In the above diagram, D denotes dominant and R recessive.

Start of Frame: Both standard frame and extended frame delimit start of the frame with one dominant bit.

Arbitration Field: Determines the priority of a frame. The differences between a standard frame and an extended frame are described below:

Frames of standard format contain a 11-bit Identifier over a range from ID28 to ID18. Configuring all the first 7 bits as recessive is not possible (by setting ID = 1111111XXXX). An extended frame contains a 29-bit Identifier. The basic ID ranges from ID28 to ID18, while the extended ID falls within ID17 ~ ID0. The basic ID is same with that of standard frames. Configuring all the first 7 bits as recessive is not possible (by setting ID = 1111111XXXX). RTR bit is the Remote Transmission Request (0: data frame; 1: remote frame); IDE bit is the Identifier Extension (0: standard ID; 1: extended ID); and SRR bit is the Substitute Remote Request, which is recessive and replaces RTR bit in a standard frame. Standard frames have higher priority over

extended frames, while data frames have higher priority over remote frames.

Control Field: Comprise r0/r1 bit and DLC bit, in which, DLC bit is transmitted with MSB and indicates 0 ~ 8 bytes of data.

Data Field: Contain 0 ~ 8 bytes of data, which is transmitted with MSB. Definitions of this field are same between standard frame and extended frame.

CRC Field: Detect errors. It comprises a 15-bit CRC delimiter and a 1-bit CRC delimiter. CRC value is calculated using SOF, Arbitration Field, Control Field and Data Field. The receiver calculates CRC value in the same way and sends an CRC Error in case of a different CRC in the transmitted message than what it has calculated itself.

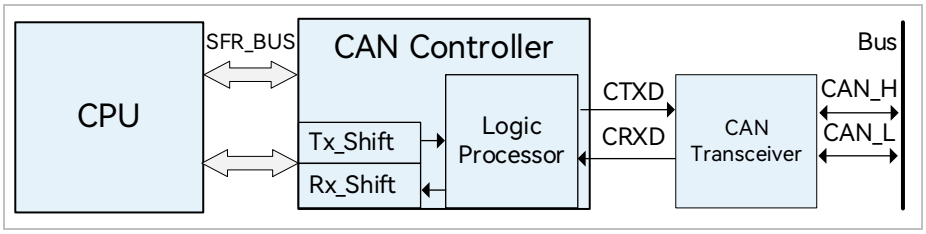
ACK Field: Acknowledge the node has received the frame correctly. It comprises ACK Slot and ACK Delimiter. The transmitter sends 2 recessive bits, and the receiver sends a dominant bit at the ACK Slot to broadcast the frame has been received correctly.

End of Frame: This simple field is defined identically between standard frame and extended frame, comprising 7 recessive bits.

12.2 General Descriptions on CAN Controller

The figure below presents topological structure of the CAN bus. CPU configures the CAN controller SFRs via SFR addressing, including initialization, bit time, operating mode, filtering and interrupt, and it completes communication between CAN module and Message XRAM so as to receive or transmit messages. The Message XRAM consists of a Transit Buffer and three Receive Buffers.

Figure 12-3 Topological Block Diagram of CAN Bus



12.2.1 Key Features

- Support CAN 2.0A and CAN 2.0B
- Bit rates up to 1Mb/s
- One Transit Buffer and three Receive Buffers; FIFO-oriented reception of message, where CPU can read the received messages using FIFO pointer CAN_CR2[RDPTR].
- Support Automatic Retransmission Quest (ARQ) Disable mode
- Sleep Mode with low-power consumption, supporting auto wakeup and manual wakeup
- Acceptance filtering in Single Filter and Double Filter modes
- Error processing

12.3 Operating Modes

12.3.1 Reset Mode

In Reset Mode, CAN controller is reset and cannot receive or send messages from or to CAN bus. The CAN controller enters the Reset Mode after CAN_CR0[RSTMOD] is set to “1”.

12.3.2 Normal Operation Mode

Before CAN controller enters the Normal Operation Mode, it is necessary to configure parameters for bit time characteristics (including CAN_BTR0, CAN_BTR1 and CAN_BTR2) and acceptance filtering (including CAN_CR0[FILMOD], CAN_FIRx and CAN_FMRx, x = 0,1,2,3) and program CAN_CR0[CAN_EN] = 1 to enable CTXD and CRXD pins and initialize CAN. After initialization, the CAN controller starts to synchronize itself to the CAN bus. Synchronization is implemented after CRXD detects a sequence of 11 consecutive recessive bits (= Bus Idle). After synchronization, CAN controller enters Normal Operation Mode and can receive or send messages from or to CAN bus.

12.3.3 Sleep Mode

CAN controller can operate in the power-efficient Sleep Mode. Programming CAN_CR1[SLPREQ] to “1” sends a Sleep Mode request. CAN controller enters the Sleep Mode only when there is no activity on the bus. It acknowledges the Sleep Mode by reading CAN_SR[SLPACK] bit. In this mode, CAN stops receive or

send messages but the Receive Buffer is accessible by software.

CAN provides two wake-up modes:

- Manual Wakeup by Software: Clearing CAN_CR1[SLPREQ] to “0” enables this feature
- Auto Wakeup by Hardware: Setting CAN_CR0[AWKMOD] to “1” enables this feature. When any activity is detected on the bus, CAN_CR1[SLPREQ] bit is cleared to “0” by hardware to wake up CAN controller. CAN controller re-synchronizes itself with CAN bus before exiting the Sleep Mode. Reading CAN_SR[SLPACK] bit acknowledges that CAN bus exits the Sleep Mode

12.4 Feature Descriptions

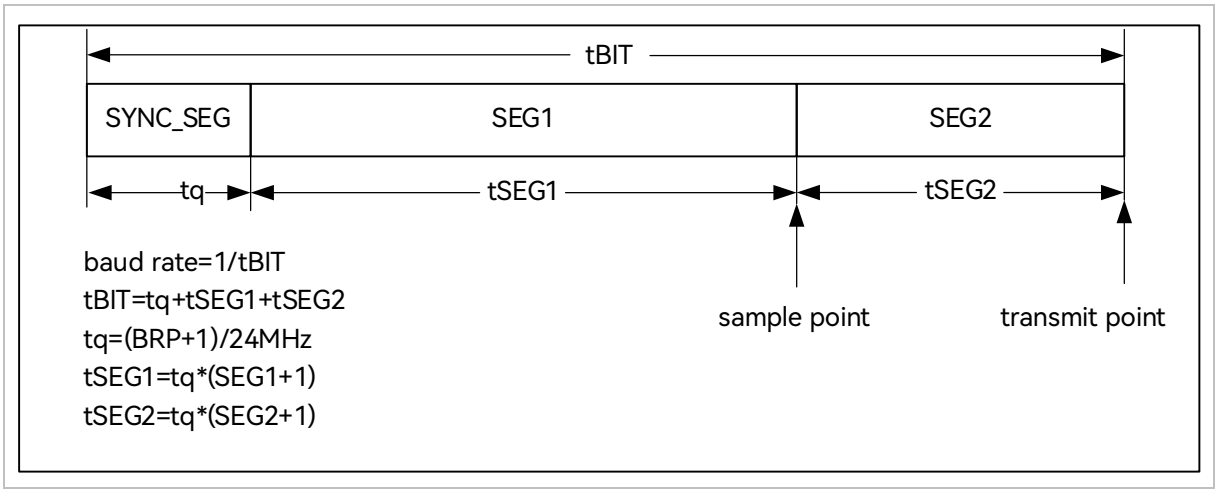
12.4.1 Bit-time Characteristics

CAN controller monitors the CAN bus sampling edges. It adjusts the Sample Point and Transit Point by synchronizing to the falling edge of Start of Frame (SOF) and re-synchronizing to the falling edge of the subsequent bits.

As shown in Figure 12-4, the bit time is divided into four segments: Synchronization Segment (SYNC_SEG), Time Segment 1 (SEG1) and Time Segment 2 (SEG2). The length of these segments is an integral multiple of time quanta (tq), the smallest unit of time defined by (CAN_BTR0[1:0] and CAN_BTR1).

1. SYNC_SEG: That part of the bit time where edges of CAN bus level are expected to occur. It is used for synchronizing different nodes on CAN bus. SYNC_SEG is a constant 1 tq.
2. SEG1: It is used for defining position of the Sample Point. It consists of the Propagation Time Segment (PROP_SEG) and Phase Buffer Segment 1 (PHASE_SEG1) specified in CAN Specification. SEG1 is programmed by CAN_BTR2[SEG1] over 1~16 tq, provided however that it can be automatically lengthened to compensate for forward phase drift caused by frequency difference between different nodes connected to CAN bus.
3. SEG2: It is used for defining position of the Transit Point. It is the Phase Buffer Segment 2 (PHASE_SEG2) specified in the CAN Specification. SEG2 is programmed by CAN_BTR2[SEG2] over the range of 1~8 tq, provided however that it can be automatically shortened to compensate for backward phase drift.

Figure 12-4 Structure of Bit Time



A synchronization error may occur during message transfer due to clock rate deviation, delay time and other factors. In this case, CAN_BTR0[SJW] is used to set tSJW for re-synchronization of different CAN nodes. tSJW is configured over the range of 1~4 tq. If an effective edge (from recessive to dominant) is detected in SEG1 instead of SYNC_SEG, SEG1 is lengthened up to the length of tSJW to shift the position of Sample Point backward. If an effective edge (from recessive to dominant) is detected in SEG2 instead of SYNC_SEG, SEG2 is shortened up to the length of tSJW to shift the position of Sample Point forward.

12.4.2 Sending Management

Messages are configured by software from the Transit Buffer in Message XRAM. A transit request is generated after CAN_CR1[TXREQ] is set to “1”. And the controller starts transfer the message to CAN bus when it is idle. An ACK bit signaled from CAN nodes indicates that the message has been successfully transmitted. CAN_SR[TXSUC] bit is an indicator for successful transmission, and CAN_SR[TXDONE] bit is an indicator for transmission completion.

In Auto Re-transmission (ARQ) mode, if the transmission fails due to arbitration lost or error, the node re-transmits the message when CAN bus is idle until the transmission is completed.

The ARQ Disable mode is activated by programming CAN_CR1[NOART] to “1”, where the message is transmitted just once.

12.4.3 Receiving Management

CAN controller consists of three Receive Buffers, including Receive Buffer0, Receive Buffer1 and Receive

Buffer2. Each buffer contains the following bits: CAN_RXxCR, CAN_RXxID0, CAN_RXxID1, CAN_RXxID2, CAN_RXxID3, CAN_RXxDR0, CAN_RXxDR1, CAN_RXxDR2, CAN_RXxDR3, CAN_RXxDR4, CAN_RXxDR5, CAN_RXxDR6 and CAN_RXxDR7(x = 0,1,2). CAN controller implements a FIFO-oriented reception of messages.

After resetting, FIFO write pointer targets Receive Buffer0. During the reception, the hardware automatically writes bytes to Receive Buffer targeted by the FIFO writer pointer. If no error is detected until the last byte of EOF and the message passes acceptance filtering, the message is considered valid and FIFO write pointer targets the next Receive Buffer. If FIFO write pointer currently targets Receive Buffer2, the pointer re-targets Receive Buffer0. When an error is detected or acceptance filtering fails, FIFO write pointer remains unchanged, suggesting that the next data packet is written to the original Receive Buffer. A Receive Buffer is full when it is loaded with three sets of message object. In such case, the buffer overruns when receiving the next data packet which will be discarded.

CAN_CR2[MESCNT] is an indicator of valid message counter, and CAN_CR2[RDPTR] is an indicator of the buffer targeted by FIFO read pointer. The software access data bytes by reading Receive Buffers. After that, CAN_CR[BUFRLS] is set to “1” to free the Receive Buffer targeted by FIFO read pointer which then points to the next Receive Buffer. If FIFO read pointer currently points to Receive Buffer2, it re-points to Receive Buffer0.

12.4.4 Receive Filtering

The Filter Mode, Identifier and Mask Bit are configured to receive desired messages only.

CAN_CR0[FILMOD] bit selects between the Single Filter mode and the Double Filter mode. In single filter mode, only one filter works for acceptance filtering. In double filter mode, two filters work for acceptance filtering.

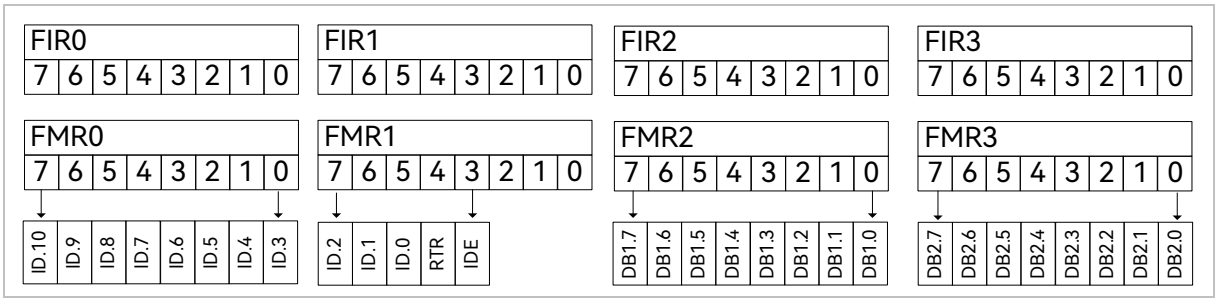
The Identifier is defined by CAN_FIRx (x = 0 ~ 3).

The Mask Bit is defined by CAN_FMRx (x = 0 ~ 3). If CAN_FMRx bit contains a “0”, the data packet must match with CAN_FIRx bit; otherwise, the data packet will be filtered; and if CAN_FMRx bit contains a “1”, it has no meaning.

12.4.4.1 Standard Frame Single Filter Mode

In Standard Frame Single Filter Mode, the filtering part contains a 11-bit ID, RTR, IDE and the first two bytes of data (DB0 and DB1), with its configuration shown in Figure 12-5. If Data Field of a message reads “0” in length, the filter feature of CAN_FIR2, CAN_FMR2, CAN_FIR3 and CAN_FMR3 is disabled; if the Data Field reads “1”, the filter feature of CAN_FIR3 and CAN_FMR3 is disabled.

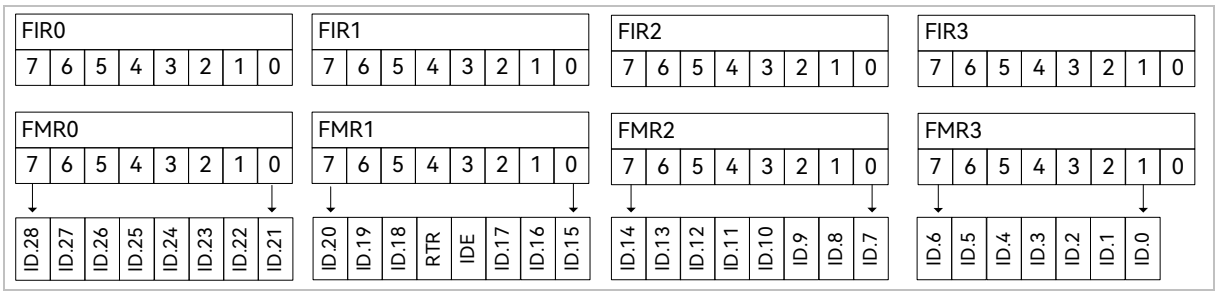
Figure 12-5 Standard Frame Single Filter Mode



12.4.4.2 Extended Frame Single Filter Mode

In Extended Frame Single Filter mode, the filtering part contains a 29-bit ID, RTR and IDE, with its configuration shown in Figure 12-6.

Figure 12-6 Extended Frame Single Filter Mode

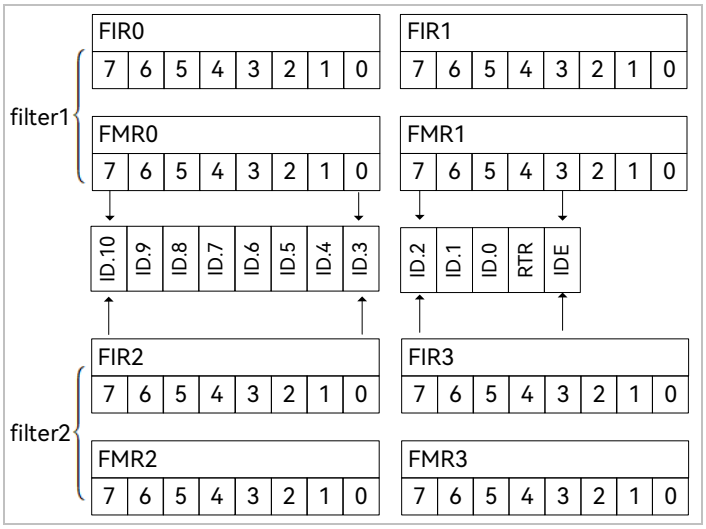


In this mode, if CAN is configured to receive standard frames only, CAN_FIR1[3] and CAN_FMR1[3] are set to “0”, and CAN_FIR2, CAN_FMR2, CAN_FIR3 and CAN_FMR3 are set to filter frames containing byte 0 or 1. If CAN is configured to receive extended frames only, CAN_FIR1[3] is set to “1”, CAN_FMR1[3] to “0”, and CAN_FIR1[2:0], CAN_FMR1[2:0], CAN_FIR2, CAN_FMR2, CAN_FIR3 and CAN_FMR3 are set to filter data containing extended frames carrying ID[17:0]. If CAN is configured to receive both standard and extended frames, CAN_FIR1[3] are set to “0” or “1”, and CAN_FMR1[3] is set to “1”.

12.4.4.3 Standard Frame Double Filter Mode

In Standard Frame Double Filter mode, both the Filter1 and Filter 2 contain a 11-bit ID, RTR and IDE respectively, with its configuration shown in Figure 12-7.

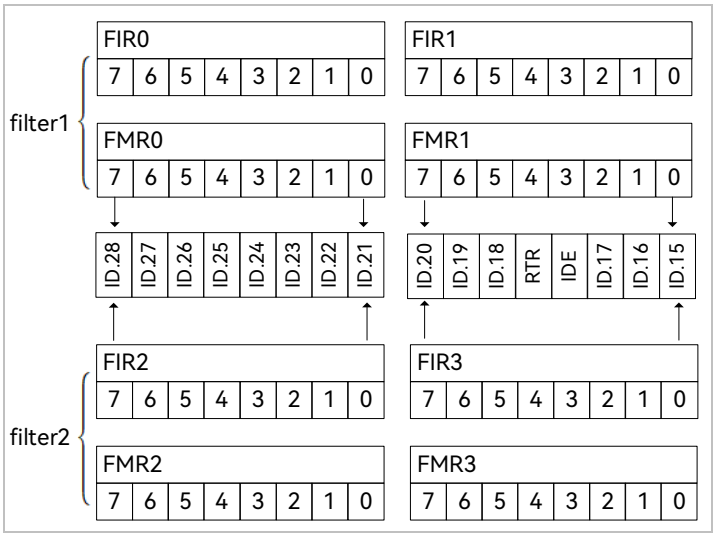
Figure 12-7 Standard Frame Double Filter Mode



12.4.4.4 Extended Frame Double Filter Mode

In Extended Frame Double Filter mode, the filtering part contains a 14-bit ID, RTR and IDE, with its configuration shown in Figure 12-8.

Figure 12-8 Extended Frame Double Filter Mode

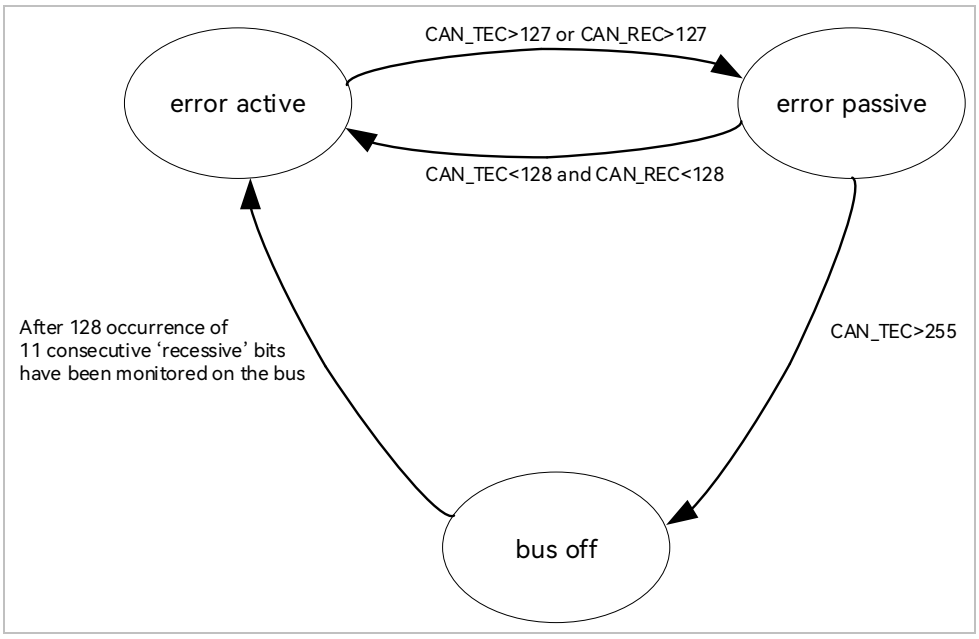


In this mode, CAN_FIR1[2:0], CAN_FMR1[2:0], CAN_FIR3[2:0] and CAN_FMR3[2:0] bits are configured for filtering extended frames only.

12.4.5 Error Management

CAN error processing is performed by hardware sending the Transmit Error Counter CAN_TEC and Receive Error Counter CAN_REC. CAN_TEC and CAN_REC increase or decrease with the errors. Error states are divided into Error Active, Error Passive and Busoff, as shown in Figure 12-9. In Busoff state, CAN cannot receive or transmit messages. The current error state and error type are accessed by reading CAN_ESR.

Figure 12-9 CAN Error Processing



12.5 CAN Interrupt Sources

CAN supports 10 interrupt sources. An interrupt is generated by setting the corresponding Enable bit of CAN_IER and CAN_CR2. After an interrupt is generated, the Interrupt Flag of CAN_IFR is set to "1".

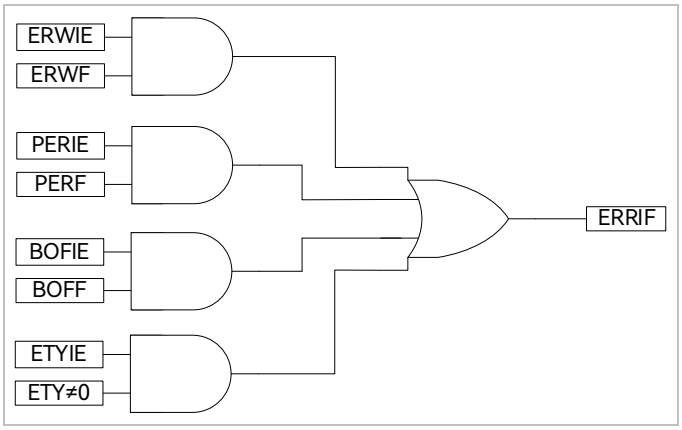
- Receive Interrupt: If no error is detected until the last byte of EOF and the frame passes acceptance filtering, the current frame received is considered valid and CAN_IFR[RXIF] is set to "1". Receive Interrupt is enabled by setting CAN_IER[RXIE] to "1" to generate an interrupt.
- Transmit Interrupt: CAN_IFR[TXIF] is set to "1" at the end of each transmission. Transmit Interrupt is enabled by setting CAN_IER[TXIE] to "1" to generate an interrupt.
- Overflow Interrupt: In the case of FIFO overflow, CAN_IFR[OVIF] is set to "1". Overflow Interrupt is enabled by setting CAN_IER[OVIE] to "1" to generate an interrupt.
- Arbitration Lost Interrupt: CAN_IFR[ARBIF] is set to "1" when a node loses arbitration to other nodes

and thus loses priority in CAN bus access. Arbitration Lost Interrupt is enabled by programming CAN_IER[ARBIE] to “1” to generate an interrupt.

- Sleep Mode Interrupt: CAN_IFR[SLPIF] is set to “1” when CAN controller enters the Sleep Mode. Sleep Mode Interrupt is enabled by setting CAN_CR2[SLPIE] to “1” to generate an interrupt.
- Wakeup Interrupt: In Auto Wakeup mode, CAN_IFR[WKUIF] is set to “1” when a SOF bit is detected. Wakeup Interrupt is enabled by programming CAN_CR0[AWKMOD] to “1” and CAN_CR2[WKUIE] to “1” to generate an interrupt.
- Error Interrupt
  - Error Warning Interrupt: The Error Warning Flag CAN_ESR[ERWF] is set to “1” when CAN_TEC ≥ CAN_LIM or CAN_REC ≥ CAN_LIM. An interrupt is generated if Error Warning Interrupt is enabled by setting CAN_IER[ERWIE] to “1”, Error Interrupt is enabled by setting CAN_CR2[ERRIE] to “1” and an error is detected.
  - Error Passive Interrupt: The Error Passive Flag CAN_ESR[PERF] is set to “1” when CAN_TEC > 127 or CAN_REC > 127. An interrupt is generated if Error Passive Interrupt is enabled by setting CAN_IER[PERIE] to “1”, Error Interrupt is enabled by setting CAN_CR2[ERRIE] to “1” and an error is detected.
  - Busoff Interrupt: The Busoff Flag CAN_ESR[BOFF] is set to “1” when CAN_TEC > 255. An interrupt is generated if Busoff Interrupt is enabled by setting CAN_IER[BOFIE] to “1”, Error Interrupt is enabled by setting CAN_CR2[ERRIE] to “1” and an error is detected.
  - Error Type Interrupt: When any of bit error, format error, stuff error, ACK error and CRC error is detected, CAN_ESR[ETY] changes into a non-zero value indicating the error type. An interrupt is generated if Error Type Interrupt is enabled by setting CAN_IER[ETYIE] to “1”, Error Interrupt is enabled by setting CAN_CR2[ERRIE] to “1” and an error is detected.

Error Interrupts are divided into Error Warning Interrupt, Error Passive Interrupt, Busoff Interrupt and Error Type Interrupt. Relationship among CAN_IFR[ERRIF], CAN_ESR[ERWF], CAN_ESR[PERF], CAN_ESR[BOFF] and CAN_ESR[ETY] bits and their Enable bits are illustrated in Figure 12-10. CAN_IFR[ERRIF] bit is an Error Interrupt indicator, while CAN_ESR[ERWF], CAN_ESR[PERF], CAN_ESR[BOFF] and CAN_ESR[ETY] are status indicators. Thus, when an Error Interrupt is triggered, the error information is accessed by reading the status flag bits. CAN_IFR[ERRIF] bit is cleared by software.

Figure 12-10 Relationship among ERRIF, ERWF, PERF, BOFF and ETY Bits and Their Enable Bits



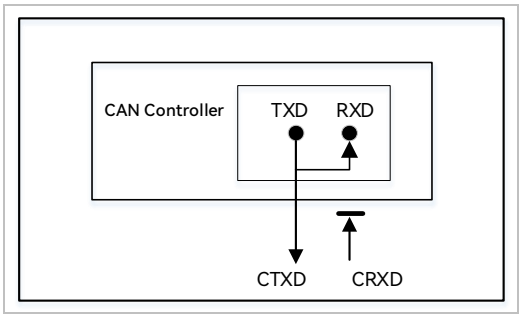
12.6 Test Mode

CAN controller can operate in Self-test Mode, Monitoring Mode and Self-test Mode & Monitoring Mode by programming CAN_CR0[SELFTST] and CAN_CR0[LISTEN] bits.

12.6.1 Self-test Mode

CAN controller operates in Self-test Mode after CAN_CR0[SELFTST] is set to “1” and CAN_CR0[LISTEN] is cleared to “0”. The connection diagram is shown in Figure 12-11. CRXD pin is disconnected from CAN controller. CAN controller performs internal feedback from its CTXD output to its CRXD input, and ignores the actual status on CRXD pin and ACK errors (recessive bit not sampled in the ACK slot of a frame). In this mode, when CAN_CR1[TXSELF] is set to “1”, CAN controller internally reroutes the sent frames and writes back to a Receive Buffer if the frames are considered valid. When CAN_CR1[TXREQ] is set to “1”, CAN controller only sends the frames and does not perform a write-back to any Receive Buffer, regardless of whether the frames are valid or not.

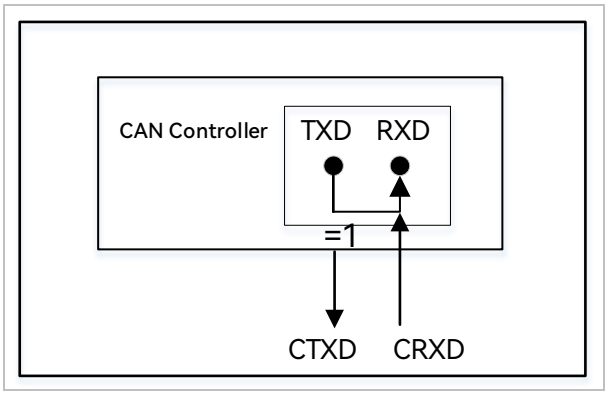
Figure 12-11 Self-test Mode



12.6.2 Monitoring Mode

CAN controller operates in Monitoring Mode when CAN_CR0[SELFTST] is set to “0” and CAN_CR0[LISTEN] to “1”. The connection diagram is shown in Figure 12-12. In this mode, CTXD pin is driven HIGH (recessive). CAN controller receives the data frames normally, but the dominant bit transmitted is rerouted internally instead of sending to CTXD pin, so the frames cannot be transferred to CAN bus. It is used to analyze CAN bus activities without affecting the bus.

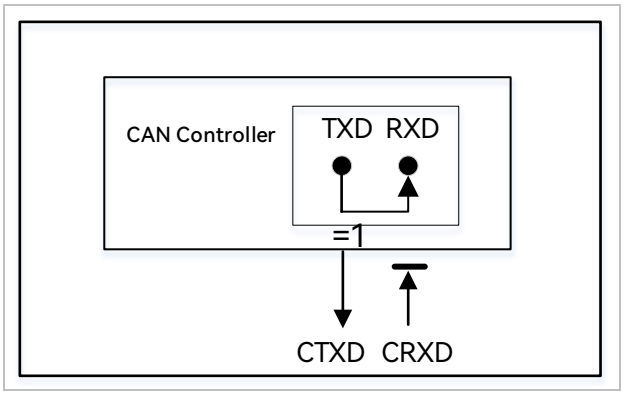
Figure 12-12 Monitoring Mode



12.6.3 Self-test Mode & Monitoring Mode

CAN controller operates in Self-test Mode & Monitoring Mode when CAN_CR0[SELFTST] and CAN_CR0[LISTEN] are set to “1” at the same time. The connection diagram is shown in Figure 12-13. CRXD pin is disconnected from CAN controller and CTXD pin is driven HIGH (recessive). In this mode, CAN controller operates without affecting CAN bus, performs internal feedback from its TXD output to its RXD input, and ignores ACK errors.

Figure 12-13 Self-test Mode & Monitoring Mode



12.7 CAN Registers

12.7.1 CAN_CR0 (0xB1)

Bit	7	6	5	4	3	2	1	0
Name	CANEN	CANCH	RSV	AWKMOD	FILMOD	SELFTST	LISTEN	RSTMOD
Type	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Reset	0	0	-	0	0	0	0	1

Bit	Name	Description
[7]	CANEN	CTXD and CRXD Pin Multiplex and CAN Enable 0: No Multiplexing; CAN Disabled. 1: Multiplexing; CAN enabled and initialized.
[6]	CANCH	CAN Pin Selection 0: P0.0 as CTXD pin and P0.1 as CRXD pin 1: P0.5 as CTXD pin and P0.6 as CRXD pin
[5]	RSV	Reserved
[4]	AWKMOD	Auto Wakeup Enable When Auto Wakeup feature is enabled, CAN_CR1[SLPREQ] and CAN_IFR[SLPACK] are automatically cleared to “0” after CAN controller detects a frame to wake up the device. 0: Disable 1: Enable
[3]	FILMOD	Filter Mode 0: Double Filter Mode 1: Single Filter Mode
[2]	SELFTST	Self-test Mode Enable 0: Disable 1: Enable
[1]	LISTEN	Monitoring Mode Enable 0: Disable 1: Enable
[0]	RSTMOD	Reset Mode Read: 0: CAN controller is not in Reset Mode 1: CAN controller is in Reset Mode Write: 0: After initialization, CAN controller synchronizes itself with CAN bus to enter Normal Operation Mode. 1: Enter the Reset Mode

12.7.2 CAN_CR1 (0xB2)

Bit	7	6	5	4	3	2	1	0
Name	RSV		SLPREQ	TXSELF	RSV	BUFRLS	NOART	TXREQ
Type	-	-	R/W	R/W	-	W	R/W	R/W
Reset	-	-	0	0	-	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	SLPREQ	Sleep Mode Request CAN controller requests to enter Sleep Mode when this bit is programmed to “1” by software. CAN controller enters the Sleep Mode after processing (transmitting or receiving) the current frame on CAN bus. A request to exit the Sleep Mode is sent by programming this bit to”0”, after which CAN controller starts re-synchronization to the CAN bus. 0: No request pending 1: Request pending
[4]	TXSELF	It is valid only when CAN_CR1[TXREQ] is set to “0” in the Self-test Mode. 0: No request pending 1: Request pending
[3]	RSV	Reserved
[2]	BUFRLS	Free Receive Buffer targeted by the read pointer CAN_CR2[RDPTR]. Write: 0: Do not free the Receive Buffer, and the read pointer CAN_CR2[RDPTR] targets the original buffer 1: Free the Receive Buffer, and the read pointer CAN_CR2[RDPTR] targets the next buffer
[1]	NOART	ARQ Disable 0: Disable, where messages are automatically re-transmitted if the transmission fails. 1: Enable, where messages are transmitted just once no matter the transmission is successful or not.
[0]	TXREQ	Transmission Request 0: No request pending 1: Request pending

12.7.3 CAN_SR (0xB3)

Bit	7	6	5	4	3	2	1	0
Name	RSV	SLPACK	TXING	RXING	TXSUC	TXDONE	BUFOV	NOEMP
Type	-	R	R	R	R	R	R	R
Reset	-	0	0	0	1	1	0	0

Bit	Name	Description

[7]	RSV	Reserved
[6]	SLPACK	Sleep Mode Acknowledge CAN controller enters Sleep Mode when this bit is programmed to “1” by hardware. It is used to acknowledge a Sleep Mode request by software. This bit is cleared to “0” by hardware after CAN controller exits the Sleep Mode. Read: 0: CAN is not in Sleep Mode 1: CAN is in Sleep Mode Write: 0: This bit is cleared to “0” and CAN controller exits the Sleep Mode 1: No effect
[5]	TXING	Transmitting Flag 0: CAN is not transmitting messages 1: CAN is transmitting messages
[4]	RXING	Receiving Flag 0: CAN is not receiving messages 1: CAN is receiving messages
[3]	TXSUC	Successful Transmission Flag It is cleared to “0” by hardware. 0: The transmission fails 1: The transmission succeeds
[2]	TXDONE	Transmission Completed Flag It is cleared to “0” by hardware. 0: Transmission is not completed. 1: Transmission is completed.
[1]	BUFOV	FIFO Overflow Flag 0: FIFO does not overflow 1: FIFO overflows
[0]	NOEMP	FIFO Empty Flag 0: FIFO is empty 1: FIFO is not empty

12.7.4 CAN_IER (0xB4)

Bit	7	6	5	4	3	2	1	0
Name	LECIE	BOFIE	PERIE	ERWIE	OVIE	ARBIE	TXIE	RXIE
Type	R/W	R/W	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]	ETYIE	Error Type Interrupt Enable 0: Disable 1: Enable

[6]	BOFIE	Busoff Interrupt Enable 0: Disable 1: Enable
[5]	PERIE	Passive Error Interrupt Enable 0: Disable 1: Enable
[4]	ERWIE	Error Warning Interrupt Enable 0: Disable 1: Enable
[3]	OVIE	Overflow Interrupt Enable 0: Disable 1: Enable
[2]	ARBIE	Arbitration Lost Interrupt 0: Disable 1: Enable
[1]	TXIE	Transmit Completed Interrupt Enable 0: Disable 1: Enable
[0]	RXIE	Receive Interrupt Enable 0: Disable 1: Enable

12.7.5 CAN_CR2 (0xB5)

Bit	7	6	5	4	3	2	1	0
Name	RDPTR		MESCNT		RSV	WKUIE	SLPIE	ERRIE
Type	R	R	R	R	-	R/W	R/W	R/W
Reset	0	0	0	0	-	0	0	0

Bit	Name	Description
[7:6]	RDPTR	FIFO Read Pointer 00: FIFO read pointer targets Receive Buffer0 01: FIFO read pointer targets Receive Buffer1 10: FIFO read pointer targets Receive Buffer2 11: Reserved
[5:4]	MESCNT	Valid Message Counter
[3]	RSV	Reserved
[2]	WKUIE	Wakeup Interrupt Enable 0: Disable 1: Enable
[1]	SLPIE	Sleep Mode Interrupt Enable 0: Disable 1: Enable

[0]	ERRIE	Error Interrupt Enable 0: Disable 1: Enable

12.7.6 CAN_IFR (0xB6)

Bit	7	6	5	4	3	2	1	0
Name	RSV	WKUIF	SLPIF	ERRIF	OVIF	ARBIF	TXIF	RXIF
Type	-	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R
Reset	-	0	0	0	0	0	0	0

Bit	Name	Description
[7]	RSV	Reserved
[6]	WKUIF	Wakeup Interrupt Flag This bit is set to “1” whenever CAN controller detects a SOF bit in Auto Wakeup mode Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[5]	SLPIF	Sleep Mode Interrupt Flag This bit is programmed to “1” when CAN controller enters the Sleep Mode. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	ERRIF	Error Interrupt Flag This bit is set to “1” when an Error Interrupt is enabled and CAN controller detects an associated error. See “CAN Interrupt Sources” for details. This bit is cleared to “0” by software. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0”. 1: No effect
[3]	OVIF	Overflow Interrupt Flag Read: 0: No Interrupt Pending

		1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	ARBIF	Arbitration Lost Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	TXIF	Transmit Completed Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	RXIF	Receive Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Note: This bit can be set to “1” by hardware only and cannot be cleared to “0” by software. It is cleared to “0” by setting CAN_CR1[BUFRLS] to “1” to free Receive Buffers.

12.7.7 CAN_BTR0 (0xB7)

Bit	7	6	5	4	3	2	1	0
Name	RSV				SJW		BRP[9:8]	
Type	-	-	-	-	R/W	R/W	R/W	R/W
Reset	-	-	-	-	0	0	0	0

Bit	Name	Description
[7:4]	RSV	Reserved
[3:2]	SJW	For resynchronization purpose, this bit defines the maximum of bit time (tSJW) that can be lengthened or shortened for CAN nodes. tSJW = tq*(SJW + 1)
[1:0]	BRP[9:8]	Two high-order bits in the Baud Rate Prescaler (BPR)

Note:

tq = (BRP[9:0]+1)/24MHz

12.7.8 CAN_BTR1 (0xC2)

Bit	7	6	5	4	3	2	1	0
Name	BRP[7:0]							
Type	R/W	R/W	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]	BRP[7:0]	Eight low-order bits in the BRP

12.7.9 CAN_BTR2 (0xC3)

Bit	7	6	5	4	3	2	1	0
Name	TTCM	SEG2			SEG1			
Type	R/W	R/W	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]	TTCM	Activate Sample Mode by Two out of Three Options 0: Disable 1: Enable
[6:4]	SEG2	tSEG2 = tq*(SEG2 + 1)
[3:0]	SEG1	tSEG1 = tq*(SEG1 + 1)

12.7.10 CAN_FIR0 (0xC4)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FIR0							
Type	R/W	R/W	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]	CAN_FIR0	Filter ID Register0

12.7.11 CAN_FIR1 (0xC5)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FIR1							
Type	R/W	R/W	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]	CAN_FIR1	Filter ID Register1

12.7.12 CAN_FIR2 (0xC6)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FIR2							
Type	R/W	R/W	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]	CAN_FIR1	Filter ID Register2

12.7.13 CAN_FIR3 (0xC7)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FIR3							
Type	R/W	R/W	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]	CAN_FIR3	Filter ID Register3

12.7.14 CAN_FMR0 (0xCA)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FMR0							
Type	R/W	R/W	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]	CAN_FMR0	Filter Mask Register0 0: It is required to match the Identifier Register CAN_FIR0. 1: It is not required to match the Identifier Register CAN_FIR0.

12.7.15 CAN_FMR1 (0xCB)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FMR1							
Type	R/W	R/W	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]	CAN_FMR1	Filter Mask Register1 0: It is required to match the Identifier Register CAN_FIR1. 1: It is not required match the Identifier Register CAN_FIR1.

12.7.16 CAN_FMR2 (0xCC)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FMR2							
Type	R/W	R/W	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]	CAN_FMR2	Filter Mask Register2 0: It is required to match the Identifier Register CAN_FIR2. 1: It is not required to match the Identifier Register CAN_FIR2.

12.7.17 CAN_FMR3 (0xCD)

Bit	7	6	5	4	3	2	1	0
Name	CAN_FMR3							
Type	R/W	R/W	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]	CAN_FMR3	Filter Mask Register3 0: It is required to match the Identifier Register CAN_FIR3. 1: It is not required to match the Identifier Register CAN_FIR3.

12.7.18 CAN_ALC (0xCE)

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

Bit	Name	Description
[7:5]	RSV	Reserved
[4:0]	CAN_ALC	Arbitration lost counter flag to indicate the position of arbitration lost

12.7.19 CAN_ESR (0xCF)

Bit	7	6	5	4	3	2	1	0
Name	ETY			RSV		BOFF	PERF	ERWF
Type	R	R	R	-	-	R	R	R
Reset	0	0	0	-	-	0	0	0

Bit	Name	Description

[7:5]	ETY	Error Type 000: No Error 001: Bit Error 010: Format Error 011: Stuff Error 100: ACK error 101: CRC error 110: Reserved 111: Reserved
[4:3]	RSV	Reserved
[2]	BOFF	Busoff Flag 0: CAN module is not in busoff state. 1: CAN module is in busoff state.
[1]	PERF	Error Passive 0: The CAN module is error active when CAN_TEC ≤ 127 and CAN_REC ≤ 127 1: The CAN module is error passive when CAN_TEC > 127 or CAN_REC > 127
[0]	ERWF	Error Warning Flag 0: No error warning occurs when CAN_TEC < CAN_LIM and CAN_REC < CAN_LIM. 1: Error warning occurs when CAN_TEC ≥ CAN_LIM or CAN_REC ≥ CAN_LIM.

12.7.20 CAN_LIM (0xDD)

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

Bit	Name	Description
[7:0]	CAN_LIM	Error warning count flag. An error warning is generated when CAN_TEC ≥ CAN_LIM or CAN_REC ≥ CAN_LIM

12.7.21 CAN_TEC (0xDE)

Bit	7	6	5	4	3	2	1	0
Name	CAN_TEC							
Type	R/W	R/W	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]	CAN_TEC	Transmit Error Counter

12.7.22 CAN_REC (0xDF)

Bit	7	6	5	4	3	2	1	0
Name	CAN_REC							
Type	R/W	R/W	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]	CAN_REC	Receive Error Counter

12.7.23 CAN_TXCR (0x0ED7)

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

Bit	Name	Description
[7]	IDE	Identifier Extension Flag 0: Standard frame 1: Extended frame
[6]	RTR	Remote Transfer Request 0: Data frame 1: Remote frame
[5:4]	RSV	Reserved
[3:0]	DLC	Data length code with valid values 0 ~ 8

12.7.24 CAN_TXID0 (0x0ED6)

Bit	7	6	5	4	3	2	1	0
Name	STDID[10:3]/EXTID[28:21]							
Type	R/W	R/W	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]	STDID[10:3]/ EXTID[28:21]	CAN_TXCR[IDE] decides these bits are standard identifier bits or extended identifier bits

12.7.25 CAN_TXID1 (0x0ED5)

Bit	7	6	5	4	3	2	1	0
Name	STDID[2:0]/EXTID[20:18]			EXTID[17:13]				
Type	R/W	R/W	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]	STDID[2:0]/ EXTID[20:18]	CAN_TXCR[IDE] decides these bits are standard identifier bits or extended identifier bits
[4:0]	EXTID[17:13]	Bits 13 ~ 17 of the extended identifier

12.7.26 CAN_TXID2 (0x0ED4)

Bit	7	6	5	4	3	2	1	0
Name	EXTID[12:5]							
Type	R/W	R/W	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]	EXTID[12:5]	Bits 5 ~ 12 of the extended identifier

12.7.27 CAN_TXID3 (0x0ED3)

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

Bit	Name	Description
[7:3]	EXTID[4:0]	Bits 0 ~ 4 of the extended identifier
[2:0]	RSV	Reserved

12.7.28 CAN_TXDR0 (0x0ED2)

Bit	7	6	5	4	3	2	1	0
Name	DATA0							
Type	R/W	R/W	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]	DATA0	Data byte 0 of CAN message

12.7.29 CAN_TXDR1 (0x0ED1)

Bit	7	6	5	4	3	2	1	0
Name	DATA1							
Type	R/W	R/W	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]	DATA1	Data byte 1 of CAN message

12.7.30 CAN_TXDR2 (0x0ED0)

Bit	7	6	5	4	3	2	1	0
Name	DATA2							
Type	R/W	R/W	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]	DATA2	Data byte 2 of CAN message

12.7.31 CAN_TXDR3 (0x0ECF)

Bit	7	6	5	4	3	2	1	0
Name	DATA3							
Type	R/W	R/W	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]	DATA3	Data byte 3 of CAN message

12.7.32 CAN_TXDR4 (0x0ECE)

Bit	7	6	5	4	3	2	1	0
Name	DATA4							
Type	R/W	R/W	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]	DATA4	Data byte 4 of CAN message

12.7.33 CAN_TXDR5 (0x0ECD)

Bit	7	6	5	4	3	2	1	0
Name	DATA5							
Type	R/W	R/W	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]	DATA5	Data byte 5 of CAN message

12.7.34 CAN_TXDR6 (0x0ECC)

Bit	7	6	5	4	3	2	1	0
Name	DATA6							
Type	R/W	R/W	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]	DATA6	Data byte 6 of CAN message

12.7.35 CAN_TXDR7 (0x0ECB)

Bit	7	6	5	4	3	2	1	0
Name	DATA7							
Type	R/W	R/W	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]	DATA7	Data byte 7 of CAN message

12.7.36 CAN_RX0CR (0x0ECA)

Bit	7	6	5	4	3	2	1	0
Name	IDE	RTR	RSV		DLC			
Type	R	R	-	-	R	R	R	R
Reset	0	0	-	-	0	0	0	0

Bit	Name	Description
[7]	IDE	Identifier Extension Flag 0: Standard frame 1: Extended frame
[6]	RTR	Remote Transfer Request 0: Data frame 1: Remote frame

[5:4]	RSV	Reserved
[3:0]	DLC	Data length code with valid values 0 ~ 8

12.7.37 CAN_RX0ID0 (0x0EC9)

Bit	7	6	5	4	3	2	1	0
Name	STDID[10:3]/EXTID[28:21]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	STDID[10:3]/ EXTID[28:21]	CAN_RX0CR[IDE] decides these bits are standard identifier bits or extended identifier bits

12.7.38 CAN_RX0ID1 (0x0EC8)

Bit	7	6	5	4	3	2	1	0
Name	STDID[2:0]/EXTID[20:18]			EXTID[17:13]				
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:5]	STDID[2:0]/ EXTID[20:18]	CAN_RX0CR[IDE] decides these bits are standard identifier bits or extended identifier bits
[4:0]	EXTID[17:13]	Bits 13 ~ 17 of the extended identifier

12.7.39 CAN_RX0ID2 (0x0EC7)

Bit	7	6	5	4	3	2	1	0
Name	EXTID[12:5]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	EXTID[12:5]	Bits 5 ~ 12 of the extended identifier

12.7.40 CAN_RX0ID3 (0x0EC6)

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

Bit	Name	Description

[7:3]	EXTID[4:0]	Bits 0 ~ 4 of the extended identifier
[2:0]	RSV	Reserved

12.7.41 CAN_RX0DR0 (0x0EC5)

Bit	7	6	5	4	3	2	1	0
Name	DATA0							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA0	Data byte 0 of CAN message

12.7.42 CAN_RX0DR1 (0x0EC4)

Bit	7	6	5	4	3	2	1	0
Name	DATA1							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA1	Data byte 1 of CAN message

12.7.43 CAN_RX0DR2 (0x0EC3)

Bit	7	6	5	4	3	2	1	0
Name	DATA2							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA2	Data byte 2 of CAN message

12.7.44 CAN_RX0DR3 (0x0EC2)

Bit	7	6	5	4	3	2	1	0
Name	DATA3							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA3	Data byte 3 of CAN message

12.7.45 CAN_RX0DR4 (0x0EC1)

Bit	7	6	5	4	3	2	1	0
Name	DATA4							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA4	Data byte 4 of CAN message

12.7.46 CAN_RX0DR5 (0x0EC0)

Bit	7	6	5	4	3	2	1	0
Name	DATA5							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA5	Data byte 5 of CAN message

12.7.47 CAN_RX0DR6 (0x0EBF)

Bit	7	6	5	4	3	2	1	0
Name	DATA6							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA6	Data byte 6 of CAN message

12.7.48 CAN_RX0DR7 (0x0EBE)

Bit	7	6	5	4	3	2	1	0
Name	DATA7							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA7	Data byte 7 of CAN message

12.7.49 CAN_RX1CR (0x0EBD)

Bit	7	6	5	4	3	2	1	0
Name	IDE	RTR	RSV		DLC			
Type	R	R	-	-	R	R	R	R
Reset	0	0	-	-	0	0	0	0

Bit	Name	Description
[7]	IDE	Identifier Extension Flag 0: Standard frame 1: Extended frame
[6]	RTR	Remote Transfer Request 0: Data frame 1: Remote frame
[5:4]	RSV	Reserved
[3:0]	DLC	Data length code with valid values 0 ~ 8

12.7.50 CAN_RX1ID0 (0x0EBC)

Bit	7	6	5	4	3	2	1	0
Name	STDID[10:3]/EXTID[28:21]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	STDID[10:3]/ EXTID[28:21]	CAN_RX1CR[IDE] decides these bits are standard identifier bits or extended identifier bits

12.7.51 CAN_RX1ID1 (0x0EBB)

Bit	7	6	5	4	3	2	1	0
Name	STDID[2:0]/EXTID[20:18]			EXTID[17:13]				
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:5]	STDID[2:0]/ EXTID[20:18]	CAN_RX1CR[IDE] decides these bits are standard identifier bits or extended identifier bits
[4:0]	EXTID[17:13]	Bits 13 ~ 17 of the extended identifier

12.7.52 CAN_RX1ID2 (0x0EBA)

Bit	7	6	5	4	3	2	1	0
Name	EXTID[12:5]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	EXTID[12:5]	Bits 5 ~ 12 of the extended identifier

12.7.53 CAN_RX1ID3 (0x0EB9)

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

Bit	Name	Description
[7:3]	EXTID[4:0]	Bits 0 ~ 4 of the extended identifier
[2:0]	RSV	Reserved

12.7.54 CAN_RX1DR0 (0x0EB8)

Bit	7	6	5	4	3	2	1	0
Name	DATA0							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA0	Data byte 0 of CAN message

12.7.55 CAN_RX1DR1 (0x0EB7)

Bit	7	6	5	4	3	2	1	0
Name	DATA1							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA1	Data byte 1 of CAN message

12.7.56 CAN_RX1DR2 (0x0EB6)

Bit	7	6	5	4	3	2	1	0
Name	DATA2							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA2	Data byte 2 of CAN message

12.7.57 CAN_RX1DR3 (0x0EB5)

Bit	7	6	5	4	3	2	1	0
Name	DATA3							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA3	Data byte 3 of CAN message

12.7.58 CAN_RX1DR4 (0x0EB4)

Bit	7	6	5	4	3	2	1	0
Name	DATA4							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA4	Data byte 4 of CAN message

12.7.59 CAN_RX1DR5 (0x0EB3)

Bit	7	6	5	4	3	2	1	0
Name	DATA5							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA5	Data byte 5 of CAN message

12.7.60 CAN_RX1DR6 (0x0EB2)

Bit	7	6	5	4	3	2	1	0
Name	DATA6							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA6	Data byte 6 of CAN message

12.7.61 CAN_RX1DR7 (0x0EB1)

Bit	7	6	5	4	3	2	1	0
Name	DATA7							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA7	Data byte 7 of CAN message

12.7.62 CAN_RX2CR (0x0EB0)

Bit	7	6	5	4	3	2	1	0
Name	IDE	RTR	RSV		DLC			
Type	R	R	-	-	R	R	R	R
Reset	0	0	-	-	0	0	0	0

Bit	Name	Description
[7]	IDE	Identifier Extension Flag 0: Standard frame 1: Extended frame
[6]	RTR	Remote Transfer Request 0: Data frame 1: Remote frame
[5:4]	RSV	Reserved
[3:0]	DLC	Data length code with valid values 0 ~ 8

12.7.63 CAN_RX2ID0 (0x0EAF)

Bit	7	6	5	4	3	2	1	0
Name	STDID[10:3]/EXTID[28:21]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	STDID[10:3]/ EXTID[28:21]	CAN_RX2CR[IDE] decides these bits are standard identifier bits or extended identifier bits

12.7.64 CAN_RX2ID1 (0x0EAE)

Bit	7	6	5	4	3	2	1	0
Name	STDID[2:0]/EXTID[20:18]			EXTID[17:13]				
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:5]	STDID[2:0]/ EXTID[20:18]	CAN_RX2CR[IDE] decides these bits are standard identifier bits or extended identifier bits
[4:0]	EXTID[17:13]	Bits 13 ~ 17 of the extended identifier

12.7.65 CAN_RX2ID2 (0x0EAD)

Bit	7	6	5	4	3	2	1	0
Name	EXTID[12:5]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	EXTID[12:5]	Bits 5 ~ 12 of the extended identifier

12.7.66 CAN_RX2ID3 (0x0EAC)

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

Bit	Name	Description
[7:3]	EXTID[4:0]	Bits 0 ~ 4 of the extended identifier
[2:0]	RSV	Reserved

12.7.67 CAN_RX2DR0 (0x0EAB)

Bit	7	6	5	4	3	2	1	0
Name	DATA0							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA0	Data byte 0 of CAN message

12.7.68 CAN_RX2DR1 (0x0EAA)

Bit	7	6	5	4	3	2	1	0
Name	DATA1							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA1	Data byte 1 of CAN message

12.7.69 CAN_RX2DR2 (0x0EA9)

Bit	7	6	5	4	3	2	1	0
Name	DATA2							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA2	Data byte 2 of CAN message

12.7.70 CAN_RX2DR3 (0x0EA8)

Bit	7	6	5	4	3	2	1	0
Name	DATA3							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA3	Data byte 3 of CAN message

12.7.71 CAN_RX2DR4 (0x0EA7)

Bit	7	6	5	4	3	2	1	0
Name	DATA4							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA4	Data byte 4 of CAN message

12.7.72 CAN_RX2DR5 (0x0EA6)

Bit	7	6	5	4	3	2	1	0
Name	DATA5							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA5	Data byte 5 of CAN message

12.7.73 CAN_RX2DR6 (0x0EA5)

Bit	7	6	5	4	3	2	1	0
Name	DATA6							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA6	Data byte 6 of CAN message

12.7.74 CAN_RX2DR7 (0x0EA4)

Bit	7	6	5	4	3	2	1	0
Name	DATA7							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DATA7	Data byte 7 of CAN message

13 MDU

13.1 MDU Introduction

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

13.2 MDU Features

The MDU module has the following features:

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

13.3 MDU Instructions

13.3.1 MDU Operations

MDU is operated as follows.

1. Configure MDU_CR[MDUMOD] register to select computing mode of the MDU module;

2. Write the data to the associated computing units, and configure MDU_CR[MDUSTA] to select computing unit of the MDU module, and start MDU computing;

3. Wait for MDU_CR[MDUBUSY] to be cleared to “0” by hardware.

Note:

The computing mode and other data must be properly configured before setting MDU_CR[MDUSTA].

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

When MDU_CR[MDUMOD] = 000, MDU module works in the 16-bit signed multiplication mode with the result shifted left by one-bit. As shown in Table 13-1, after 16-bit signed data is written to MULx_MA and MULx_MB as multiplicand and multiplier respectively, 32-bit signed data is obtained by the product shifting left by one bit. The result is accessed by reading MULx_MC register.

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

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

13.3.3 16-bit Signed Multiplication

When MDU_CR[MDUMOD] = 001, MDU module works in the 16-bit signed multiplication mode. As shown in Table 13-2, 31-bit signed data is obtained after 16-bit signed data is written to MULx_MA and MULx_MB as multiplicand and multiplier respectively. The result is accessed by reading MULx_MC register.

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

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

13.3.4 16-bit Unsigned Multiplication

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

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

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

13.3.5 32-bit/16-bit Unsigned Division

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

Table 13-4 Register Definitions in the Unsigned Division Mode

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

13.3.6 LPF

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

The calculation formula of LPF is:

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

Where,

Y_k: Filtered data

Y_{k-1}: Previous filtered output

K: Filter coefficient

X_k: Data to be filtered

As shown in Table 13-5, Y_k and Y_{k-1} are 32-bit signed data, X_k and K are 16-bit signed data. Y_k is obtained after Y_{k-1} is written to LPFx_Y, K to LPFx_K and X_k to LPFx_X, and is accessed by reading LPFx_Y.

Table 13-5 Register Definitions in LPF Mode

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

13.3.7 Coordinate Transformation

When MDU_CR[MDUMOD] = 100, MDU module works in Coordinate Transformation mode. As shown in Figure 13-1, the coordinate transformation converts the components cos_i and sin_i of vector A under x-y axis to the components cos_o and sin_o under the x'-y' axis, with the x'-y' axis lagging the x-y axis by θ.

The formula for coordinate transformation is:

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

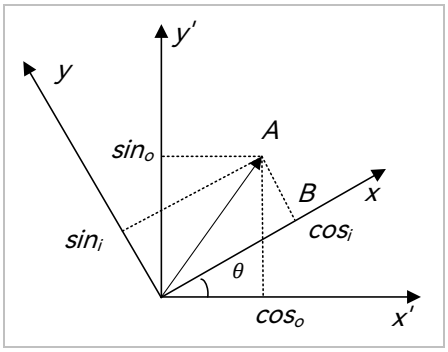
$$sin_o = cos_i \times sin\,\theta + sin_i \times cos\,\theta$$

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

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

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

Figure 13-1 Coordinate Transformation



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

Table 13-6 Register Definitions in the Coordinate Transformation Mode

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

13.3.8 Arctangent

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

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

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

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

Where,

Usinθ: Sin component of the vector

Ucosθ: Cosine component of the vector

θ: Calculated vector angle

U: Calculated vector amplitude

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

Table 13-7 Register Definitions in ATAN Mode

Data Register	Input	Output
SCATx_COS	*Ucosθ*	-
SCATx_SIN	*Usinθ*	-
SCATx_RES1	-	*U*
SCATx_RES2	-	*θ*

13.3.9 PI/PID

13.3.9.1 PI/PID Introduction

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

PI algorithm:

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

PID algorithm:

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

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

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

13.3.9.2 PI/PID Features

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

13.3.9.3 PI/PID Operations

1. Initialize MDU before operations, and configure K_p, K_i, K_d and the maximum and minimum values of U_k;

2. Set MDU_CR[MDUMOD] to 111, and then select Comp_Unit0 and Comp_Unit1 as PI Mode, and Comp_Unit2 and Comp_Unit3 as PID Mode. Later, configure MDU_CR[MDUSTA] to select the desired computing unit and start PI/PID computing. At this time, the busy flag MDU_CR[MDUBUSY] is automatically set to "1".

3. Read MDU_CR[MDUBUSY] by software. When this bit is "0", it indicates that the calculation is completed, and the calculation result PIx_UK is updated.

4. Read PIx_UK to obtain the output.

Note:

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

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

13.4 MDU Registers

13.4.1 MDU_CR (0xC1)

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

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

13.4.2 MUL0_MA (0x0FA0, 0x0FA1)

MUL0_MAH(0x0FA0)								
Bit	15	14	13	12	11	10	9	8
Name	MUL0_MA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL0_MAL(0x0FA1)								
Bit	7	6	5	4	3	2	1	0
Name	MUL0_MA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL0_MA	Data register A of MUL0; Multiplicand of the multiplication

13.4.3 MUL0_MB (0x0FA2, 0x0FA3)

MUL0_MBH(0x0FA2)								
Bit	15	14	13	12	11	10	9	8
Name	MUL0_MB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL0_MBL(0x0FA3)								
Bit	7	6	5	4	3	2	1	0
Name	MUL0_MB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

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

MUL0_MCHH(0x0FA4)								
Bit	31	30	29	28	27	26	25	24
Name	MUL0_MC[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL0_MCHL(0x0FA5)								
Bit	23	22	21	20	19	18	17	16
Name	MUL0_MC[13:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL0_MCLH(0x0FA6)								
Bit	15	14	13	12	11	10	9	8
Name	MUL0_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL0_MCLL(0x0FA7)								
Bit	7	6	5	4	3	2	1	0
Name	MUL0_MC[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description

[31:0]	MUL0_MC	Product of MUL0. The 16 high-order bits of the data are held by MUL0_MCH, and the 16 low-order bits by MUL0_MCL.

13.4.5 MUL1_MA (0x0F98, 0x0F99)

MUL1_MAH(0x0F98)								
Bit	15	14	13	12	11	10	9	8
Name	MUL1_MA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL1_MAL(0x0F99)								
Bit	7	6	5	4	3	2	1	0
Name	MUL1_MA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL1_MA	Data register A of MUL1; Multiplicand of the multiplication

13.4.6 MUL1_MB (0x0F9A, 0x0F9B)

MUL1_MBH(0x0F9A)								
Bit	15	14	13	12	11	10	9	8
Name	MUL1_MB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL1_MBL(0x0F9B)								
Bit	7	6	5	4	3	2	1	0
Name	MUL1_MB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL1_MB	Data register B of MUL1; Multiplier of the multiplication

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

MUL1_MCHH(0x0F9C)								
Bit	31	30	29	28	27	26	25	24
Name	MUL1_MC[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL1_MCHL(0x0F9D)								
Bit	23	22	21	20	19	18	17	16

Name	MUL1_MC[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL1_MCLH(0x0F9E)								
Bit	15	14	13	12	11	10	9	8
Name	MUL1_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL1_MCLL(0x0F9F)								
Bit	7	6	5	4	3	2	1	0
Name	MUL1_MC[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	MUL1_MC	Product of MUL1. The 16 high-order bits of the data are held by MUL1_MCH, and the 16 low-order bits by MUL1_MCL.

13.4.8 MUL2_MA (0x0F40, 0x0F41)

MUL2_MAH(0x0F40)								
Bit	15	14	13	12	11	10	9	8
Name	MUL2_MA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MAL(0x0F41)								
Bit	7	6	5	4	3	2	1	0
Name	MUL2_MA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL2_MA	Data register A of MUL2; Multiplicand of the multiplication

13.4.9 MUL2_MB (0x0F42, 0x0F43)

MUL2_MBH(0x0F42)								
Bit	15	14	13	12	11	10	9	8
Name	MUL2_MB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MBL(0x0F43)								
Bit	7	6	5	4	3	2	1	0

Name	MUL2_MB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL2_MB	Data register B of MUL2; Multiplier of the multiplication

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

MUL2_MCHH(0x0F44)								
Bit	31	30	29	28	27	26	25	24
Name	MUL2_MC[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MCHL(0x0F45)								
Bit	23	22	21	20	19	18	17	16
Name	MUL2_MC[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MCLH(0x0F46)								
Bit	15	14	13	12	11	10	9	8
Name	MUL2_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL2_MCLL(0x0F47)								
Bit	7	6	5	4	3	2	1	0
Name	MUL2_MC[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	MUL2_MC	Product of MUL2. The 16 high-order bits of the data are held by MUL2_MCH, and the 16 low-order bits by MUL2_MCL.

13.4.11 MUL3_MA (0x0F38, 0x0F39)

MUL3_MAH(0x0F38)								
Bit	15	14	13	12	11	10	9	8
Name	MUL3_MA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MAL(0x0F39)								
Bit	7	6	5	4	3	2	1	0

Name	MUL3_MA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL3_MA	Data register A of MUL3; Multiplicand of the multiplication

13.4.12 MUL3_MB (0x0F3A, 0x0F3B)

MUL3_MBH(0x0F3A)								
Bit	15	14	13	12	11	10	9	8
Name	MUL3_MB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MBL(0x0F3B)								
Bit	7	6	5	4	3	2	1	0
Name	MUL3_MB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	MUL3_MB	Data register B of MUL3; Multiplier of the multiplication

13.4.13 MUL3_MC (0x0F3C, 0x0F3D, 0x0F3E, 0x0F3F)

MUL3_MCHH(0x0F3C)								
Bit	31	30	29	28	27	26	25	24
Name	MUL3_MC[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MCHL(0x0F3D)								
Bit	23	22	21	20	19	18	17	16
Name	MUL3_MC[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MCLH(0x0F3E)								
Bit	15	14	13	12	11	10	9	8
Name	MUL3_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
MUL3_MCLL(0x0F3F)								
Bit	7	6	5	4	3	2	1	0

Name	MUL3_MC[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

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

DIV0_DAHH(0x0F8C)								
Bit	31	30	29	28	27	26	25	24
Name	DIV0_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_DAHL(0x0F8D)								
Bit	23	22	21	20	19	18	17	16
Name	DIV0_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_DALH(0x0F8E)								
Bit	15	14	13	12	11	10	9	8
Name	DIV0_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_DALL(0x0F8F)								
Bit	7	6	5	4	3	2	1	0
Name	DIV0_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	DIV0_DA	Dividend of DIV0. The 16 high-order bits of the data are held by DIV0_DAH, and the 16 low-order bits by DIV0_DAL.

13.4.15 DIV0_DB (0x0F90, 0x0F91)

DIV0_DBH(0x0F90)								
Bit	15	14	13	12	11	10	9	8
Name	DIV0_DB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

DIV0_DBL(0x0F91)								
Bit	7	6	5	4	3	2	1	0
Name	DIV0_DB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	DIV0_DB	Data register B of DIV0; Divisor of the division

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

DIV0_DQHH(0x0F92)								
Bit	31	30	29	28	27	26	25	24
Name	DIV0_DQ[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_DQHL(0x0F93)								
Bit	23	22	21	20	19	18	17	16
Name	DIV0_DQ[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_DQLH(0x0F94)								
Bit	15	14	13	12	11	10	9	8
Name	DIV0_DQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_DQLL(0x0F95)								
Bit	7	6	5	4	3	2	1	0
Name	DIV0_DQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	DIV0_DQ[31:0]	Quotient of DIV0. The 16 high-order bits of the data are held by DIV0_DQH, and the 16 low-order bits by DIV0_DQL.

13.4.17 DIV0_DR (0x0F96, 0x0F97)

DIV0_DRH(0x0F96)								
Bit	15	14	13	12	11	10	9	8
Name	DIV0_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV0_DRL(0x0F97)								
Bit	7	6	5	4	3	2	1	0
Name	DIV0_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]	DIV0_DR	Remainder of DIV0

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

DIV1_DAHH(0x0F80)								
Bit	31	30	29	28	27	26	25	24
Name	DIV1_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DAHL(0x0F81)								
Bit	23	22	21	20	19	18	17	16
Name	DIV1_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DALH(0x0F82)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DALL(0x0F83)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	DIV1_DA	Dividend of DIV1. The 16 high-order bits of the data are held by DIV1_DA, and the 16 low-order bits by DIV1_DA

13.4.19 DIV1_DB (0x0F84, 0x0F85)

DIV1_DBH(0x0F84)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DBL(0x0F85)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

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

DIV1_DQHH(0x0F86)								
Bit	31	30	29	28	27	26	25	24
Name	DIV1_DQ[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DQHL(0x0F87)								
Bit	23	22	21	20	19	18	17	16
Name	DIV1_DQ[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DQLH(0x0F88)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DQLL(0x0F89)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	DIV1_DQ	Quotient of DIV1. The 16 high-order bits of the data are held by DIV1_DQH, and the 16 low-order bits by DIV1_DQL

13.4.21 DIV1_DR (0x0F8A, 0x0F8B)

DIV1_DRH(0x0F8A)								
Bit	15	14	13	12	11	10	9	8
Name	DIV1_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV1_DRL(0x0F8B)								
Bit	7	6	5	4	3	2	1	0
Name	DIV1_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	DIV1_DR	Remainder of DIV1

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

DIV2_DAHH(0x0F2C)								
Bit	31	30	29	28	27	26	25	24
Name	DIV2_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DAHL(0x0F2D)								
Bit	23	22	21	20	19	18	17	16
Name	DIV2_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DALH(0x0F2E)								
Bit	15	14	13	12	11	10	9	8
Name	DIV2_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DALL(0x0F2F)								
Bit	7	6	5	4	3	2	1	0
Name	DIV2_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	DIV2_DA	Dividend of DIV2. The 16 high-order bits of the data are held by DIV2_DA, and the 16 low-order bits by DIV2_DA

13.4.23 DIV2_DB (0x0F30, 0x0F31)

DIV2_DBH(0x0F30)								
Bit	15	14	13	12	11	10	9	8
Name	DIV2_DB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DBL(0x0F31)								
Bit	7	6	5	4	3	2	1	0
Name	DIV2_DB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	DIV2_DB	Data register B of DIV2; Divisor of the division

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

DIV2_DQHH(0x0F32)								
Bit	31	30	29	28	27	26	25	24
Name	DIV2_DQ[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DQHL(0x0F33)								
Bit	23	22	21	20	19	18	17	16
Name	DIV2_DQ[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DQLH(0x0F34)								
Bit	15	14	13	12	11	10	9	8
Name	DIV2_DQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DQLL(0x0F35)								
Bit	7	6	5	4	3	2	1	0
Name	DIV2_DQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	DIV2_DQ	Quotient of DIV2. The 16 high-order bits of the data are held by DIV2_DQH, and the 16 low-order bits by DIV2_DQL

13.4.25 DIV2_DR (0x0F36, 0x0F37)

DIV2_DRH(0x0F36)								
Bit	15	14	13	12	11	10	9	8
Name	DIV2_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV2_DRL(0x0F37)								
Bit	7	6	5	4	3	2	1	0
Name	DIV2_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	DIV2_DR	Remainder of DIV2

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

DIV3_DAHH(0x0F20)								
Bit	31	30	29	28	27	26	25	24
Name	DIV3_DA[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DAHL(0x0F21)								
Bit	23	22	21	20	19	18	17	16
Name	DIV3_DA[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DALH(0x0F22)								
Bit	15	14	13	12	11	10	9	8
Name	DIV3_DA[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DALL(0x0F23)								
Bit	7	6	5	4	3	2	1	0
Name	DIV3_DA[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.27 DIV3_DB (0x0F24, 0x0F25)

DIV3_DBH(0x0F24)								
Bit	15	14	13	12	11	10	9	8
Name	DIV3_DB[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DBL(0x0F25)								
Bit	7	6	5	4	3	2	1	0
Name	DIV3_DB[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	DIV3_DB	Data register B of DIV3; Divisor of the division

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

DIV3_DQHH(0x0F26)								
Bit	31	30	29	28	27	26	25	24
Name	DIV3_DQ[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DQHL(0x0F27)								
Bit	23	22	21	20	19	18	17	16
Name	DIV3_DQ[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DQLH(0x0F28)								
Bit	15	14	13	12	11	10	9	8
Name	DIV3_DQ[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DQLL(0x0F29)								
Bit	7	6	5	4	3	2	1	0
Name	DIV3_DQ[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	DIV3_DQ	Quotient of DIV3. The 16 high-order bits of the data are held by DIV3_DQH, and the 16 low-order bits by DIV3_DQL

13.4.29 DIV3_DR (0x0F2A, 0x0F2B)

DIV3_DRH(0x0F2A)								
Bit	15	14	13	12	11	10	9	8
Name	DIV3_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
DIV3_DRL(0x0F2B)								
Bit	7	6	5	4	3	2	1	0
Name	DIV3_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	DIV3_DR	Remainder of DIV3

13.4.30 SCAT0_COS (0x0F16, 0x0F17)

SCAT0_COSH(0x0F16)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT0_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT0_COSL(0x0F17)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT0_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.31 SCAT0_SIN (0x0F18, 0x0F19)

SCAT0_SINH(0x0F18)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT0_SIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT0_SINL(0x0F19)								
Bit	7	6	5	4	3	2	1	0

Name	SCAT0_SIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT0_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT0

13.4.32 SCAT0_THE (0x0F1A, 0x0F1B)

SCAT0_THEH(0x0F1A)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT0_THE[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT0_THEL(0x0F1B)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT0_THE[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT0_THE	THE input in SIN/COS mode of computing unit SCAT0

13.4.33 SCAT0_RES1 (0x0F1C, 0x0F1D)

SCAT0_RES1H(0x0F1C)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT0_RES1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT0_RES1L(0x0F1D)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT0_RES1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT0_RES1	COS output in SIN/COS mode of computing unit SCAT0; U output in ATAN mode

13.4.34 SCAT0_RES2 (0x0F1E, 0x0F1F)

SCAT0_RES2H(0x0F1E)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT0_RES2[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT0_RES2L(0x0F1F)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT0_RES2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT0_RES2	SIN output in SIN/COS mode of computing unit SCAT0; θ output in ATAN mode

13.4.35 SCAT1_COS (0x0F0C, 0x0F0D)

SCAT1_COSH(0x0F0C)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT1_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT1_COSL(0x0F0D)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT1_COS[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT1_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT1

13.4.36 SCAT1_SIN (0x0F0E, 0x0F0F)

SCAT1_SINH(0x0F0E)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT1_SIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT1_SINL(0x0F0F)								
Bit	7	6	5	4	3	2	1	0

Name	SCAT1_SIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT1_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT1

13.4.37 SCAT1_THE (0x0F10, 0x0F11)

SCAT1_THEH(0x0F10)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT1_THE[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT1_THEL(0x0F11)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT1_THE[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT1_THE	THE input in SIN/COS mode of computing unit SCAT1

13.4.38 SCAT1_RES1 (0x0F12, 0x0F13)

SCAT1_RES1H(0x0F12)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT1_RES1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT1_RES1L(0x0F13)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT1_RES1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT1_RES1	COS output in SIN/COS mode of computing unit SCAT1; U output in ATAN mode

13.4.39 SCAT1_RES2 (0x0F14, 0x0F15)

SCAT1_RES2H(0x0F14)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT1_RES2[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT1_RES2L(0x0F15)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT1_RES2[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT1_RES2	SIN output in SIN/COS mode of computing unit SCAT1; θ output in ATAN mode

13.4.40 SCAT2_COS (0x0F02, 0x0F03)

SCAT2_COSH(0x0F02)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT2_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT2_COSL(0x0F03)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT2_COS[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT2_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT2

13.4.41 SCAT2_SIN (0x0F04, 0x0F05)

SCAT2_SINH(0x0F04)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT2_SIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT2_SINL(0x0F05)								
Bit	7	6	5	4	3	2	1	0

Name	SCAT2_SIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT2_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT2

13.4.42 SCAT2_THE (0x0F06, 0x0F07)

SCAT2_THEH(0x0F06)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT2_THE[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT2_THEL(0x0F07)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT2_THE[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT2_THE	THE input in SIN/COS mode of computing unit SCAT2

13.4.43 SCAT2_RES1 (0x0F08, 0x0F09)

SCAT2_RES1H(0x0F08)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT2_RES1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT2_RES1L(0x0F09)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT2_RES1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT2_RES1	COS output in SIN/COS mode of computing unit SCAT2; U output in ATAN mode

13.4.44 SCAT2_RES2 (0x0F0A, 0x0F0B)

SCAT2_RES2H(0x0F0A)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT2_RES[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT2_RES2L(0x0F0B)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT2_RES[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT2_RES2	SIN output in SIN/COS mode of computing unit SCAT2; θ output in ATAN mode

13.4.45 SCAT3_COS (0x0EF8, 0x0EF9)

SCAT3_COSH(0x0EF8)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT3_COS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT3_COSL(0x0EF9)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT3_COS[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT3_COS	COS input in SIN/COS or ATAN mode of computing unit SCAT3

13.4.46 SCAT3_SIN (0x0EFA, 0x0EFB)

SCAT3_SINH(0x0EFA)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT3_SIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT3_SINL(0x0EFB)								
Bit	7	6	5	4	3	2	1	0

Name	SCAT3_SIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT3_SIN	SIN input in SIN/COS or ATAN mode of computing unit SCAT3

13.4.47 SCAT3_THE (0x0EFC, 0x0EFD)

SCAT3_THEH(0x0EFC)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT3_THE[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT3_THEL(0x0EFD)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT3_THE[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT3_THE	THE input in SIN/COS mode of computing unit SCAT3

13.4.48 SCAT3_RES1 (0x0EFE, 0x0EFF)

SCAT3_RES1H(0x0EFE)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT3_RES1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT3_RES1L(0x0EFF)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT3_RES1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT3_RES1	COS output in SIN/COS mode of computing unit SCAT3; U output in ATAN mode

13.4.49 SCAT3_RES2 (0x0F00, 0x0F01)

SCAT3_RES2H(0x0F00)								
Bit	15	14	13	12	11	10	9	8
Name	SCAT3_RES[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
SCAT3_RES2L(0x0F01)								
Bit	7	6	5	4	3	2	1	0
Name	SCAT3_RES[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	SCAT3_RES2	SIN output in SIN/COS mode of computing unit SCAT3; θ output in ATAN mode

13.4.50 LPF0_K (0x0FD0, 0x0FD1)

LPF0_KH(0x0FD0)								
Bit	15	14	13	12	11	10	9	8
Name	LPF0_K[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_KL(0x0FD1)								
Bit	7	6	5	4	3	2	1	0
Name	LPF0_K[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF0_K[15:0]	K input of LPF0

13.4.51 LPF0_X (0x0FD2, 0x0FD3)

LPF0_XH(0x0FD2)								
Bit	15	14	13	12	11	10	9	8
Name	LPF0_X[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_XL(0x0FD3)								
Bit	7	6	5	4	3	2	1	0

Name	LPF0_X[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF0_X[15:0]	X input of LPF0

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

LPF0_YHH(0x0FD4)								
Bit	31	30	29	28	27	26	25	24
Name	LPF0_Y[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_YHL(0x0FD5)								
Bit	23	22	21	20	19	18	17	16
Name	LPF0_Y[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_YLH(0x0FD6)								
Bit	15	14	13	12	11	10	9	8
Name	LPF0_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF0_YLL(0x0FD7)								
Bit	7	6	5	4	3	2	1	0
Name	LPF0_Y[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	LPF0_Y[31:0]	Input and output of the register in LPF0 Input: LPF0_Y_{k-1} Output: LPF0_Y_k

13.4.53 LPF1_K (0x0FC8, 0x0FC9)

LPF1_KH(0x0FC8)								
Bit	15	14	13	12	11	10	9	8
Name	LPF1_K[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

LPF1_KL(0x0FC9)								
Bit	7	6	5	4	3	2	1	0
Name	LPF1_K[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF1_K[15:0]	K input of LPF1

13.4.54 LPF1_X (0x0FCA, 0x0FCB)

LPF1_XH(0x0FCA)								
Bit	15	14	13	12	11	10	9	8
Name	LPF1_X[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF1_XL(0x0FCB)								
Bit	7	6	5	4	3	2	1	0
Name	LPF1_X[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF1_X[15:0]	X input of LPF1

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

LPF1_YHH(0x0FCC)								
Bit	31	30	29	28	27	26	25	24
Name	LPF1_Y[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF1_YHL(0x0FCD)								
Bit	23	22	21	20	19	18	17	16
Name	LPF1_Y[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF1_YLH(0x0FCE)								
Bit	15	14	13	12	11	10	9	8
Name	LPF1_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

LPF1_YLL(0x0FCF)								
Bit	7	6	5	4	3	2	1	0
Name	LPF1_Y[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	LPF1_Y[31:0]	Input and output of the register in LPF1 Input: LPF1_Y_{k-1} Output: LPF1_Y_k

13.4.56 LPF2_K (0x0F78, 0x0F79)

LPF2_KH(0x0F78)								
Bit	15	14	13	12	11	10	9	8
Name	LPF2_K[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF2_KL(0x0F79)								
Bit	7	6	5	4	3	2	1	0
Name	LPF2_K[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF2_K[15:0]	K input of LPF2

13.4.57 LPF2_X (0x0F7A, 0x0F7B)

LPF2_XH(0x0F7A)								
Bit	15	14	13	12	11	10	9	8
Name	LPF2_X[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF2_XL(0x0F7B)								
Bit	7	6	5	4	3	2	1	0
Name	LPF2_X[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF2_X[15:0]	X input of LPF2

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

LPF2_YHH(0x0F7C)								
Bit	31	30	29	28	27	26	25	24
Name	LPF2_Y[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF2_YHL(0x0F7D)								
Bit	23	22	21	20	19	18	17	16
Name	LPF2_Y[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF2_YLH(0x0F7E)								
Bit	15	14	13	12	11	10	9	8
Name	LPF2_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF2_YLL(0x0F7F)								
Bit	7	6	5	4	3	2	1	0
Name	LPF2_Y[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	LPF2_Y[31:0]	Input and output of the register in LPF2 Input: LPF2_Y_{k-1} Output: LPF2_Y_k

13.4.59 LPF3_K (0x0F70, 0x0F71)

LPF3_KH(0x0F70)								
Bit	15	14	13	12	11	10	9	8
Name	LPF3_K[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF3_KL(0x0F71)								
Bit	7	6	5	4	3	2	1	0
Name	LPF3_K[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF3_K[15:0]	K input of LPF3

13.4.60 LPF3_X (0x0F72, 0x0F73)

LPF3_XH(0x0F72)								
Bit	15	14	13	12	11	10	9	8
Name	LPF3_X[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF3_XL(0x0F73)								
Bit	7	6	5	4	3	2	1	0
Name	LPF3_X[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	LPF3_K[15:0]	X input of LPF3

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

LPF3_YHH(0x0F74)								
Bit	31	30	29	28	27	26	25	24
Name	LPF3_Y[31:24]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF3_YHL(0x0F75)								
Bit	23	22	21	20	19	18	17	16
Name	LPF3_Y[23:16]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF3_YLH(0x0F76)								
Bit	15	14	13	12	11	10	9	8
Name	LPF3_Y[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
LPF3_YLL(0x0F77)								
Bit	7	6	5	4	3	2	1	0
Name	LPF3_Y[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[31:0]	LPF3_Y[31:0]	Input and output of the register in LPF3 Input: LPF3_Y_{k-1} Output: LPF3_Y_k

13.4.62 PI0_KP (0x0FB8, 0x0FB9)

PI0_KPH(0x0FB8)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_KPL(0x0FB9)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.63 PI0_EK1 (0x0FBA, 0x0FBB)

PI0_EK1H(0x0FBA)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_EK1L(0x0FBB)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.64 PI0_EK (0x0FBC, 0x0FBD)

PI0_EKH(0x0FBC)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_EKL(0x0FBD)								
Bit	7	6	5	4	3	2	1	0

Name	PI0_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.65 PI0_KI (0x0FBE, 0x0FBF)

PI0_KIH(0x0FBE)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_KIL(0x0FBF)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.66 PI0_UKH (0x0FC0, 0x0FC1)

PI0_UKHH(0x0FC0)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKHL(0x0FC1)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.67 PI0_UKL (0x0FC2, 0x0FC3)

PI0_UKLH(0x0FC2)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKLL(0x0FC3)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.68 PI0_UKMAX (0x0FC4, 0x0FC5)

PI0_UKMAXH(0x0FC4)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKMAXL(0x0FC5)								
Bit	7	6	5	4	3	2	1	0
Name	PI0_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.69 PI0_UKMIN (0x0FC6, 0x0FC7)

PI0_UKMINH(0x0FC6)								
Bit	15	14	13	12	11	10	9	8
Name	PI0_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI0_UKMINL(0x0FC7)								
Bit	7	6	5	4	3	2	1	0

Name	PI0_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.70 PI1_KP (0x0FA8, 0x0FA9)

PI1_KPH(0x0FA8)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_KPL(0x0FA9)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.71 PI1_EK1 (0x0FAA, 0x0FAB)

PI1_EK1H(0x0FAA)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_EK1L(0x0FAB)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.72 PI1_EK (0x0FAC, 0x0FAD)

PI1_EKH(0x0FAC)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_EKL(0x0FAD)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.73 PI1_KI (0x0FAE, 0x0FAF)

PI1_KIH(0x0FAE)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_KIL(0x0FAF)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.74 PI1_UKH (0x0FB0, 0x0FB1)

PI1_UKHH(0x0FB0)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKHL(0x0FB1)								
Bit	7	6	5	4	3	2	1	0

Name	PI1_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.75 PI1_UKL (0x0FB2, 0x0FB3)

PI1_UKLH(0x0FB2)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKLL(0x0FB3)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.76 PI1_UKMAX (0x0FB4, 0x0FB5)

PI1_UKMAXH(0x0FB4)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKMAXL(0x0FB5)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.77 PI1_UKMIN (0x0FB6, 0x0FB7)

PI1_UKMINH(0x0FB6)								
Bit	15	14	13	12	11	10	9	8
Name	PI1_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI1_UKMINL(0x0FB7)								
Bit	7	6	5	4	3	2	1	0
Name	PI1_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.78 PI2_KP (0x0F5C, 0x0F5D)

PI2_KPH(0x0F5C)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KPL(0x0F5D)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.79 PI2_EK1 (0x0F5E, 0x0F5F)

PI2_EK1H(0x0F5E)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EK1L(0x0F5F)								
Bit	7	6	5	4	3	2	1	0

Name	PI2_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.80 PI2_EK (0x0F60, 0x0F61)

PI2_EKH(0x0F60)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_EKL(0x0F61)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.81 PI2_KI (0x0F62, 0x0F63)

PI2_KIH(0x0F62)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_KIL(0x0F63)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.82 PI2_UKH (0x0F64, 0x0F65)

PI2_UKHH(0x0F64)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKHL(0x0F65)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.83 PI2_UKL (0x0F66, 0x0F67)

PI2_UKLH(0x0F66)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKLL(0x0F67)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.84 PI2_UKMAX (0x0F68, 0x0F69)

PI2_UKMAXH(0x0F68)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKMAXL(0x0F69)								
Bit	7	6	5	4	3	2	1	0

Name	PI2_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.85 PI2_UKMIN (0x0F6A, 0x0F6B)

PI2_UKMINH(0x0F6A)								
Bit	15	14	13	12	11	10	9	8
Name	PI2_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI2_UKMINL(0x0F6B)								
Bit	7	6	5	4	3	2	1	0
Name	PI2_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.86 PI2_KD (0x0F6C, 0x0F6D)

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

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

13.4.87 PI2_EK2 (0x0F6E, 0x0F6F)

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

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

13.4.88 PI3_KP (0x0F48, 0x0F49)

PI3_KPH(0x0F48)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_KP[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KPL(0x0F49)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_KP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.89 PI3_EK1 (0x0F4A, 0x0F4B)

PI3_EK1H(0x0F4A)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_EK1[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EK1L(0x0F4B)								
Bit	7	6	5	4	3	2	1	0

Name	PI3_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.90 PI3_EK (0x0F4C, 0x0F4D)

PI3_EKH(0x0F4C)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_EK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_EKL(0x0F4D)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_EK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.91 PI3_KI (0x0F4E, 0x0F4F)

PI3_KIH(0x0F4E)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_KI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_KIL(0x0F4F)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_KI[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.92 PI3_UKH (0x0F50, 0x0F51)

PI3_UKHH(0x0F50)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKH[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKHL(0x0F51)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_UKH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.93 PI3_UKL (0x0F52, 0x0F53)

PI3_UKLH(0x0F52)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKL[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKLL(0x0F53)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_UKL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.94 PI3_UKMAX (0x0F54, 0x0F55)

PI3_UKMAXH(0x0F54)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKMAX[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKMAXL(0x0F55)								
Bit	7	6	5	4	3	2	1	0

Name	PI3_UKMAX[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.95 PI3_UKMIN (0x0F56, 0x0F57)

PI3_UKMINH(0x0F56)								
Bit	15	14	13	12	11	10	9	8
Name	PI3_UKMIN[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
PI3_UKMINL(0x0F57)								
Bit	7	6	5	4	3	2	1	0
Name	PI3_UKMIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

13.4.96 PI3_KD (0x0F58, 0x0F59)

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

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

13.4.97 PI3_EK2 (0x0F5A, 0x0F5B)

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

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

14 FOC

14.1 FOC Overview

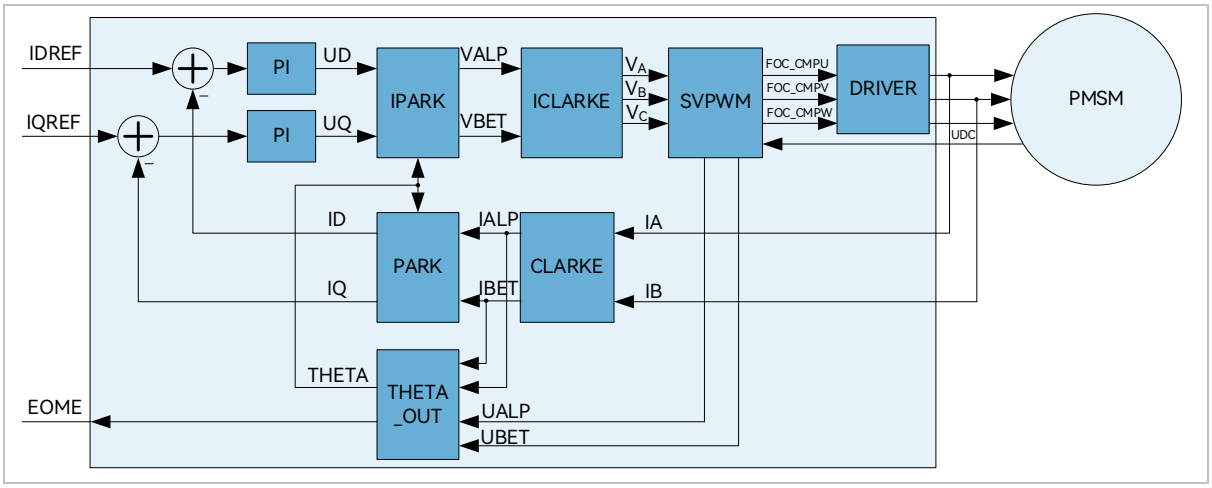
14.1.1 FOC Introduction

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

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

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

Figure 14-1 Block Diagram of FOC Module



14.1.2 Reference Voltage (VREF) Input

The 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 to realize current closed-loop control. FOC module outputs real-time estimated motor speed FOC_ EOME, and MCU can use FOC_EOME as the feedback to build speed loop and send the output of speed loop to FOC_IQREF to implement speed-current dual closed loop control.

14.1.3 PI Controller

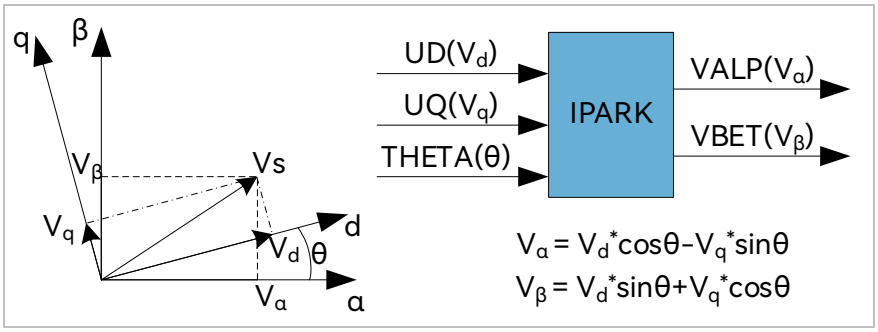
FOC module integrates two PI controllers:

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

14.1.4 Coordinate Transformation

14.1.4.1 Inverse Park Transformation

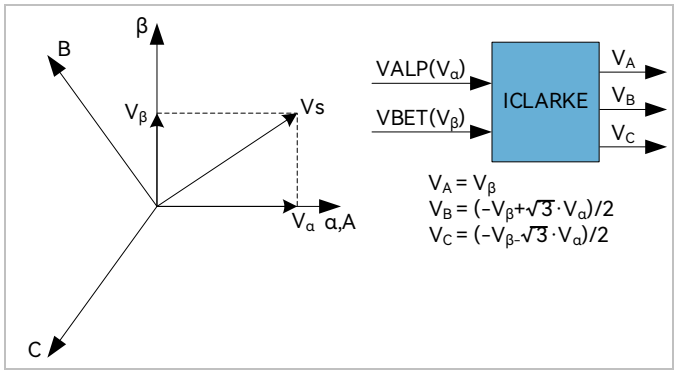
Figure 14-2 Inverse Park Transformation



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

14.1.4.2 Inverse Clarke Transformation

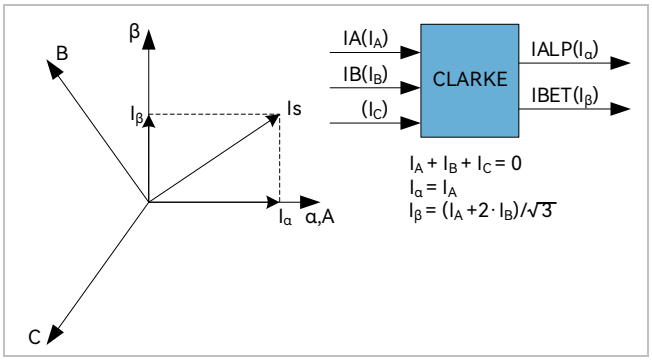
Figure 14-3 Inverse Clarke Transformation



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

14.1.4.3 Clarke Transformation

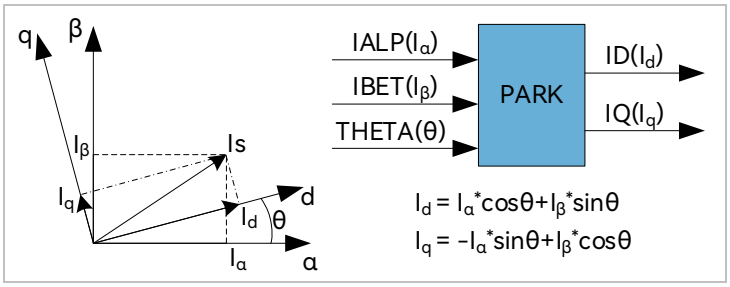
Figure 14-4 Clarke Transformation



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

14.1.4.4 Park Transformation

Figure 14-5 Park Transformation



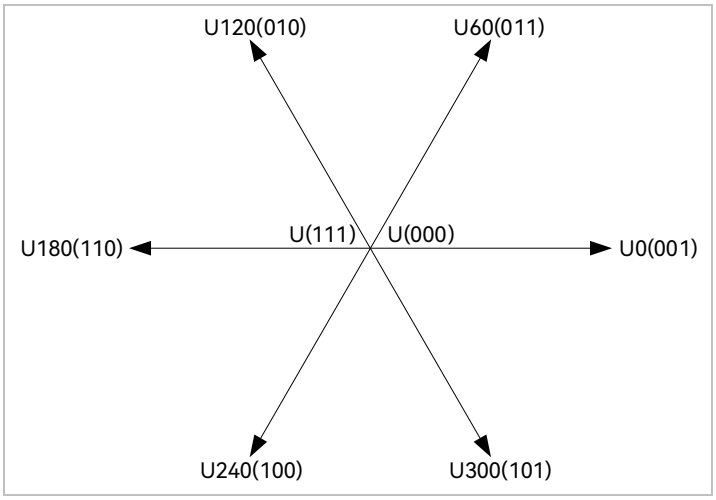
Park transformation is used to transform the current vectors, obtained after Clarke transformation, from α/β-axis coordinate to d/q-axis coordinate to get the sampled d/q-axis current FOC_ID and FOC_IQ.

14.1.5 SVPWM

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

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

Figure 14-6 SVPWM Voltage Vector



SVPWM uses the sum of two adjacent vectors to generate any voltage vector located in the voltage vector space. As shown in Figure 14-7, U_{OUT} is the desired vector and it is in the sector between U60 and U0. Based

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

Figure 14-7 SVPWM Voltage Vector Synthesis

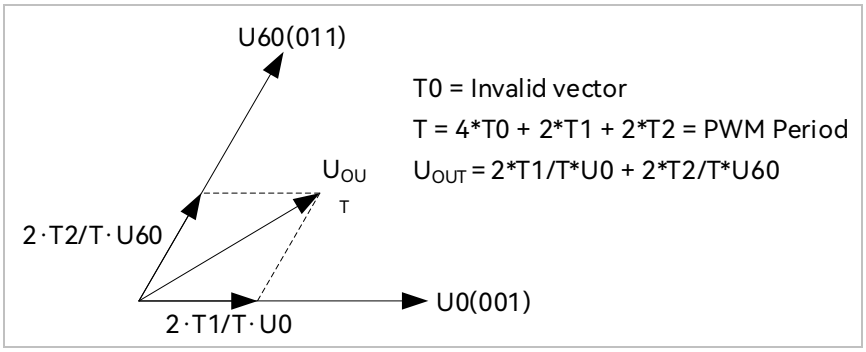


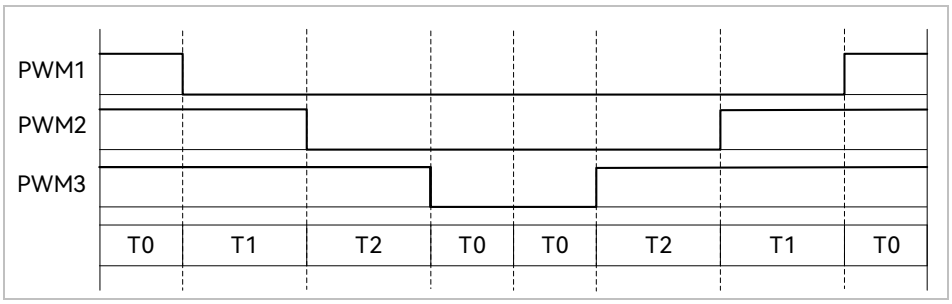
Table 14-1 States of SVPWM Inverter

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

14.1.5.1 Continuous SVPWM

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

Figure 14-8 Output Level of Continuous SVPWM

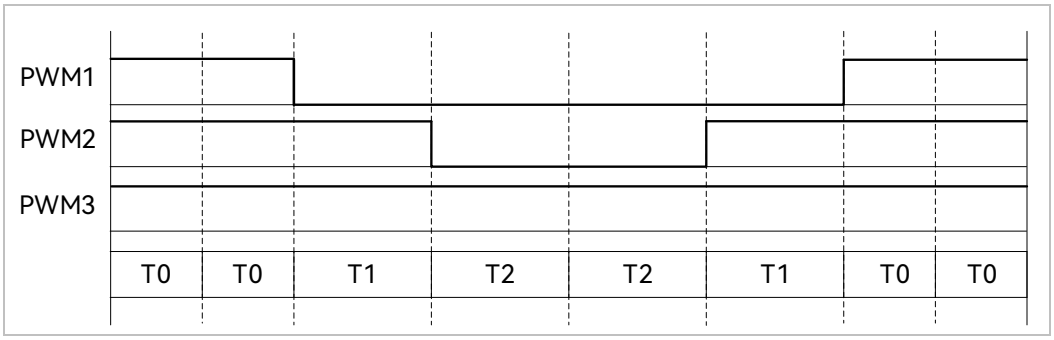


14.1.5.2 Discontinuous SVPWM

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

“1” to activate this mode.

Figure 14-9 Output Level of Discontinuous SVPWM



14.1.6 Overmodulation

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

14.1.7 Deadtime Compensation

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

14.1.8 Current and Voltage Sampling

In FOC mode, bus voltage and phase current are sampled by hardware automatically. Before the FOC module operates, ADC and operational amplifier shall be enabled and the corresponding control registers be configured. No configuration is required for ADC channel and mode. Single/dual/triple/advanced single-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 3 are the default sampling channels of ia, ib and C-phase current (ic) respectively. Channel 2 can be selected for bus voltage sampling.

14.1.8.1 Single-shunt Current Sampling Mode

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

samples itrip (channel 4) twice during DRV timer counting-up operation, and samples bus voltage during DRV timer counting-down operation and after FOC module completes the calculation.

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, FOC_TRGDLY = 5, the sampling time is delayed by 5*T = 208ns; FOC_TRGDLY = 0xFB(−5), the sampling time is advanced by 5*T = 208ns.

Figure 14-10 Single-shunt Current Sampling Timing

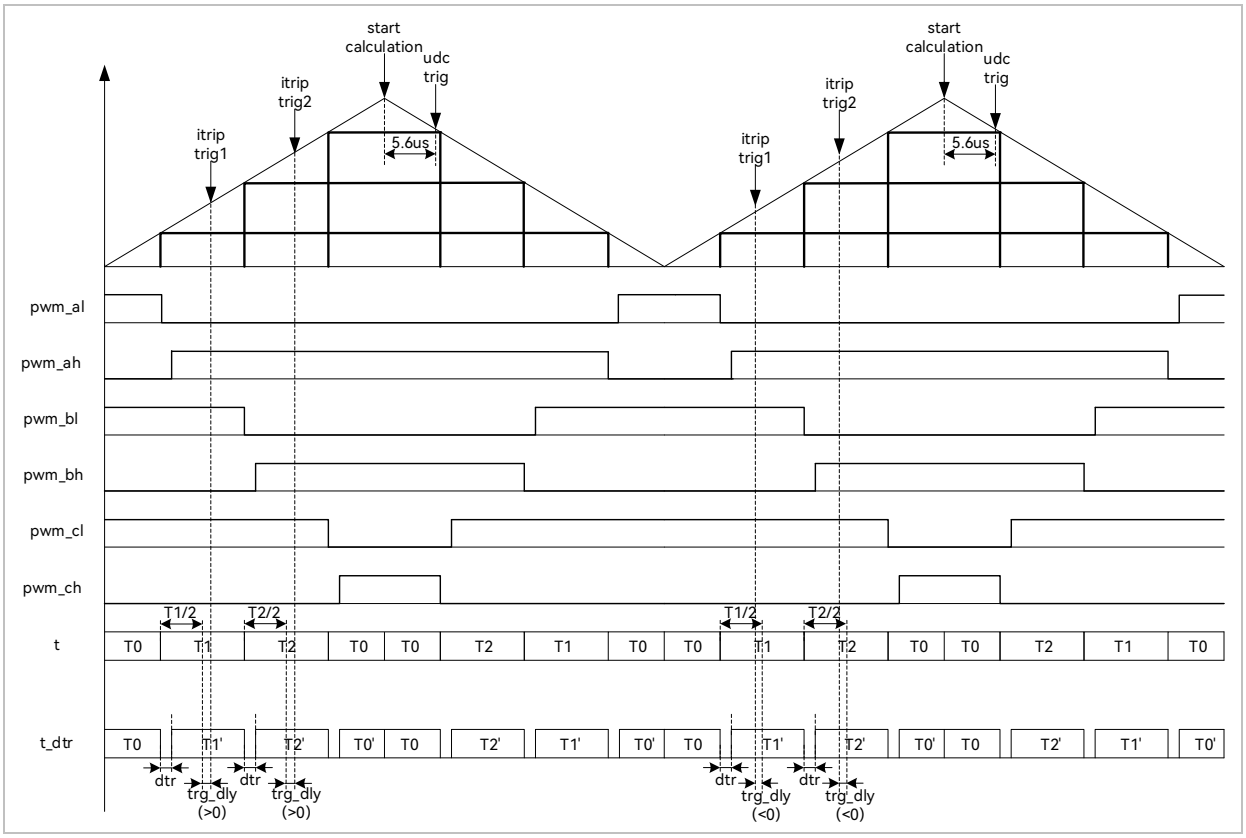
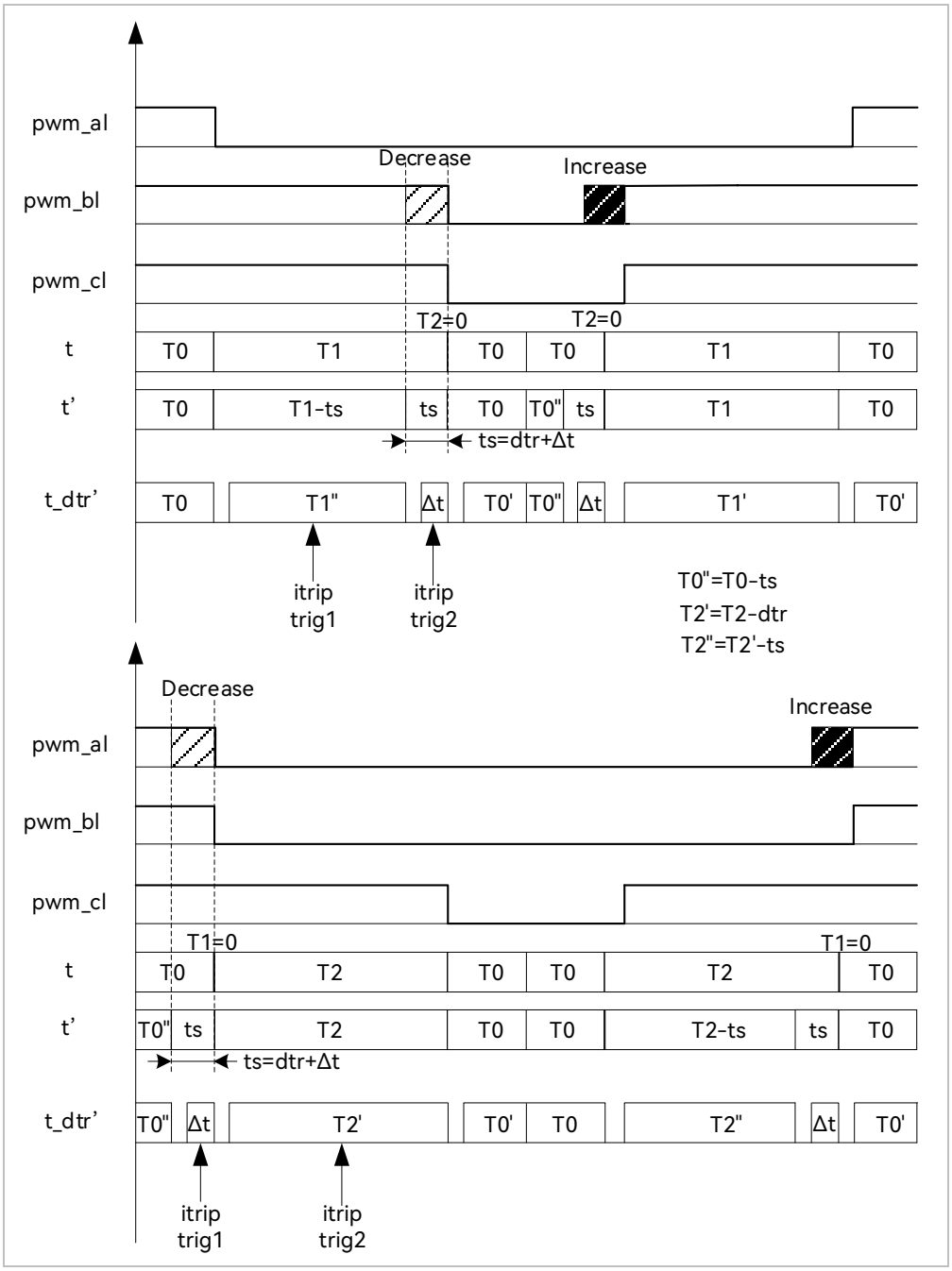


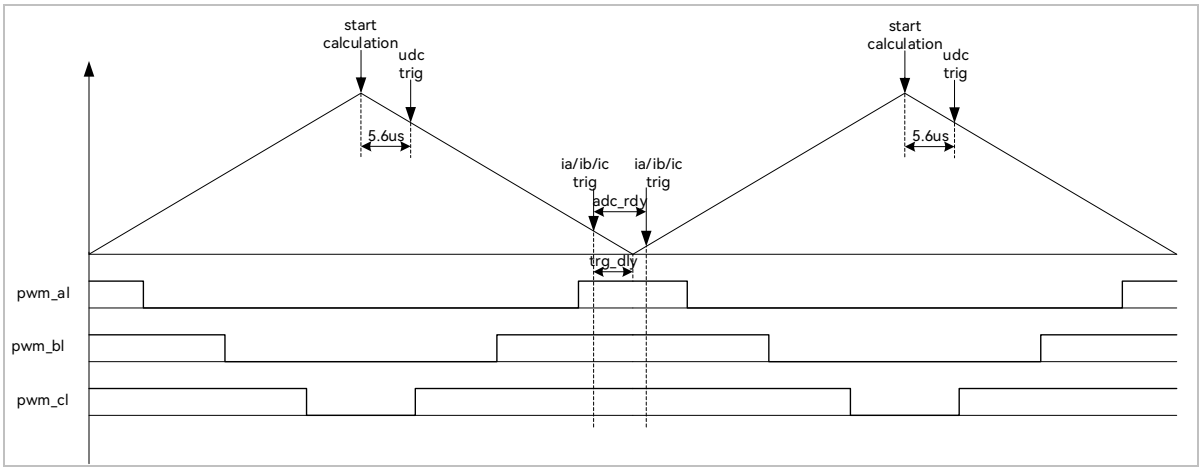
Figure 14-11 Time Compensation for Single-shunt Current Sampling



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

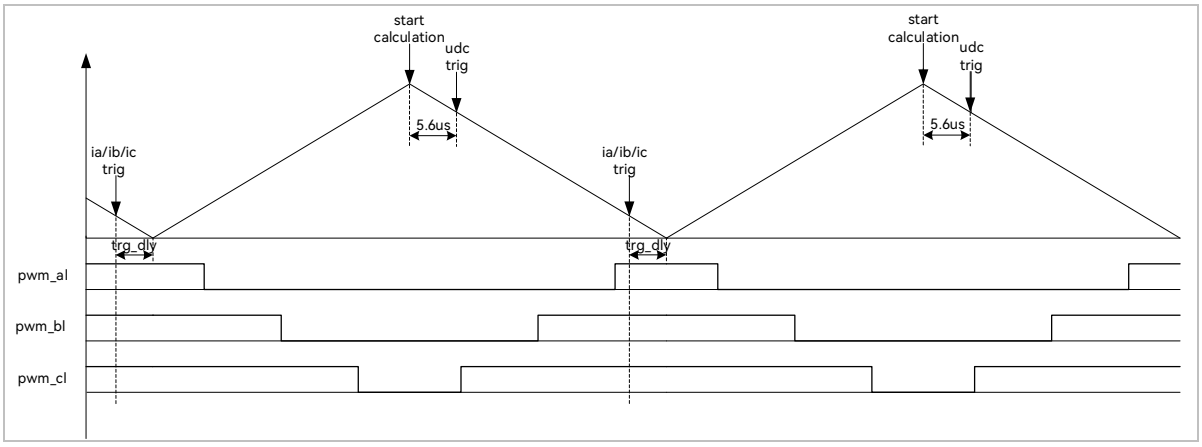
14.1.8.2 Dual/Triple-shunt Current Sampling Mode

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



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

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



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

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

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

14.1.8.3 Current Sampling Offset

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

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

14.1.9 Angle Mode

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

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

- Forced angle of estimator

Table 14-2 Sources of Angle

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

14.1.9.1 Forced Ramping Angle

Forced ramping angle is controlled by angle register FOC_THETA, speed register FOC_RTHESTEP, acceleration register FOC_RTHEACC and ramping counter FOC_RTHECNT. The formula is:

FOC_RTHESTEP (32 bits) = FOC_RTHESTEP (32 bits) + FOC_RTHEACC (16 low-order bits)

THETA_OL (16 bits) = THETA_OL (16 bits) + FOC_RTHESTEP (16 high-order bits)

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

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

14.1.9.2 Forced Pulling Angle

Forced pulling angle is controlled by angle FOC_THETA and speed FOC_RTHESTEP. The formula is:

THETA_OL (16 bits) = THETA_OL (16 bits) + FOC_RTHESTEP (16 high-order bits)

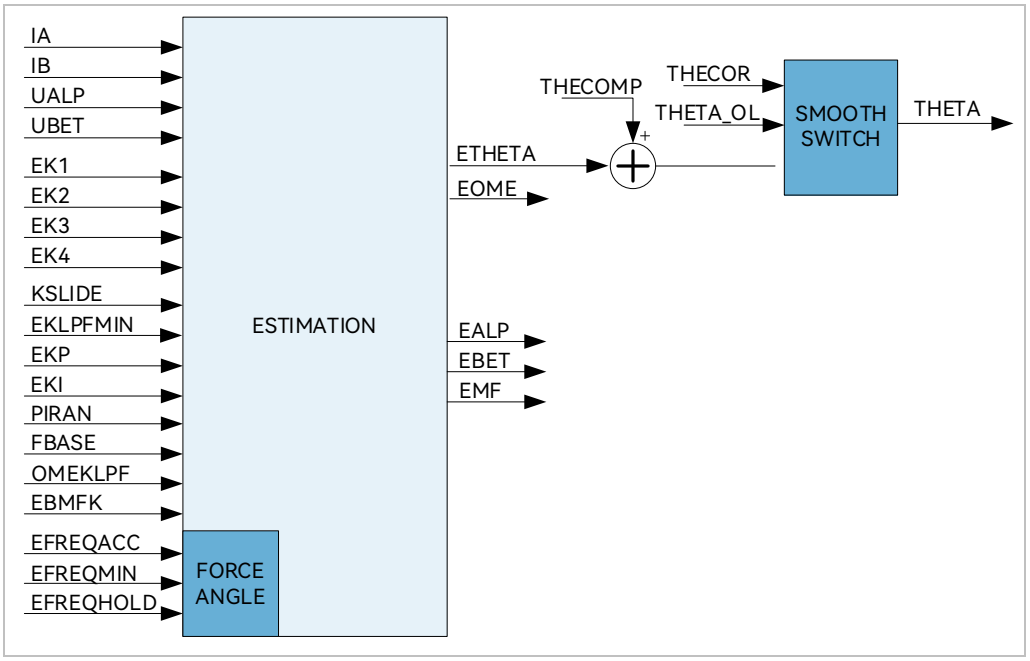
Where, THETA_OL is an internal variable of the chip. In forced pulling angle mode, THETA_OL is written to FOC_THETA as the used angle. If the software writes a value to FOC_THETA, this value is written to

THETA_OL as well.

- When FOC_CR1[RFAE] is set to “1” and FOC_CR1[ANGM] to “0”, MCU switches to forced pulling angle mode 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 to “0” enables the pre-position feature. The sensor-based FOC is implemented after the motor speed is set with FOC_RTHESTEP. (Principle of sensor-based FOC: The angle and speed are written to FOC_THETA and FOC_RTHESTEP by software, and FOC module generates an angle in each PWM cycle based on the written values.)

14.1.9.3 Estimator Output Angle

Figure 14-14 Schematic Block Diagram of Estimator

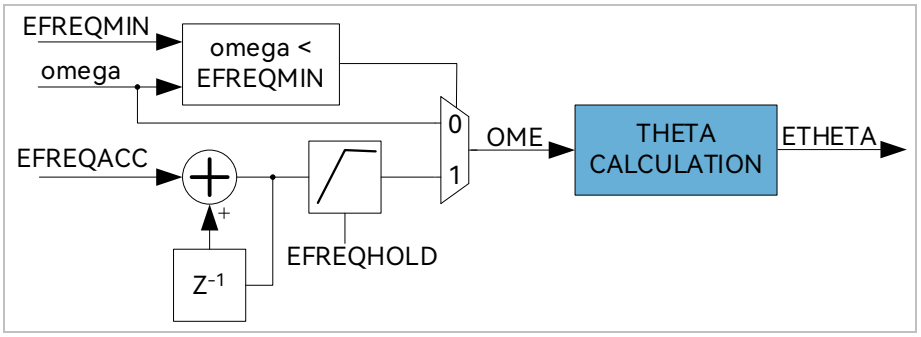


14.1.9.3.1 Estimated Angle of Estimator

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

14.1.9.3.2 Forced Angle of Estimator

Figure 14-15 Schematic Diagram of Forced Angle of Estimator



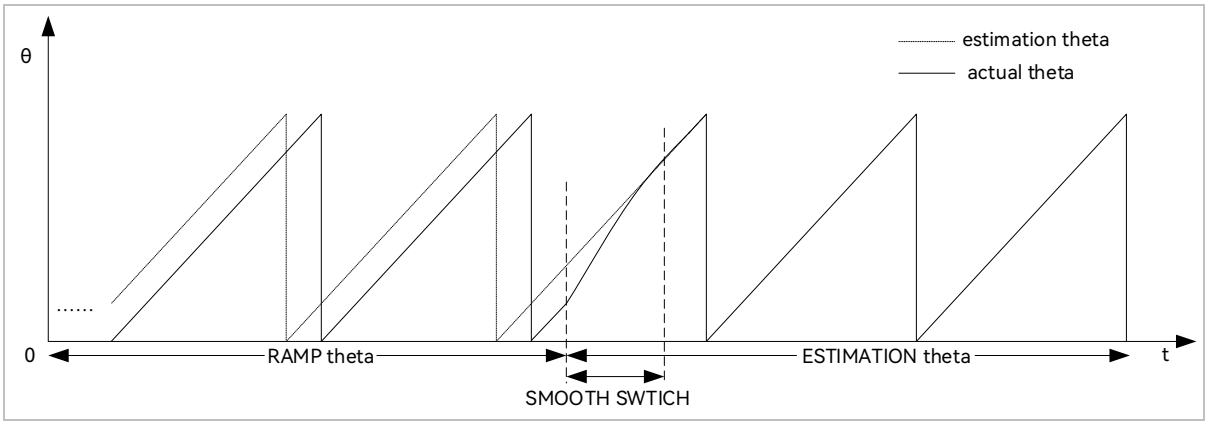
This feature is similar to the ramping feature. Due to the low speed at motor starting process, there may be a deviation in angle and speed estimation with small effective signals, resulting in startup failure. In this case, the estimator outputs the forced angle to ensure the motor start normally.

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

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

14.1.9.3.3 Angle Smooth Switching

Figure 14-16 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 in every PWM cycle until it is close to FOC_ETHETA. After the deviation is less than FOC_THECOR, FOC_ETHETA is selected as the output angle.

14.1.9.3.4 Angle Compensation

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

14.1.10 Motor Real-time Parameters

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

- Used angle FOC_THETA
- Estimated angle FOC_ETHETA and estimated speed FOC_EOME
- d-axis voltage FOC_UD and q-axis voltage FOC_UQ
- d-axis current FOC_ID and q-axis current FOC_IQ
- α-axis voltage FOC_VALP and β-axis voltage FOC_VBET
- Bus voltage FOC_UDCFLT
- Phase current FOC_IA, FOC_IB, FOC_IC and maximum phase current FOC_IAMAX, FOC_IBMAX, FOC_ICMAX
- α-axis current (equal to FOC_IA) and β-axis current FOC_IBET
- α-axis BEMF FOC_EALP and β-axis BEMF FOC_EBET
- Magnitude of BEMF FOC_EMF
- Motor power FOC_POW

14.1.10.1 Tailwind/headwind Detection

FOC module provides tailwind/headwind detection feature. FOC module starts to operate when FOC_CR0 [ESCMS] is set to “1”, FOC_ IDREF to “0” and FOC_ IQREF to “0”. Motor’s rotor state is detected by FOC_ETHETA and FOC_EOME. If FOC_ETHETA decreases or FOC_EOME is a negative value, the motor rotates in the headwind state and it is necessary to brake first and then start the motor with ramping forced angle mode. If FOC_ETHETA increases or FOC_EOME is a positive value, the motor rotates in the tailwind state and can be started using estimated angle directly.

14.1.10.2 BEMF Detection

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

14.1.10.3 Motor Power

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

14.1.11 FG Output 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 using the following algorithm: FOC_KFG = SYSCLK/(2^TIM4_CR0[T4PSC]*fbase*x), where, x refers to the expected number of FG signal in one electric cycle. If the result exceeds 65535, the clock division factor TIM4_CR0[T4PSC] shall be adjusted.

When FOC_KFG = 0, this feature is disabled, and TIM4_ARR and TIM4_DR keeps unchanged.

14.2 FOC Registers

14.2.1 FOC_CR0 (0x409F)

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

Bit	Name	Description
[7]	OMIF	omega < FOC_ EFREQMIN Flag. This bit is valid even if FOC_ CR1 [EFAE] is 0. 0: omega ≥ FOC_EFREQMIN 1: omega < FOC_EFREQMIN
[6]	OMAF	omega > FOC_EFREQMIN Flag 0: omega ≤ FOC_EFREQMIN 1: omega > FOC_EFREQMIN
[5:4]	MERRS	Maximum Error of SMO Algorithm Selection 00: 0.5 01: 0.25 10: 0.125 11: 1.0
[3]	UCSEL	Sampling Channel for Bus Voltage (UDC) In FOC mode, bus voltage is sampled automatically by hardware after Driver counter is enabled. The FOC_CR0[UCSEL] bit selects the channel for sampling. 0: ADC channel 2 1: ADC channel 15 ADC channel 15 is an internal channel dedicated for bus voltage sampling. The voltage division ratio is selected by configuring ADC_CR[ADCRATIO]. ADC channel 2 is the external bus voltage sampling channel. Note: It is not necessary to set the associated Enable Bit in ADC_MASK register to “1”.
[2]	OMAS	Output selection when omega is too large When omega[15:8] > FOC_EFREQMAX, the output OME is set as: 0: FOC_EFREQMAX*256 1: FOC_EFREQHOLD
[1]	ESCMS	Angle Mode Selection 0: Internal Test Mode 1: Recommended Mode
[0]	EDIS	FOC__EALP/FOC__EBET Auto-computation Disable 0: Disable 1: Enable

14.2.2 FOC_CR1 (0x40A0)

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

Bit	Name	Description
[7]	OVMDL	Overmodulation Enable 0: Disable 1: Enable
[6]	EFAE	Forced Angle of Estimator Enable When this feature is enabled, angle mode is determined by the estimator, and it switches to estimated angle mode automatically. 0: Disable 1: Enable
[5]	RFAE	Forced Ramping Angle Enable When this feature is enabled, angle mode is determined by the ramping module. After ramping, it switches to estimated mode or forced pulling mode according to FOC_CR1[ANGM]. FOC_CR1[RFAE] is cleared to “0” by hardware as well. 0: Disable 1: Enable
[4]	ANGM	Angle Mode When FOC_CR1[RFAE] = 0, angle mode is determined by this bit. When FOC_CR1[RFAE] = 1, angle mode is determined by this bit after ramping. 0: Forced Pulling Angle Mode 1: Estimated Angle of Estimator Mode
[3:2]	CSM	Current Sampling Mode 00: Single-shunt Current Sampling 01: Dual-shunt Current Sampling 10: Advanced Single-shunt Current Sampling 11: Triple-shunt Current Sampling Note: The Advanced Single-shunt Current Sampling Mode works better in low duty cycles than high duty cycles. You can switch between Single-shunt Current Sampling Mode and Advanced Single-shunt Current Sampling Mode according to the duty cycle, and set FOC_TSMIN accordingly
[1]	RSV	Reserved
[0]	SVPWMEN	SVPWM Module Enable 0: Disable 1: Enable

14.2.3 FOC_CR2 (0x40A1)

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

Bit	Name	Description
[7]	ESEL	Angle Estimator Mode Selection 0: SMO 1: PLL (phase-locked loop). FOC_KSILDE register works as FOC_PLLKP of PI controller, and FOC_KLPFMIN register as FOC_PLLKI of PI controller
[6]	ICLR	Clear FOC_IAMAX/FOC_IBMAX/FOC_ICMAX to “0” 0: No effect 1: This bit is automatically set to “0” after FOC_IAMAX/FOC_IBMAX/ FOC_ICMAX are cleared to “0”
[5]	F5SEG	SVPWM Mode Selection 0: Continuous SVPWM 1: Discontinuous SVPWM (it cannot be selected in single-shunt current sampling mode)
[4]	DSS	Dual/Triple-shunt Current Sampling Mode 0: Sequential Sampling Mode, where two-phase currents are sampled in each carrier period 1: Alternate Sampling Mode. FOC module completes the calculation in every carrier cycle. Single-phase current is sampled in each carrier cycle, and two-phase currents are sampled alternately in two adjacent carrier cycles.
[3:2]	CSOC	Current Sampling Offset Calibration This bit is written to select the offset of FOC_CSO. In single-shunt sampling, “00” or “11” is written to calibrate itrip offset. In dual-shunt sampling, “01” is written to calibrate ia offset and “10” to calibrate ib offset. In triple-shunt sampling, “01” is written to calibrate ia offset, “10” to calibrate ib offset and “00” or “11” to calibrate ic offset. 00: itrip & ic 01: ia 10: ib 11: itrip & ic
[1]	UQD	q-axis PI Controller Disable. When it is enabled, FOC_UQ value is no longer updated by the PI controller. 0: Disable 1: Enable
[0]	UDD	d-axis PI Controller Disable. When it is enabled, FOC_UD value is no longer updated by the PI controller. 0: Disable 1: Enable

14.2.4 FOC_CR3 (0x409E)

Bit	7	6	5	4	3	2	1	0
Name	EFAM	TAMD	MFP_EN	FOC_THECOMP_DIS	FOCFEN	HALL_PLL_EN	TSMINH9	TSMINH8
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	EFAM	OMEGA Startup Force Enable When FOC_CR1[EFAE] is set to “0” and FOC_CR1[EFAM] to “1”, FOC_OMEGA register is forced to stay as FOC_EFREQHOLD. 0: Disable 1: Enable
[6]	TAMD	Angle Calculation Method The angle derived from atan (ealpha/ebeta) is used as FOC_THETA. 0: Disable 1: Enable
[5]	MFP_EN	Adaptive Observer Enable 0: Disable 1: Enable
[4]	FOC_THECOMP_DIS	Algorithm w/o Compensation Angle Enable When it is enabled, angle compensation of 26.5° is not executed even if the SMO or AO algorithm is selected. 0: Disable 1: Enable
[3]	FOCFEN	FOC Force Enable When DRV_CR[MESEL] is set to “1”, FOC module performs calculation even if DRV_CR[OCS] = 0. 0: Disable 1: Enable
[2]	HALL_PLL_EN	Hall Filter in PLL Mode Enable In Hall mode, Hall angle written to FOC_THETA is sent to other modules after smooth switching. 0: Disable 1: Enable
[1:0]	TSMINH	Scale up by two bits of FOC_TSMIN, forming 10-bit data with the 0x40a2 register

14.2.5 FOC_CR4 (0x409D)

Bit	7	6	5	4	3	2	1	0
Name	HFI_ST	HFI_DEL_MAN_EN	HFI_SMO_EN	HFI_THE_SEL	HFI_MAN_EN	HFI_OME_EN	HFI_TOG_EN	HFI_EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	HFI_ST	High frequency injection state, which determines FOC_UD and FOC_UDCPS are added or subtracted. 0: Positive state, and FOC_UD is added with FOC_UDCPS 1: Negative state, and FOC_UD is subtracted from FOC_UDCPS
[6]	HFI_DEL_MAN_EN	Dedicated for internal functional testing under high frequency injection
[5]	HFI_SMO_EN	Estimated Loop Angle Compensation Enable 0: Disable 1: Enable
[4]	HFI_THE_SEL	Dedicated for internal functional testing under high frequency injection
[3]	HFI_MAN_EN	Dedicated for internal functional testing under high frequency injection
[2]	HFI_OME_EN	Angle Calculation by High Frequency Injection-based Speed Enable 0: Disable. The angle calculation is performed with the estimated speed 1: Enable. The angle calculation is performed with the high frequency injection-based speed
[1]	HFI_TOG_EN	FOC_CR4[HFI_ST] Automatic Reverse Enable 0: Disable 1: Enable
[0]	HFI_EN	High Frequency Injection Enable 0: Disable 1: Enable

14.2.6 FOC_CR5 (0x409C)

Bit	7	6	5	4	3	2	1	0
Name	SAM_MD	HFI_FDB_SEL	DT_X2_EN	VAB_CPS_EN	SMO_REV_EN	HFI_MD	OME_ABS_EN	CAL2_EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	SAM_MD	Dedicated for internal functional testing in triple-shunt current sampling mode
[6]	HFI_FDB_SEL	Dedicated for internal functional testing under high frequency injection
[5]	DT_X2_EN	Dedicated for internal functional testing under high frequency injection
[4]	VAB_CPS_EN	Dedicated for internal functional testing under high frequency injection

[3]	SMO_REV_EN	Estimated Loop Angle Compensation Enable in Reverse Stage. When the software detects that the motor rotates reversely, it enables this bit to provide a proper angle compensation (FOC_CR4[HFI_SMO_EN] is enabled). 0: Disable 1: Enable
[2]	HFI_MD	Dedicated for internal functional testing under high frequency injection
[1]	OME_ABS_EN	Estimated Angle Enable in Reverse Stage. It shall be enabled if it is required to use high frequency injection in reverse state. 0: Disable 1: Enable
[0]	CAL2_EN	Dedicated for internal functional testing under high frequency injection

14.2.7 FOC_TSMIN (0x40A2)

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

Bit	Name	Description
[7:0]	FOC_TSMIN	Single-shunt Current Sampling Mode: minimum window for sampling Dual/triple-shunt Current Sampling Mode: deadtime compensation Range [0, 255] TSMIN = sampling window T_{window} + deadtime T_{DT} Example: Assuming that T_{window} = 1μs, T_{DT} = 1μs, TSMIN = 2μs and carrier period = 62.5μs, then FOC_TSMIN = (1 + 1)/62.5*4096 = 131. Advanced Single-shunt Current Sampling Mode: minimum window for sampling TSMIN = sampling window T_{window} + deadtime T_{DT} Example: Assuming that T_{window} = 1μs, T_{DT} = 1μs and TSMIN = 2μs, then FOC_TSMIN = (1 + 1)*24 = 48.

14.2.8 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 minimum pulse required by high side of the pre-driver must be longer than a certain time. The pre-driver is not turned on if the pulse is less than the value set by this bit.

		Range [0, 255] Example: Assuming that it is required to remove narrow pulses with less than 1μs width, deadtime T_{DT} = 1μs, and carrier period = 62.5μs, then FOC_TGLI = (1 + 1)/62.5*4096 =131.

14.2.9 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 the low side is turned on for less than FOC_TBLO, the phase current is not sampled and obtained through special process. Range [0, 255] Example: Assuming that the phase current is not sampled if the low side is turned on for less than 1μs, then FOC_TBLO = 1000ns/41.67ns = 24.

14.2.10 FOC_TRGDLY (0x40A5)

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

Bit	Name	Description
[7:0]	FOC_TRGDLY	Time Configuration for Current Sampling When FOC_TRGDLY is set to “0”, FOC module samples the current as follows. Single-shunt Current Sampling Mode: Midpoint between deadtime and applied time of active voltage vector Dual/Triple-shunt/Advanced Single-shunt Current Sampling Mode: Midpoint of vector 000 (Driver count value = 0) Range [-128, 127] Single-shunt Current Sampling Mode: If FOC_TRGDLY = 5, it delays by 5*T = 208ns to sample the current, and if FOC_TRGDLY = 0xFB (complement) or FOC_TRGDLY = -5, it advances by 5*T=208ns. Dual-shunt/Triple-shunt/Advanced Single-shunt Current Sampling Mode: If FOC_TRGDLY = 0x85 (the highest bit, and the remaining 7 bits are absolute values) and Driver timer counts down, it samples the current at 5*T = 208ns before an overflow event occurs. If OC_TRGDLY = 5 and Driver timer counts up, it samples the current at 5*T = 208ns after an overflow event occurs.

14.2.11 FOC_CSO (0x40A6, 0x40A7)

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

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

14.2.12 FOC_RTHESTEP (0x40A8, 0x40A9)

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

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

14.2.13 FOC_RTHEACC (0x40AA, 0x40AB)

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

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

14.2.14 FOC_EOMELPF (0x40AA, 0x40AB)

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

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

14.2.15 FOC_RTHECNT (0x40AC)

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

Bit	Name	Description
[7:0]	FOC_RTHECNT	Max. ramping counts = FOC_RTHECNT*256 When ramping feature is enabled, the ramping calculation is performed in each carrier cycle. After FOC_RTHECNT*256 times, the ramping feature is disabled.

14.2.16 FOC_THECOR (0x40AD)

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

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

14.2.17 FOC_EMF (0x40AE, 0x40AF)

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

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

14.2.18 FOC_THECOMP (0x40AE, 0x40AF)

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

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

14.2.19 FOC_DMAX (0x40B0, 0x40B1)

FOC_DMAXH(0x40B0)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_DMAX[15:8]							
Type	W	W	W	W	W	W	W	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	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_DMAX	Max. output of d-axis PI controller Range [-32768, 32767]

14.2.20 FOC_OMEEST (0x40B0, 0x40B1)

FOC_OMEESTH(0x40B0)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_OMEEST[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

FOC__OMEESTL(0x40B1)								
Bit	7	6	5	4	3	2	1	0
Name	FOC__OMEEST[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC__OMEEST	FOC Calculated Speed of Estimator Range [0, 32767]

14.2.21 FOC_DMIN (0x40B2, 0x40B3)

FOC_DMINH(0x40B2)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_DMIN[15:8]							
Type	W	W	W	W	W	W	W	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	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_DMIN	Min. output of d-axis PI controller Range [-32768, 32767]

14.2.22 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 of q-axis PI controller Range [-32768, 32767]

14.2.23 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 of q-axis PI controller Range [-32768, 32767]

14.2.24 FOC_UD (0x40B8, 0x40B9)

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

Bit	Name	Description
[15:0]	FOC_UD	d-axis voltage calculated by d-axis PI controller Range [-32768, 32767]

14.2.25 FOC_UQ (0x40BA, 0x40BB)

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

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

Bit	Name	Description
[15:0]	FOC_UQ	q-axis voltage calculated by q-axis PI controller Range [-32768, 32767]

14.2.26 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	Filtered d-axis current Range [-32768, 32767]

14.2.27 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	Filtered q-axis current Range [-32768, 32767]

14.2.28 FOC_IBET (0x40C0, 0x40C1)

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

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

14.2.29 FOC_IQ_LPFK (0x40C0)

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

Bit	Name	Description
[7:0]	FOC_IQ_LPFK	LPF coefficient of FOC_IQ, set to 0xFF by default Range [0, 255]

14.2.30 FOC_ID_LPFK (0x40C1)

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

Bit	Name	Description
[7:0]	FOC_ID_LPFK	LPF coefficient of FOC_ID, set to 0xFF by default Range [0, 255]

14.2.31 FOC_VBET (0x40C2, 0x40C3)

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

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

14.2.32 FOC_UDCPS (0x40C2, 0x40C3)

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

Bit	Name	Description
[15:0]	FOC_UDCPS	d-axis Voltage Compensation The result of d-axis PI controller (FOC_UD) added to FOC_UDCPS is transferred to the next module. Range [-32768, 32767]

14.2.33 FOC_UQCPS (0x40C4, 0x40C5)

FOC_UQCPSH(0x40C4)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_UQCPS[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_UQCPSL(0x40C5)								

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

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

14.2.34 FOC_VALP (0x40C4, 0x40C5)

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

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

14.2.35 FOC_FLUX (0x40C6, 0x40C7)

FOC_FLUXH(0x40C6)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_FLUX[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_FLUXL(0x40C7)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_FLUX[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_FLUX	Magnetic Flux of Motor Range [0, 32767]

14.2.36 FOC_IC (0x40C6, 0x40C7)

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

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

14.2.37 FOC_LQ (0x40C8, 0x40C9)

FOC_LQH(0x40C8)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_LQ[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_LQ(0x40C9)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_LQ[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_LQ	Q-axis Inductance Range [0, 32767]

14.2.38 FOC_IB (0x40C8, 0x40C9)

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

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

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

14.2.39 FOC_LD (0x40CA, 0x40CB)

FOC_LDH(0x40CA)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_LD[15:8]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
FOC_LD(0x40CB)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_LD[7:0]							
Type	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	FOC_LD	D-axis Inductance Range [0, 32767]

14.2.40 FOC_IA (0x40CA, 0x40CB)

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

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

14.2.41 FOC_THETA (0x40CC, 0x40CD)

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

Bit	Name	Description
[15:0]	FOC_THETA	Output Angle of FOC Module Range [-32768, 32767] The bit value -32768 ~ 32767 corresponds to angle range -180°~ 180° Example: Assuming that FOC_THETA = 8192, the output angle is 8192/32768*180°= 45°.

14.2.42 FOC_ETHETA (0x40CE, 0x40CF)

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

Bit	Name	Description
[15:0]	FOC_ETHETA	Read: Output Angle of Estimator (angle before FOC_THECOMP is applied); the format is same as that of FOC_THETA. Write: Start Angle of Estimator Range [-32768, 32767]

14.2.43 FOC_EALP (0x40D0, 0x40D1)

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

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

14.2.44 FOC_EBET (0x40D2, 0x40D3)

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

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

14.2.45 FOC_EOME (0x40D4, 0x40D5)

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

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

Bit	Name	Description
[15:0]	FOC_EOME	Output Speed of Estimator Range [-32768, 32767]

14.2.46 FOC_UQEX (0x40D6, 0x40D7)

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

Bit	Name	Description
[15:0]	FOC_UQEX	Overflow Value of q-axis PI Controller 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, 32767]

14.2.47 FOC_KFG (0x40D6, 0x40D7)

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

Bit	Name	Description

[15:0]	FOC_KFG	Coefficient of FG Calculation FOC module performs the calculation based on FOC_EOMELPF and FOC_KFG in each PWM cycle. The result is updated to TIM4_ARR and half of the result (TIM4_ARR/2) to TIM4_DR by hardware. See FG Output Generation for more details. Range [0, 65535] Note: The clock division factor TIM4_CR0[T4PSC] of Timer4 shall be adjusted if FOC_KFG overflows. When FOC_ KFG = 0, this feature is disabled.

14.2.48 FOC_POW (0x40D8, 0x40D9)

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

Bit	Name	Description
[15:0]	FOC_POW	Motor Power Range [-32768, 32767]

14.2.49 FOC_EOMEKLPF (0x40D8)

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

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

14.2.50 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 Recorded maximum value of phase-A current. This value may be unreliable unless the motor rotates in a full electrical cycle. This maximum value will not be cleared to “0” automatically unless FOC_CR2[ICLR] is set to “1”. Range [-32768, 32767]

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

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

14.2.53 FOC_EFREQMAX (0x406F)

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

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

14.2.54 FOC_DKP (0x4070, 0x4071)

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

Bit	Name	Description
[15:0]	FOC_DKP	KP of d-axis PI Controller Range [0, 32767]; MSB is always 0; Q12 format

14.2.55 FOC_EKP (0x4074, 0x4075)

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

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

14.2.56 FOC_EKI (0x4076, 0x4077)

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

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

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

14.2.57 FOC_KSLIDE (0x4078, 0x4079)

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

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

14.2.58 FOC_EKLPFMIN (0x407A, 0x407B)

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

Bit	Name	Description
[15:0]	FOC_EKLPFMIN/ FOC_PLLKI	FOC_CR2[ESEL] = 0: The minimum value of low-pass filtering coefficient for SMO BEMF. The low-pass filtering coefficient is forced to be this value when

		it is lower than this value. Q15 format. FOC_CR2[ESEL] = 1: KI coefficient of PI controller on PLL. Q15 format. Range [0, 32767], MSB is always 0.

14.2.59 FOC_DKI (0x407C, 0x407D)

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

Bit	Name	Description
[15:0]	FOC_DKI	KI of d-axis PI Controller Range [0, 32767]. MSB is always 0. Q15 format

14.2.60 FOC_OMEKLPF (0x407E, 0x407F)

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

Bit	Name	Description
[15:0]	FOC_OMEKLPF	LPF coefficient of estimated speed of the estimator MSB is always 0. Q15 format. Range [0, 32767]

14.2.61 FOC_FBASE (0x4080, 0x4081)

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

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

14.2.62 FOC_EFREQACC (0x4082, 0x4083)

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

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

14.2.63 FOC_EFREQMIN (0x4084, 0x4085)

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

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

14.2.64 FOC_EFREQHOLD (0x4086, 0x4087)

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

Bit	Name	Description
[15:0]	FOC_EFREQHOLD	Maximum Value of Forced Speed of the Estimator FOC_EFREQHOLD is an internal 24-bit variable, and MSB is sign bit. 16 high-order bits are written by the software. Range [-32768, 32767]

		Example: Assuming that fbase = 200Hz and pp (Pole_Pairs) = 4, then speed_base = 60*fbase/pp = 3000rpm. If max. forced speed = 60rpm, then FOC_EFREQHOLD = 60rpm/speed_base*32768 = 655(0x028F).

14.2.65 FOC_EK3 (0x4088, 0x4089)

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

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

14.2.66 FOC_EK4 (0x408A, 0x408B)

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

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

14.2.67 FOC_EK1 (0x408C, 0x408D)

FOC_EK1H(0x408C)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_EK1[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_EK1L(0x408D)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_EK1[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

14.2.68 FOC_EK2 (0x408E, 0x408F)

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

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

14.2.69 FOC_IDREF (0x4090, 0x4091)

FOC_IDREFH(0x4090)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_IDREF[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
FOC_IDREFL(0x4091)								

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

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

14.2.70 FOC_IQREF (0x4092, 0x4093)

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

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

14.2.71 FOC_QKP (0x4094, 0x4095)

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

Bit	Name	Description
[15:0]	FOC_QKP	KP coefficient of q-axis PI controller. MSB is always 0. Q12 format. Range [0, 32767] corresponds to range of Q12 [0, 8]

14.2.72 FOC_QKI (0x4096, 0x4097)

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

Bit	Name	Description
[15:0]	FOC_QKI	KI coefficient of q-axis PI controller. MSB is always 0. Q15 format. . The bit value range [0, 32767] corresponds to Q15 value range [0, 1]

14.2.73 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 bus voltage and filters it to obtain FOC_UDCFLT. ADC channel 2 (external voltage divider) or channel 15 (internal voltage divider) can be selected. Range [0, 32767] Example: The bus voltage is scaled down by 1/6 before feeding into the ADC module, ADC VREF = 5V (namely, the sampling range is [0V ~ 30V]) and FOC_UDCFLT = 19661(0x4CCD), then bus voltage = 19661/32768 *5V*6 = 18V.

14.2.74 HFI_KP (0x40E0, 0x40E1)

HFI_KPH(0x40E0)								
Bit	15	14	13	12	11	10	9	8
Name	HFI_KP[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
HFI_KPL(0x40E1)								
Bit	7	6	5	4	3	2	1	0
Name	HFI_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]	HFI_KP	KP coefficient of PI controller under high frequency injection. MSB is always 0. Q12 format. Range [0, 32767] corresponds to range of Q12 [0, 8]

14.2.75 HFI_KI (0x40E2, 0x40E3)

HFI_KIH(0x40E2)								
Bit	15	14	13	12	11	10	9	8
Name	HFI_KI[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
HFI_KIL(0x40E3)								
Bit	7	6	5	4	3	2	1	0
Name	HFI_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]	HFI_KI	KI coefficient of PI controller under high frequency injection MSB is always 0. Q15 format. Range [0, 32767] corresponds to Q15 value range [0, 1]

14.2.76 OMEGA_HFI (0x40E4, 0x40E5)

OMEGA_HFIH(0x40D4)								
Bit	15	14	13	12	11	10	9	8
Name	OMEGA_HFI[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
OMEGA_HFIL(0x40D5)								

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

Bit	Name	Description
[15:0]	OMEGA_HFI	Output speed under high frequency injection Range [-32768, 32767]

14.2.77 HFI_ALPHAH (0x40E6, 0x40E7)

HFI_ALPHAH(0x40E6)								
Bit	15	14	13	12	11	10	9	8
Name	HFI_ALPHA[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
HFI_ALPHAL(0x40E7)								
Bit	7	6	5	4	3	2	1	0
Name	HFI_ALPHA[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]	HFI_ALPHAH	Dedicated for internal functional testing under high frequency injection Range [-32768, 32767]

14.2.78 HFI_BETAH (0x40E8, 0x40E9)

HFI_BETAH(0x40E8)								
Bit	15	14	13	12	11	10	9	8
Name	HFI_BETA[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
HFI_BETAL(0x40E9)								
Bit	7	6	5	4	3	2	1	0
Name	HFI_BETA[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]	HFI_BETAH	Dedicated for internal functional testing under high frequency injection Range [-32768, 32767]

14.2.79 FOC_ID0 (0x40EA, 0x40EB)

FOC_ID0H(0x40EA)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_ID0[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC_ID0L(0x40EB)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_ID0[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_ID0	d-axis current from coordinate transformation Range [-32768, 32767]

14.2.80 FOC_ID1 (0x40EC, 0x40ED)

FOC_ID1H(0x40EC)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_ID1[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC_ID1L(0x40ED)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_ID1[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_ID1	d-axis current obtained from the previous carrier wave Range [-32768, 32767]

14.2.81 DELTA_THETA (0x40EE, 0x40EF)

DELTA_THETAH(0x40EE)								
Bit	15	14	13	12	11	10	9	8
Name	DELTA_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
DELTA_THETAL(0x40EF)								
Bit	7	6	5	4	3	2	1	0

Name	DELTA_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]	DELTA_THETA	Angle difference of one carrier wave Range [-32768, 32767] The bit value -32768 ~ 32767 corresponds to angle range -180° ~ 180°

14.2.82 FOC_IQ0 (0x40F0, 0x40F1)

FOC_IQ0H(0x40F0)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_IQ0[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC_IQ0L(0x40F1)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_IQ0[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_IQ0	q-axis current from coordinate transformation Range [-32768, 32767]

14.2.83 FOC_IQ1 (0x40F2, 0x40F3)

FOC_IQ1H(0x40F2)								
Bit	15	14	13	12	11	10	9	8
Name	FOC_IQ1[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
FOC_IQ1L(0x40F3)								
Bit	7	6	5	4	3	2	1	0
Name	FOC_IQ1[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_IQ1	q-axis current obtained from the previous carrier wave Range [-32768, 32767]

14.2.84 HFI_I1H (0x40F4, 0x40F5)

HFI_I1H(0x40F4)								
Bit	15	14	13	12	11	10	9	8
Name	HFI_I1[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
HFI_I1L(0x40F5)								
Bit	7	6	5	4	3	2	1	0
Name	HFI_I1[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]	HFI_I1H	Dedicated for internal functional testing under high frequency injection Range [-32768, 32767]

14.2.85 HFI_I2H (0x40F6, 0x40F7)

HFI_I2H(0x40F6)								
Bit	15	14	13	12	11	10	9	8
Name	HFI_I2[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
HFI_I2L(0x40F7)								
Bit	7	6	5	4	3	2	1	0
Name	HFI_I2[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]	HFI_I2H	Dedicated for internal functional testing under high frequency injection Range [-32768, 32767]

14.2.86 HFI_I3H (0x40F8, 0x40F9)

HFI_I3H(0x40F8)								
Bit	15	14	13	12	11	10	9	8
Name	HFI_I3[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
HFI_I3L(0x40F9)								
Bit	7	6	5	4	3	2	1	0
Name	HFI_I3[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]	HFI_I3H	Dedicated for internal functional testing under high frequency injection Range [-32768, 32767]

15 Timer1

15.1 Timer1 Operations

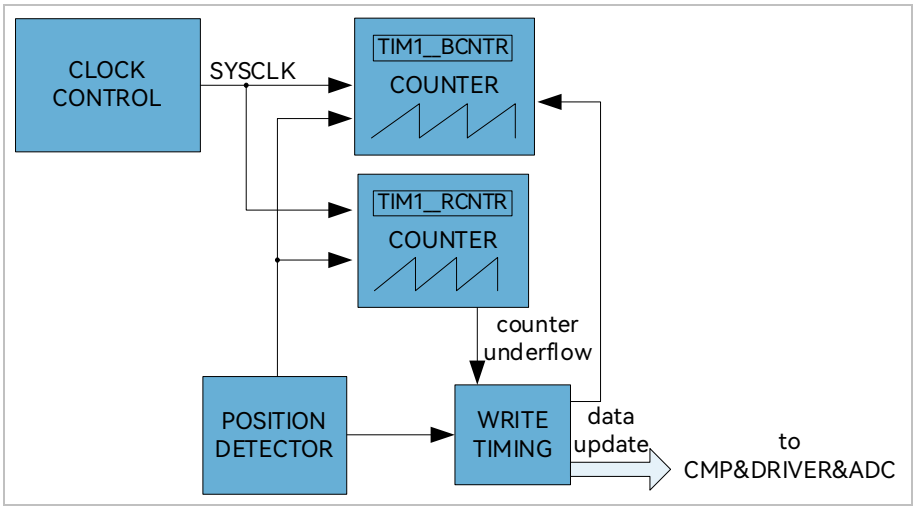
Timer1 consists of a 16-bit up-counting Base Timer and a 16-bit up-counting Reload Timer. Timer1 can be used for square-wave control applications of BLDC motors.

Timer1 features as follows.

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

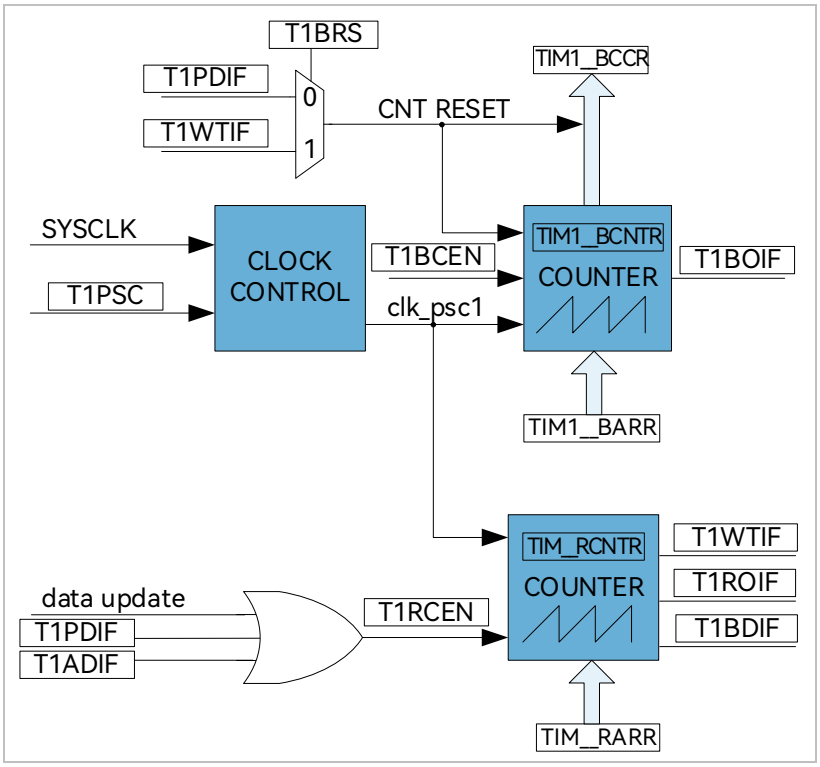
The internal structure of Timer1 is shown in Figure 15-1.

Figure 15-1 Internal Structure of Timer1



15.1.1 Timer1 Counter Module

Figure 15-2 Timebase Unit



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

15.1.1.1 Prescaler

Prescaler divides the system clock frequency and generates the counter clock source for Base Timer and Reload Timer. It offers 8 division coefficients and can be selected through TIM1_CR3[T1PSC]. Since this register has no buffer, the clock rate is immediately updated after the division coefficient is written. Therefore, the division coefficient shall be configured when both the Basic Timer and Reload Timer are not working. The clock rate clk_psc1 = SYSCLK/(2^TIM1_CR3[T1PSC]). The clock rate corresponding to TIM1_CR3[T1PSC] is shown in Table 15-1.

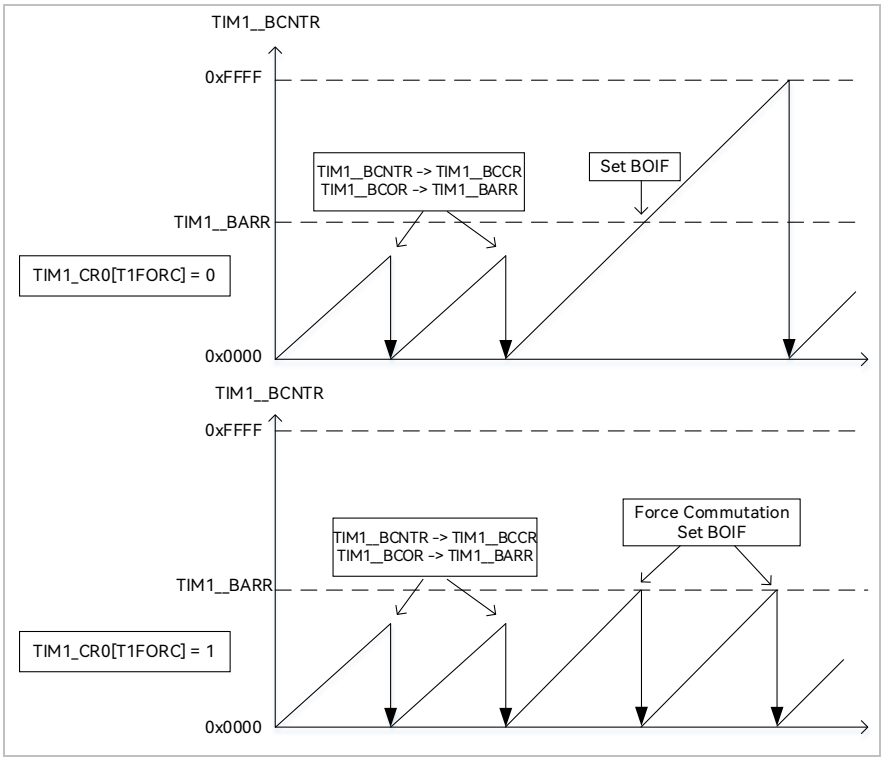
Table 15-1 Mapping between Clock Rate and TIM1_CR3[T1PSC]

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

15.1.1.2 Base Timer

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

Figure 15-3 Waveform of Base Timer

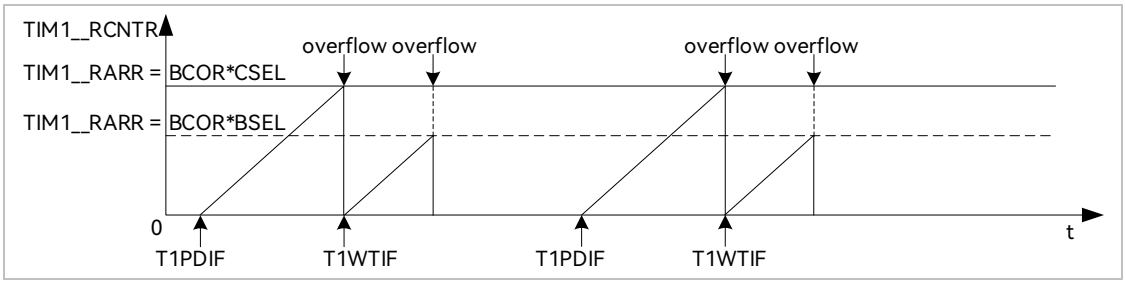


In Manual mode (TIM1_IER[T1MAME] = 1), TIM1__BCNTR is cleared by Base Timer Overflow event instead of TIM1_CR2[T1BRS].

15.1.1.3 Reload Timer

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

Figure 15-4 Waveform of Reload Timer

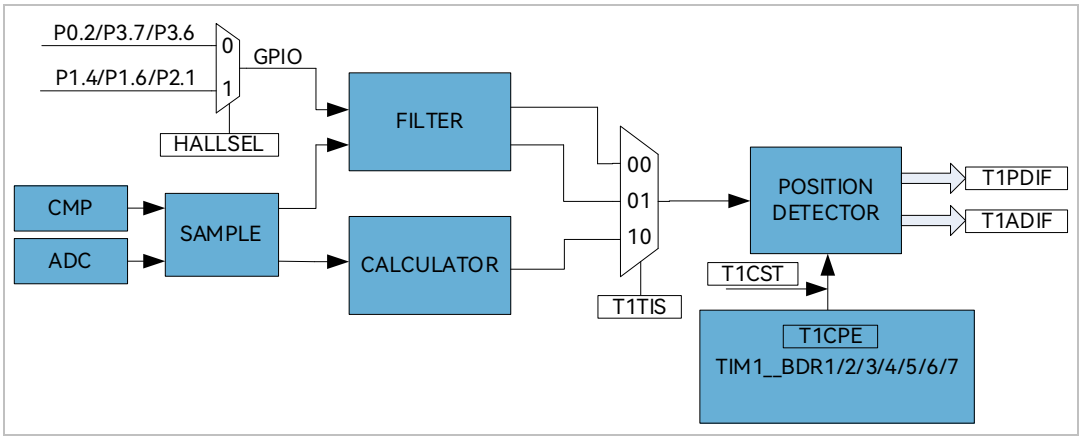


15.1.2 Position Detection

15.1.2.1 Position Detection Signal

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

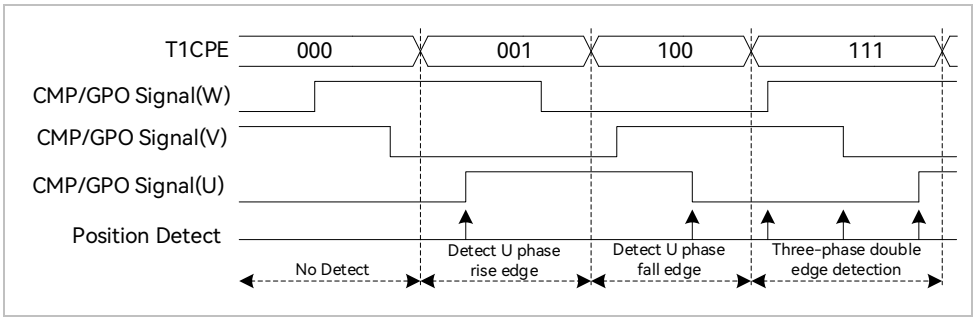
Figure 15-5 Functional Block Diagram of Position Detection



15.1.2.2 CMP/GPIO Position Detection Event

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

Figure 15-6 Timing Diagram of CMP/GPIO Position Detection



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

Table 15-2 Mapping between Active Edge and TIM1_DBR1/2/3/4/5/6/7[T1CPE]

T1CPE	Description	T1CPE	Description
000	0	100	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 falling edge of phase-U+W+V is detected

15.1.2.3 ADC Position Detection Event

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

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

Where,

K: ADC Position Detection Coefficient

TIM1_UCOP: ADC sampled value of active phase

TIM1_UFLP: ADC sampled value of floating phase

K/TIM1_UCOP/TIM1_UFLP definitions are determined by TIM1_DBR1/2/3/4/5/6/7[T1CPE] bit, as detailed in Table 15-3.

Table 15-3 Relation between TIM1_DBR1/2/3/4/5/6/7[T1CPE]
and K, TIM1_UCOP and TIM1_UFLP

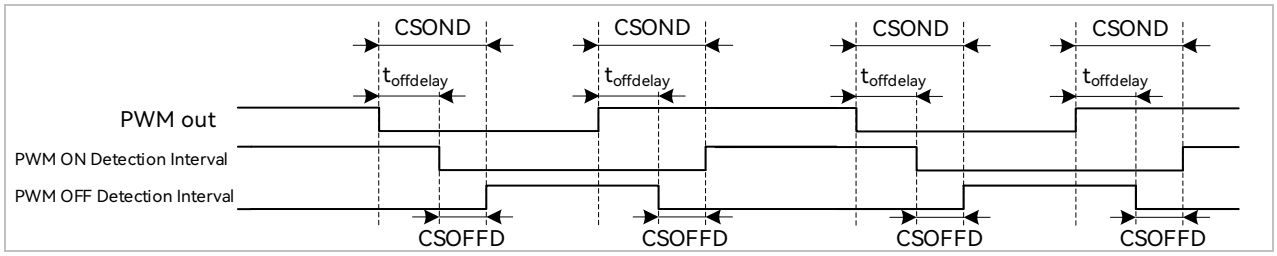
T1CPE	Description
000	Reserved
001	TIM1_KR for K, phase-W voltage for TIM1_UCOP, and phase-U voltage for TIM1_UFLP
010	TIM1_KF for K, phase-U voltage for TIM1_UCOP, and phase-W voltage for TIM1_UFLP
011	TIM1_KR for K, phase-U voltage for TIM1_UCOP, and phase-V voltage for TIM1_UFLP

T1CPE	Description
100	TIM1_KR for K, phase-V voltage for TIM1_UCOP, and phase-U voltage for TIM1_UFLP
101	TIM1_KR for K, phase-V voltage for TIM1_UCOP, and phase-W voltage for TIM1_UFLP
110	TIM1_KF for K, phase-W voltage for TIM1_UCOP, and phase-V voltage for TIM1_UFLP
111	Reserved

When TIM1_URES has a negative step or a positive step, an ADC Position Detection Interrupt is generated and TIM1_SR[T1ADIF] (Position Detection Interrupt Flag) is set to "1". The position at which ADC Position Detection Interrupt is generated is controlled by setting the coefficient K. In this case, the phase commutation degree can be controlled flexibly.

15.1.2.4 Sampling

Figure 15-7 Timing Diagram of Sampling

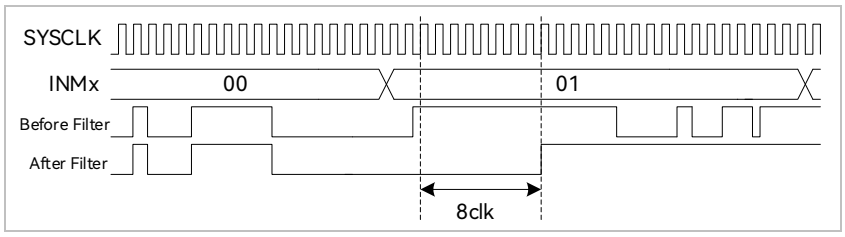


Affected by switching rate of the power device, BEMF signal lags behind PWM output. CMP_SAMR[CSOFFD], CMP_SAMR[CSOND] and CMP_CR4[FAEN] bits shall be set reasonably to adjust the sampling interval and obtain the valid position detection signal. When TIM1_CR3[T1TIS] = 01 or 10, Timer1 enables CMP0/1/2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

See section 31.1.4 for details.

15.1.2.5 Filtering

Figure 15-8 Timing Diagram of Filtering Module

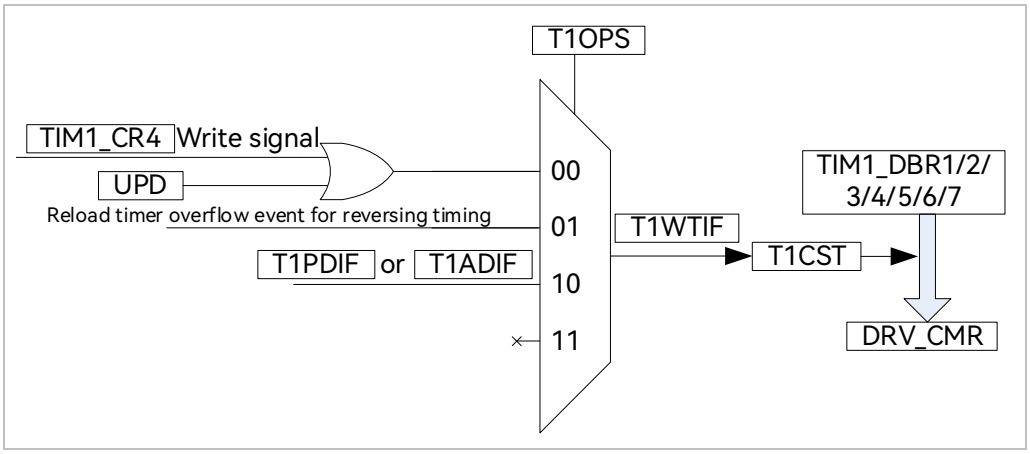


According to TIM1_CR3[T1INM] and CMP_CR4[FAEN], the filtered pulse width of input noise can be

selected as 8/16/24/32/64/96 system clock. After this feature is enabled, the signal is lagged behind about 8/16/24/32/64/96 system clocks.

15.1.3 Write Timing Interrupt

Figure 15-9 Write Timing Block Diagram



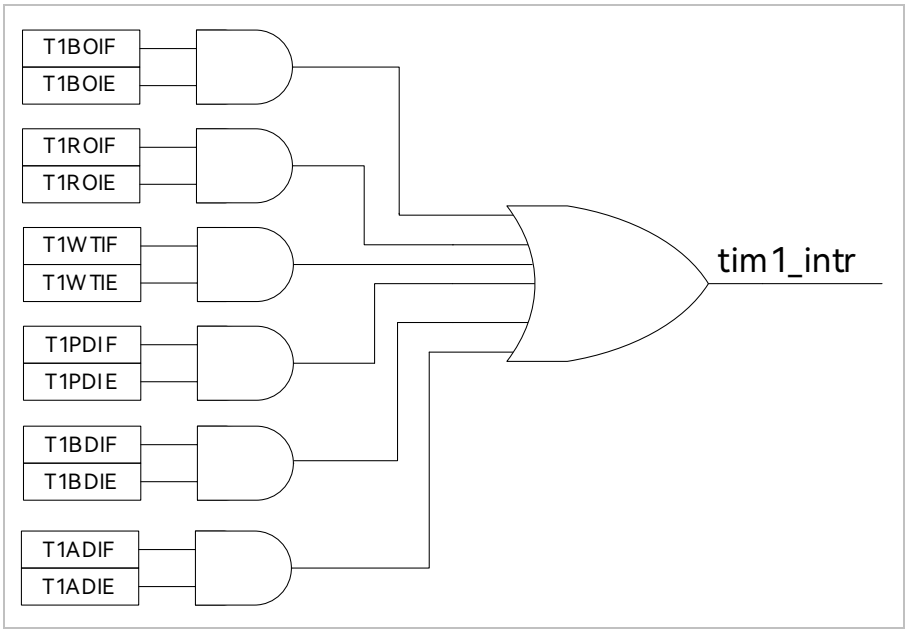
When the control logic, predefined in TIM1_DBR1/2/3/4/5/6/7, is sent to driver register DRV_CMR, a write timing interrupt is generated. The triggered source is selected by the configuration of TIM1_CR0[T1OPS], and software, Reload Timer overflow event or position detected event can be selected. When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

15.1.4 Timer1 Interrupt

Timer1 supports 6 interrupt sources:

- Base Timer overflow interrupt
- Reload Timer overflow interrupt
- Write timing interrupt
- Diode Freewheeling End Interrupt
- CMP/GPIO Position Detection Interrupt
- ADC Position Detection Interrupt

Figure 15-10 Timer1 Interrupt Sources



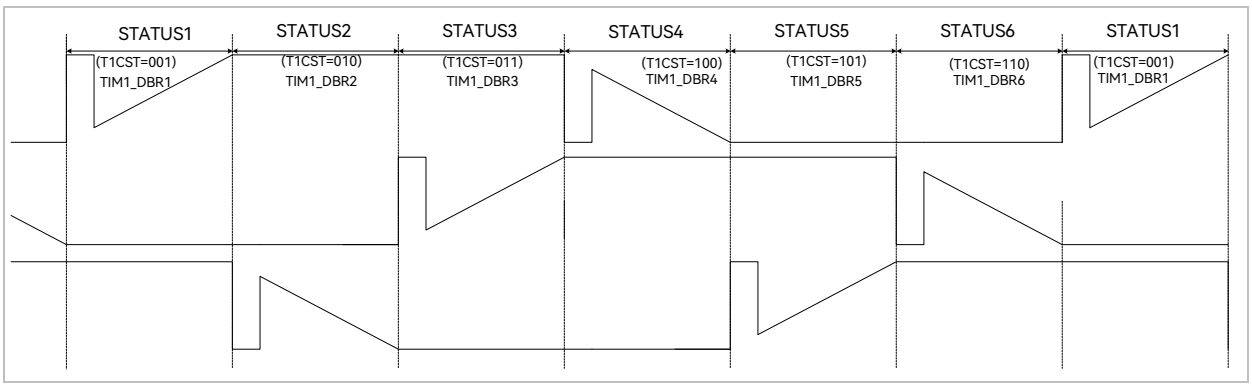
15.2 Square-wave Control Based BLDC Motor

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

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

15.2.1 Six-step Phase Commutation of Square Wave Control

Figure 15-11 Diagram of Six-step Phase Commutation of Square Wave Control

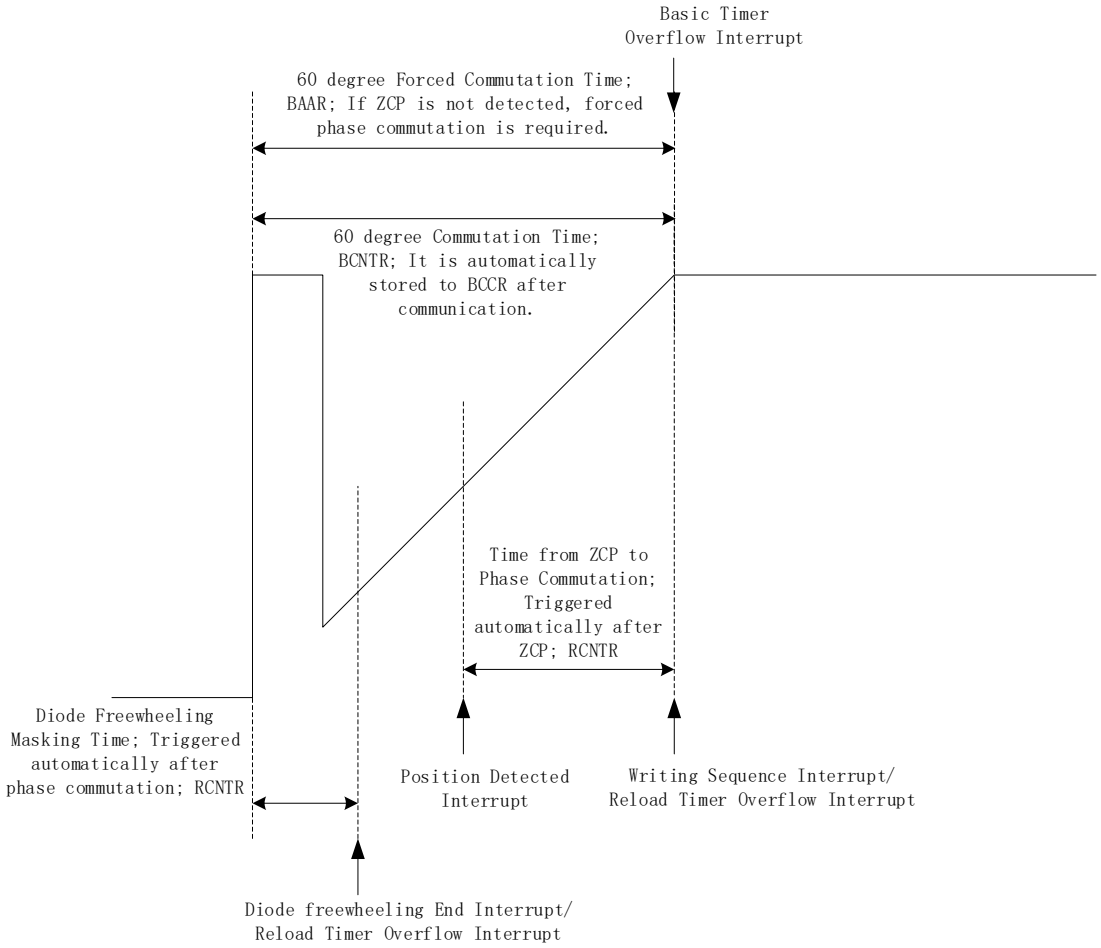


TIM1_CR4[T1CST] is the commutation state machine. Among them, state 0 is used to output off state, and state 7 is customizable for braking, pre-charging, pre-positioning, startup, etc. States 1 ~ 6 are used for six-step automatic commutation, and the state machine TIM1_CR4[T1CST] automatically adds 1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

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

15.2.2 Square Wave Control Working Principle

Figure 15-12 Working Principle of BLDC Motors



15.2.2.1 60° Commutation Base Time

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

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

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

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

15.2.2.2 Forced Commutation at 60°

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

In this case, TIM1_CR0[T1FORC] is set to “1” to enable the forced commutation feature. During previous commutation, the timer TIM1__BCNTR is cleared to “0” by write timing interrupt and restarts counting, while TIM1__BCCR captures the count value held in TIM1__BCNTR, which is filtered and stored in TIM1_BCOR as the 60 degree base time. When auto-load feature is enabled (TIM1_CR1[T1BAPE] = 1), the value held in TIM1_BCOR is loaded into TIM1__BARR after the Base Timer is cleared. If no ZCP is detected in 60 degree after commutation (TIM1__BCNTR matches TIM1__BARR), TIM1_SR[T1BOIF] (overflow interrupt flag of the Basic Timer) is set to “1” for forced phase commutation, and the timer TIM1__BCNTR is cleared to “0”.

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 no forced phase commutation is automatically performed. Phase commutation can be performed manually by Basic Timer overflow interrupt flag and the position detection interrupt flag.

15.2.2.3 Diode Freewheeling Masking

After commutation, inductance energy of the phase is released to the power supply or ground through the diode since the original active phase becomes a floating phase. During diode freewheeling, the floating phase BEMF signal cannot be measured. By masking comparator signal or ADC sampling value during diode freewheeling 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.

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

15.2.2.4 Angle of Position Detection to Commutation

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

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

15.2.2.5 Cycle-by-cycle Current Limiting

See section 31.1.1.2.

15.3 Timer1 Registers

15.3.1 TIM1_CR0 (0x4068)

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

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

		configured simultaneously to enable or disable TIM1_CR0[T1RCEN]. A write of "0x81" to TIM1_CR0 enables TIM1_CR0[T1RCEN] and "0x80" disables TIM1_CR0[T1RCEN]. TIM1_CR0[T1RCEN] is automatically enabled upon a Position Detection Interrupt and a Write Timing Interrupt. TIM1_CR0[T1RCEN] is cleared to "0" by hardware upon a Reload Timer Overflow Interrupt. TIM1_CR0[T1RCEN] cannot be automatically enabled or disabled by hardware in Manual mode. 0: Disable 1: Enable

15.3.2 TIM1_CR1 (0x4069)

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

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

15.3.3 TIM1_CR2 (0x406A)

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

Bit	Name	Description
[7]	T1BRS	Base Timer Reset Source Selection This bit is invalid in Manual mode (TIM1_IER[T1MAME] = 1). TIM1__BCNTR can only be cleared by a BCNTR Overflow Interrupt. 0: Write Timing Reset 1: Position Detection Interrupt Reset
[6:0]	CSEL	Phase Commutation Angle Selection After a position detection event, phase commutation is implemented after the degree configured by TIM1_CR2[CSEL]. Equation: Commutation angle = TIM1_CR2[CSEL]/128*60°

15.3.4 TIM1_CR3 (0x406B)

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

Bit	Name	Description
[7]	RSV	Reserved
[6:4]	T1PSC	Timer Clock Source Frequency Selection This bit is configured to divide the system clock as the clock source for Base Timer and Reload Timer. The clock source frequency of the two timers: 000: 24MHz 001: 12MHz 010: 6MHz 011: 3MHz 100: 1.5MHz 101: 750kHz 110: 375kHz 111: 187.5kHz
[3:2]	T1TIS	Position Detection Signal Selection 00: GPIO (P1.4, P1.6, P2.1 or P0.2, P3.6 according to HALL_CR[HALLSEL]) 01: Output signal of CMP0/1/2 10: Output signal of ADC 11: Reserved
[1:0]	T1INM	Filter Pulse Width for Position Detection Signal Selection When pulse width of the input signal is less than the set value, it is filtered as noise. The filtering time changes according to CMP_CR4[FAEN]. When CMP_CR4[FAEN] = 0:

		00: 4 system clock cycles 01: 8 system clock cycles 10: 16 system clock cycles 11: 24 system lock cycles When CMP_CR4[FAEN] = 1: 00: 32 system clock cycles 01: 64 system clock cycles 10: 96 system clock cycles 11: 128 system clock cycles

15.3.5 TIM1_CR4 (0x406C)

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

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

Table 15-4 Mapping between TIM1_CR4[T1CST] and TIM1_DBRx

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

15.3.6 TIM1_IER (0x406D)

Bit	7	6	5	4	3	2	1	0
Name	T1UPD	T1MAME	T1ADIE	T1BOIE	T1ROIE	T1WTIE	T1PDIE	T1BDIE
Type	W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	T1UPD	When TIM1_CR0[T1OPS] = 00, a write of “1” to this bit enables data transfer. This bit is cleared to “0” by hardware after “1” is written.
[6]	T1MAME	Manual Mode Enable With this bit enabled, Basic Timer and Reload Timer act 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 operated by software only. TIM1_RCNTR of the Reload Timer can be cleared to “0” upon a Reload Timer Overflow Interrupt only. TIM1_RARR of the Reload Timer cannot be updated automatically, and is operated by software only. 0: Disable 1: Enable
[5]	T1ADIE	ADC Position Detection 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	CMP/GPIO Position Detection Interrupt Enable 0: Disable 1: Enable
[0]	T1BDIE	Diode Freewheeling Masking Interrupt Enable 0: Disable 1: Enable

15.3.7 TIM1_SR (0x406E)

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

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	T1ADIF	ADC Position Detection Interrupt Flag A Position Detection Interrupt is generated when TIM1_DBRx[T1CPE] matches ACD Position Detection signal. Read:

		0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	T1BOIF	Base Timer Overflow Interrupt Flag An overflow event occurs when Basic Timer counts up and TIM1_BCNTR matches with TIM1_BARR. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[3]	T1ROIF	Reload Timer Overflow Interrupt Flag An overflow event occurs and TIM1_RCNTR is cleared to “0” when TIM1_RCNTR matches TIM1_RARR. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[2]	T1WTIF	Write Timing Interrupt Flag A Write Timing Interrupt is generated when TIM1_DBRx is transferred to DRV_CMR. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[1]	T1PDIF	CMP/GPIO Position Detection Interrupt Flag A position detection interrupt is generated when CMP/GPIO Position Detection matches TIM1_DBRx[T1CPE]. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[0]	T1BDIF	Diode Freewheeling Masking End Interrupt Flag Diode freewheeling masking starts after phase commutation and an interrupt is generated at end.

		Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

15.3.8 TIM1_BCOR (0x4070, 0x4071)

TIM1_BCORH(0x4070)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_BCOR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_BCORL(0x4071)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCOR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_BCOR	This bit is configured to capture filtered count values held in Base Timer. TIM1_BCCR hods the filtered count value, i.e. 60 Degree Base Time.

15.3.9 TIM1_CR5 (0x4072)

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

Bit	Name	Description
[7]	T1POP	Data Transfer Triggered by Driver Counter Overflow This bit is valid only when TIM_CR0[T1OPS] = 00. With it enabled, data transfer is triggered by Driver Counter Overflow, namely, commuting the phase once every PWM cycle 0: Disable 1: Enable
[6]	T1WTS	Commutation enabled at PWM OFF to remove narrow pulses. PWM Synchronization Enable 0: Disable 1: Enable
[5:4]	RSV	Reserved

[3]	ITRIP_DIS	Bus Current Sampling Disable 0: Disable 1: Enable
[2]	UCOP_DIS	Active Phase Voltage Sampling Disable 0: Disable 1: Enable
[1:0]	T1AFL	ADC Sampled Voltage Calculation Filtering Counts 00: 1 01: 2 10: 4 11: 8

15.3.10 TIM1_DBR1 (0x4074, 0x4075)

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

Bit	Name	Description
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Input Edge Polarity and Comparator Enable Select This bit is used to define the edge of Position Detection Input and enable or disable the associated comparators. The detected edge in input signal, corresponding to the configuration, generates a position detection interrupt. See CMP/GPIO Position Detection Event and Table 15-2.
[11]	T1WHP	High-side Output Polarity of phase-W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of phase-W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of phase-V 0: Active High 1: Active Low

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

Note:

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

15.3.11 TIM1_DBR2 (0x4076, 0x4077)

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

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

[4]	T1WLE	Low-side Output Enable of phase-W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of phase-V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of phase-V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of phase-U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of phase-U 0: Disable 1: Enable

15.3.12 TIM1_DBR3 (0x4078, 0x4079)

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

Bit	Name	Description
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Input Edge Polarity and Comparator Enable Select This bit is used to define the edge of Position Detection Input and enable or disable the associated comparators. The detected edge in input signal, corresponding to the configuration, generates a position detection interrupt. See CMP/GPIO Position Detection Event and Table 15-2.
[11]	T1WHP	High-side Output Polarity of phase-W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of phase-W 0: Active High 1: Active Low

[9]	T1VHP	High-side Output Polarity of phase-V 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of phase-V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of phase-U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of phase-U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of phase-W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of phase-W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of phase-V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of phase-V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of phase-U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of phase-U 0: Disable 1: Enable

15.3.13 TIM1_DBR4 (0x407A, 0x407B)

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

Reset	0	0	0	0	0	0	0	0

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

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

15.3.14 TIM1_DBR5 (0x407C, 0x407D)

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

Bit	Name	Description
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Input Edge Polarity and Comparator Enable Select This bit is used to define the edge of Position Detection Input and enable or disable the associated comparators. The detected edge in input signal, corresponding to the configuration, generates a position detection interrupt. See CMP/GPIO Position Detection Event and Table 15-2.
[11]	T1WHP	High-side Output Polarity of phase-W 0: Active High 1: Active Low
[10]	T1WLP	Low-side Output Polarity of phase-W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of phase-V 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of phase-V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of phase-U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of phase-U 0: Active High 1: Active Low

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

15.3.15 TIM1_DBR6 (0x407E, 0x407F)

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

Bit	Name	Description
[15]	RSV	Reserved
[14:12]	T1CPE	Position Detection Input Edge Polarity and Comparator Enable Select This bit is used to define the edge of Position Detection Input and enable or disable the associated comparators. The detected edge in input signal, corresponding to the configuration, generates a position detection interrupt. See CMP/GPIO Position Detection Event and Table 15-2.
[11]	T1WHP	High-side Output Polarity of phase-W 0: Active High 1: Active Low

[10]	T1WLP	Low-side Output Polarity of phase-W 0: Active High 1: Active Low
[9]	T1VHP	High-side Output Polarity of phase-V 0: Active High 1: Active Low
[8]	T1VLP	Low-side Output Polarity of phase-V 0: Active High 1: Active Low
[7]	T1UHP	High-side Output Polarity of phase-U 0: Active High 1: Active Low
[6]	T1ULP	Low-side Output Polarity of phase-U 0: Active High 1: Active Low
[5]	T1WHE	High-side Output Enable of phase-W 0: Disable 1: Enable
[4]	T1WLE	Low-side Output Enable of phase-W 0: Disable 1: Enable
[3]	T1VHE	High-side Output Enable of phase-V 0: Disable 1: Enable
[2]	T1VLE	Low-side Output Enable of phase-V 0: Disable 1: Enable
[1]	T1UHE	High-side Output Enable of phase-U 0: Disable 1: Enable
[0]	T1ULE	Low-side Output Enable of phase-U 0: Disable 1: Enable

15.3.16 TIM1_DBR7 (0x4080, 0x4081)

TIM1_DBR7H(0x4080)								
Bit	15	14	13	12	11	10	9	8
Name	RSV	T1CPE			T1WHP	T1WLP	T1VHP	T1VLP
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0
TIM1_DBR7L(0x4081)								

Bit	7	6	5	4	3	2	1	0
Name	T1UHP	T1ULP	T1WHE	T1WLE	T1VHE	T1VLE	T1UHE	T1ULE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

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

15.3.17 TIM1_BCNTR (0x4082, 0x4083)

TIM1_BCNTRH(0x4082)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_BCNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0
TIM1_BCNTRL(0x4083)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_BCNTR	This bit holds count values of the Base Timer and is used for clocking commutation at 60°. Auto mode: TIM1_BCNTR register selects the reset source according to TIM1_CR2[T1BRS], and TIM1_BCNTR does not restart when TIM1_BCNTR overflow interrupt is generated. Manual mode: TIM1_BCNTR restarts when TIM1_BCNTR overflow interrupt is generated.

15.3.18 TIM1_BCCR (0x4084, 0x4085)

TIM1_BCCRH(0x4084)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_BCCR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_BCCRL(0x4085)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_BCCR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_BCCR	This bit is configured to capture count values held in Base Timer. Auto mode: When the Base Timer is reset on a Position Detection Interrupt or a Write Timing Interrupt, the count values before the reset are stored into TIM1_BCCR. Manual mode: When the Base Timer is reset on an Overflow Interrupt, the count values before the reset are stored into TIM1_BCCR.

15.3.19 TIM1_BARR (0x4086, 0x4087)

TIM1_BARRH(0x4086)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_BARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_BARRL(0x4087)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_BARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_BARR	Reload Value of Base Timer When the count value of the Base Timer equals to TIM1_BARR value, an overflow interrupt is generated and the counter is cleared to “0”.

15.3.20 TIM1_RARR (0x4088, 0x4089)

TIM1_RARRH(0x4088)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_RARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_RARRL(0x4089)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_RARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_RARR	Auto-Reload Value of Reload Timer When count of the Reload Timer is equal to TIM1_RARR, an overflow interrupt is generated and the value of timer is cleared to “0”.

		Auto mode: The value of diode freewheeling masking angle held in TIM1_CR1[BSEL] is updated to TIM1_RARR when a Write Timing Interrupt is generated. The value of commutation angle held in TIM1_CR2[CSEL] is updated to TIM1_RARR when a Position Detection Interrupt occurs. Manual mode: TIM1_RARR is written by software.

15.3.21 TIM1_RCNTR (0x408A, 0x408B)

TIM1_RCNTRH(0x408A)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_RCNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
TIM1_RCNTRL(0x408B)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_RCNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	Name	Description
[15:0]	TIM1_RCNTR	Count value of the Reload Timer for counting numbers of diode freewheeling masking and ZCP to phase commutation.  Note: In Manual mode, TIM1_RCNTR is cleared to “0” only by a Reload Timer overflow interrupt.

15.3.22 TIM1_UCOP (0x408C, 0x408D)

TIM1_UCOPH(0x408C)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_UCOP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
TIM1_UCOPL(0x408D)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_UCOP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_UCOP	ADC Sampled Value of active phase voltage (second-highest bit alignment)

15.3.23 TIM1_UFLP (0x408E, 0x408F)

TIM1_UFLPH(0x408E)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_UCOP[15:8]							
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_UFLPL(0x408F)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_UCOP[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_UFLP	ADC Sampled Value of floating phase voltage (second-highest bit alignment)

15.3.24 TIM1_URES (0x4090, 0x4091)

TIM1_URESH(0x4090)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_URES[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM1_URESL(0x4091)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_URES[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_URES	Result of ADC Position Detection formula; Q15 format

15.3.25 TIM1_KRMAX (0x4092)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_KRMAX							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	TIM1_KRMAX	Max. Coefficient of Raising Edge Range [0, 255]

15.3.26 TIM1_KFMIN (0x4093)

Bit	7	6	5	4	3	2	1	0
Name	TIM1_KFMIN							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	TIM1_KFMIN	Min. Coefficient of Falling Edge Range [0, 255]

15.3.27 TIM1_KF (0x4094, 0x4095)

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

Bit	Name	Description
[15:0]	TIM1_KF	ADC Position Detection Coefficient when floating phase voltage drops Range [0, 32767]

15.3.28 TIM1_KR (0x4096, 0x4097)

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

Bit	Name	Description
[15:0]	TIM1_KR	ADC Position Detection Coefficient when floating phase voltage rises Range [0, 32767]

15.3.29 TIM1_ITRIP (0x4098, 0x4099)

TIM1_ITRIPH(0x4098)								
Bit	15	14	13	12	11	10	9	8
Name	TIM1_ITRIP[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
TIM1_ITRIPL(0x4099)								
Bit	7	6	5	4	3	2	1	0
Name	TIM1_ITRIP[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM1_ITRIP	Filtered Bus Current When DRV_CNTR = 0, the hardware automatically samples the bus current and filters it for software application. The default channel is ADC channel 4. Range [0, 32767] Note: The value is obtained by averaging eight instantaneous sampled current

16 Timer2

16.1 Timer2 Instructions

Timer2 has the following five working modes:

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

Timer2 features:

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

16.1.1 Prescaler

Prescaler divides the system clock frequency and generates clock source for Base Timer. 8 frequency division coefficients of prescaler are available and can be selected by TIM2_CR0[T2PSC]. Since this register has no buffer, the clock source frequency is updated immediately after TIM2_CR0[T2PSC] is written. Therefore, the frequency division coefficients shall be configured when Basic Timer is not working. The clock source frequency formula is: clk_psc2 = SYSCLK/(2^TIM2_CR0[T2PSC]). The clock rate corresponding to different TIM2_CR0[T2PSC] value as shown in Table 16-1.

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

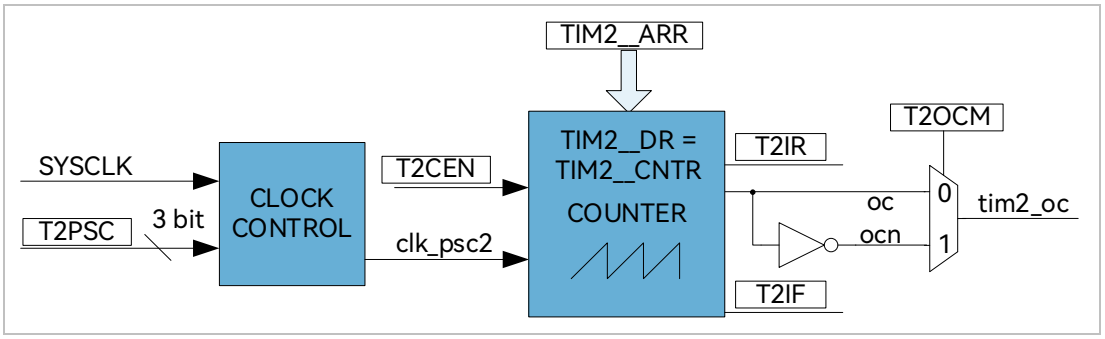
TIM2_CR0[T2PSC]	Division Factor	clk_psc2(Hz)	TIM2_CR0[T2PSC]	Division Factor	clk_psc2(Hz)
000	1	24M	100	16	1.5M
001	2	12M	101	32	750k
010	4	6M	110	64	375k
011	8	3M	111	128	187.5k

16.1.2 Reading, Writing and Counting of TIM2__CNTR

When TIM2_CR1[T2CEN] = 1, TIM2__CNTR starts to count. The write operation to TIM2__CNTR directly changes the value of the register, so Base Timer shall be disabled before the write operation. When reading TIM2__CNTR, software reads the high-order bytes first, and the hardware synchronously caches the low-order bytes. When reading the low-order bytes, the software reads the cached data.

16.1.3 Output Mode

Figure 16-1 Block Diagram of Output Mode



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

16.1.3.1 Reading and Writing of TIM2__ARR/TIM2__DR

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

TIM2__ARR/TIM2__DR is a 16-bit register, which requires to write the high byte first and then the low byte.

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

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

16.1.3.2 High/Low Level Output Mode

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

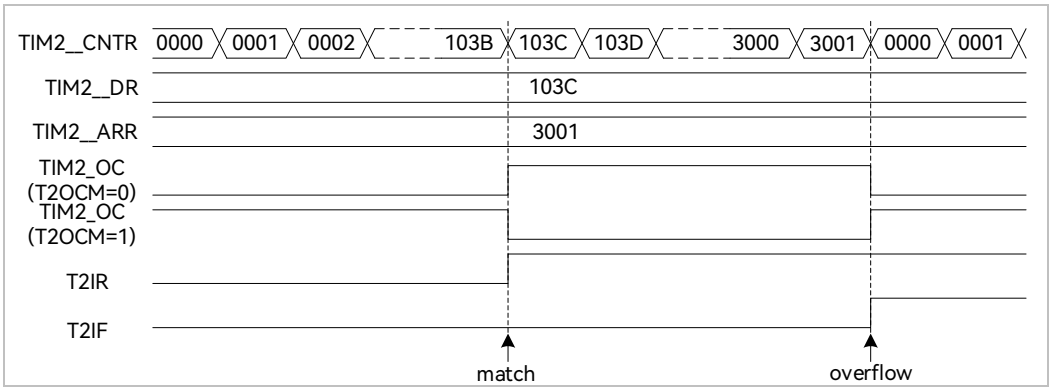
16.1.3.3 PWM Generation

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

16.1.3.4 Interrupt Events

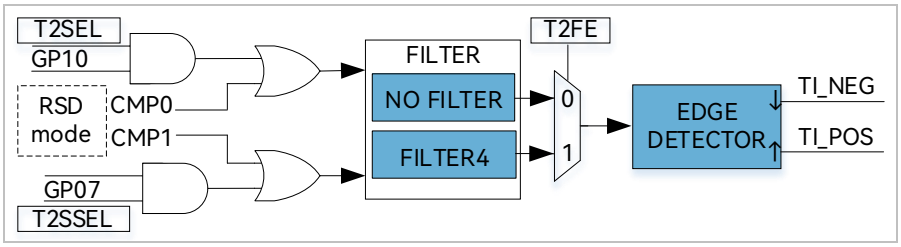
- When TIM2__CNTR = TIM2__DR, a compare match event is generated and the interrupt flag bit TIM2_CR1[T2IR] is set to “1”. The timer continues.
- When TIM2__CNTR = TIM2__ARR, an overflow event is generated, and the interrupt flag bit TIM2_CR1[T2IF] is set to “1”. The timer is cleared to “0” and then restarts.

Figure 16-2 Output Mode Waveform



16.1.4 Input Signal Filtering and Edge Detection

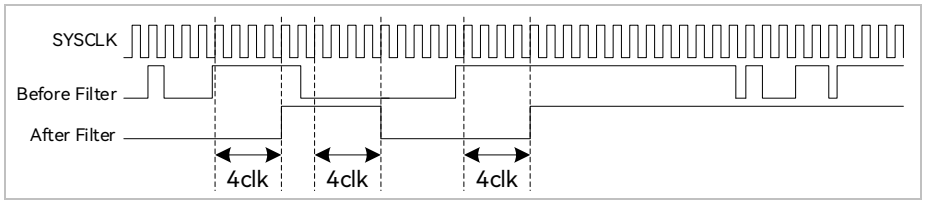
Figure 16-3 Block Diagram of Input Signal Filtering and Edge Detection



The input signal of Timer2 comes from P0.7 or P1.0, set by PH_SEL[T2SEL] and PH_SEL[T2SSEL] (See section 22.3.15). The filter of input signal is optional.

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

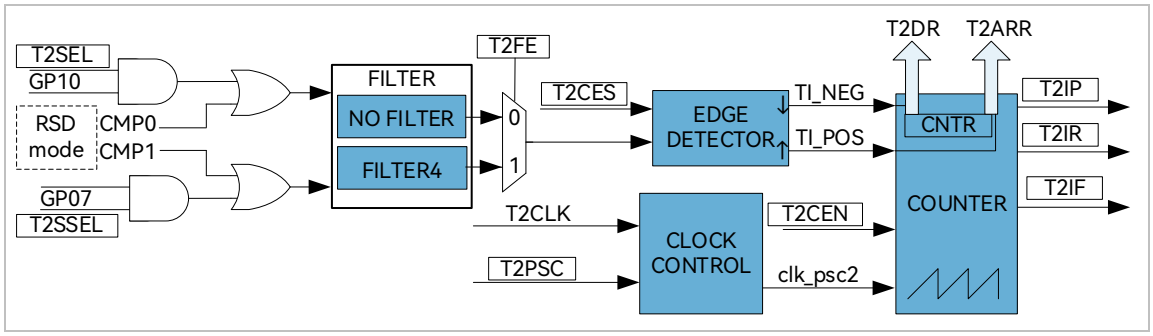
Figure 16-4 Timing Diagram of Filter Module



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

16.1.5 Input Capture Mode

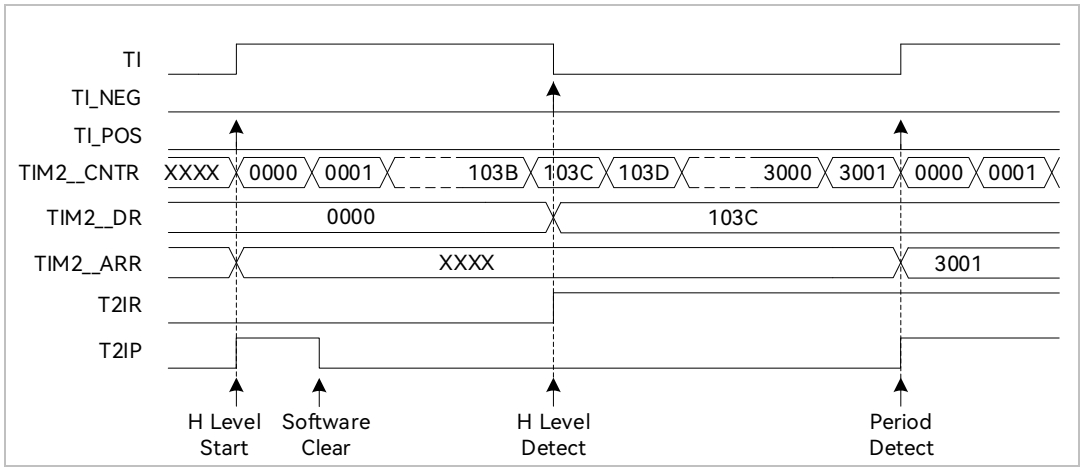
Figure 16-5 Schematic Diagram of Input Capture Mode



The input capture mode detects duty cycle and period of the PWM signal. When TIM_CR0[T2CES] = 0, the time between two adjacent rising edges forms one cycle, and the time from rising edge to falling edge forms the pulse

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

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

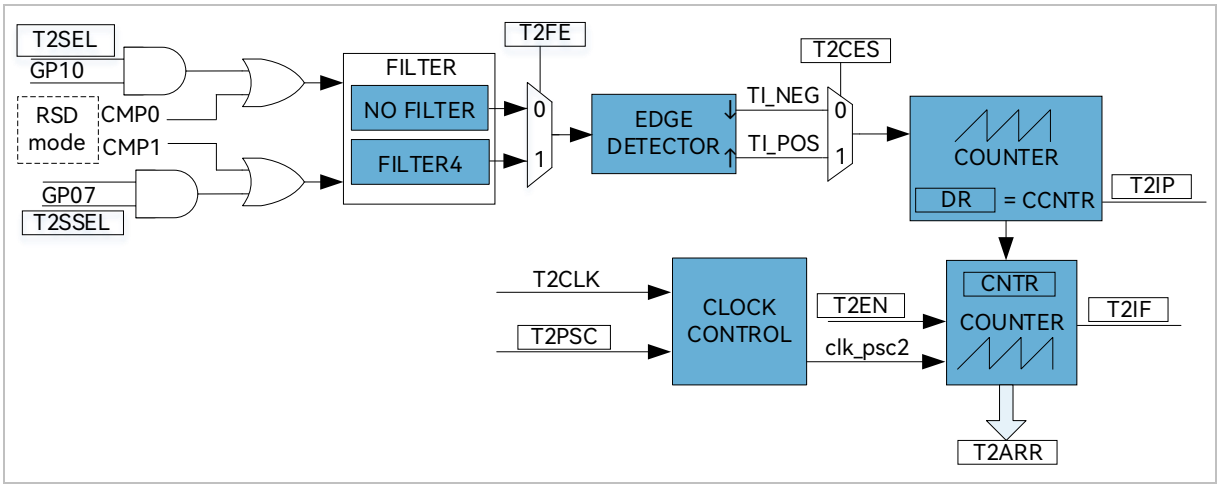


For example, when TIM2_CR0[T2CES] = 0, TIM2_CR1[T2CEN] is set to “1” to enable the Base Timer. When the first rising edge of the input (falling edge is invalid) is detected, TIM2__CNTR is cleared and restarts. When falling edge of the input is detected, the value of TIM2__CNTR is stored in TIM2__DR,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

An overflow event occurs if Timer2 does not detect the second rising edge of the input and TIM2__CNTR reaches 0xFFFF. In this case, the interrupt flag TIM2_CR1[T2IF] is set to “1”, and TIM2__CNTR is cleared to “0” and restarts. At this point, TIM2__ARR value is 0xFFFF, and the TIM2__DR value is determined by the input level and TIM2_CR0[T2OCM] XOR.

16.1.6 Input Counter Mode

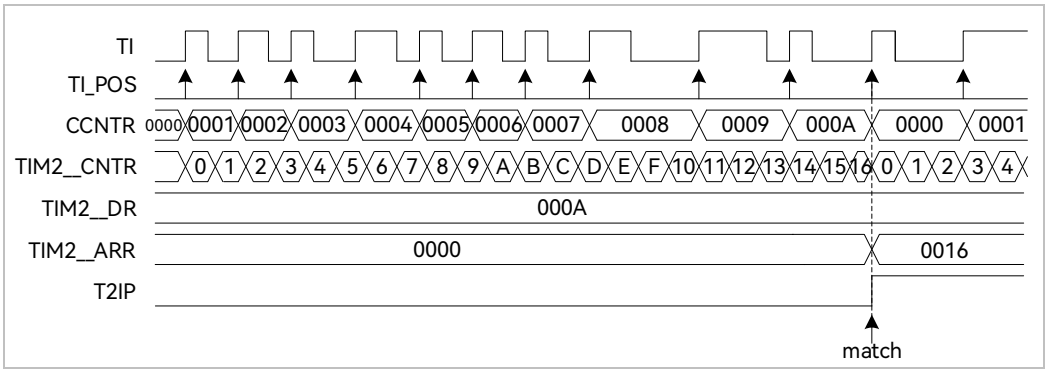
Figure 16-7 Schematic Diagram of Input Counter Mode



In input counter mode, TIM2__DR includes preload register and shadow register. When the software writes TIM2__DR register, the data is saved in the preload register first, and then sent to the shadow register in case of compare match event (TIM2_CR1[T2IP] = 1), overflow event (TIM2_CR1[T2IF] = 1) or timer disable (TIM2_CR1[T2CEN] = 0). TIM2__DR is a 16-bit register, which requires the software writes the high byte first and then the low byte. The hardware ensures that the data in the preload register is not updated to the shadow register after the high byte is written and before the low byte is written.

The input counter mode is used to detect the time to input the set PWM waves. When the number of input PWM counted by the special timer CCNTR reaches the set value (TIM2__DR), TIM2__CNTR is stored in TIM2__ARR. When TIM2_CR0[T2CES] is set to “1”, the rising edge of the input PWM signal serves as the active counting edge of the special timer; and when TIM2_CR0[T2CES] is set to “0”, the falling edge of the input signal as the active edge.

Figure 16-8 Timing Diagram of Input Counter Mode



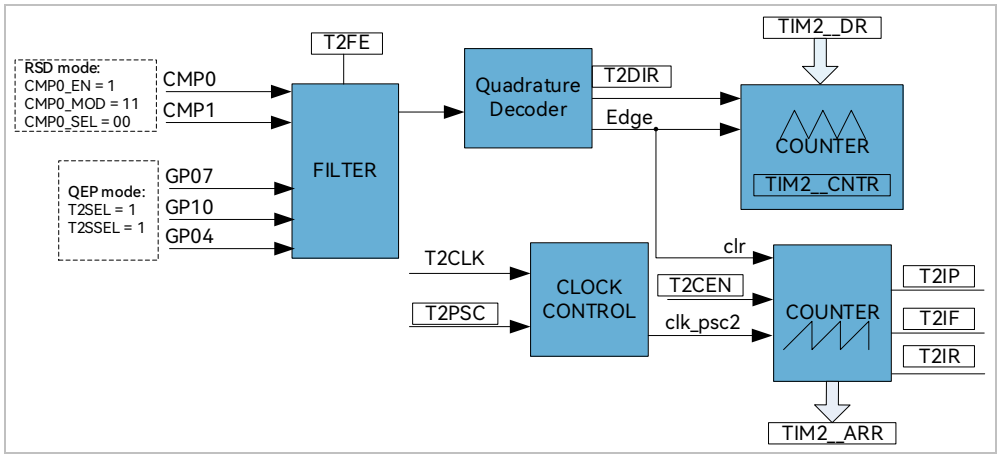
The Basic Timer is enabled when TIM2_CR1[T2CEN] is set to “1”. If the first active edge of the input signal

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

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

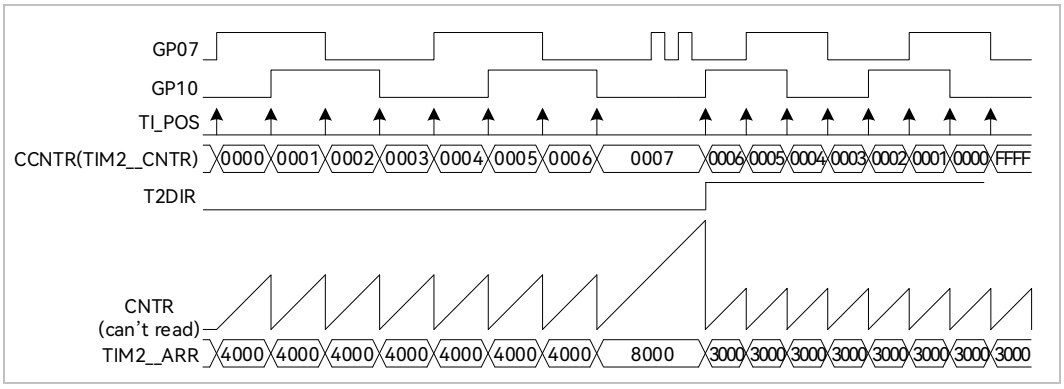
16.1.7 QEP&RSD Mode

Figure 16-9 Schematic Diagram of QEP&RSD Mode



QEP&RSD mode obtains relative position, direction and speed of the motor by detecting orthogonal signals on two channels. P0.7 and P1.0 (QEP mode) or CMP0, CMP1 (RSD mode) are the input signal sources, which are sent to the quadrature decoding module from the filtering module to obtain active edge and direction (TIM2_CR1[T2DIR]).

Figure 16-10 Timing Diagram of QEP&RSD Mode

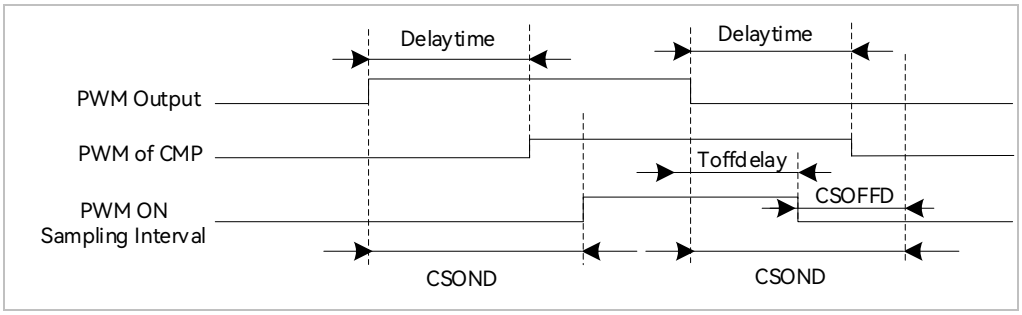


The special timer is an up/down counter, and the signal source is the active edge from orthogonal decoding module. If TIM2_CR1[T2DIR] = 0, the direction is positive, and special timer counts upward. When the active edge arrives, the counter increases by one. If TIM2_CR1[T2DIR] = 1, the direction is reverse and special timer counts down. When the active edge arrives, the timer decreases by one. In QEP Mode, after configuring the code value held in TIM2__DR, the count-up timer is cleared to “0” and restarts when it reaches TIM2__DR, and the count-down timer is reloaded with TIM2__DR when it decrements up to 0. The mechanical zero signal (“Z signal”) of QEP encoder is input from P04 and generates the interrupt event flag TIM2_CR1[T2IR].

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

16.1.7.1 RSD Comparator Sampling

Figure 16-11 PWM ON Sampling Mode

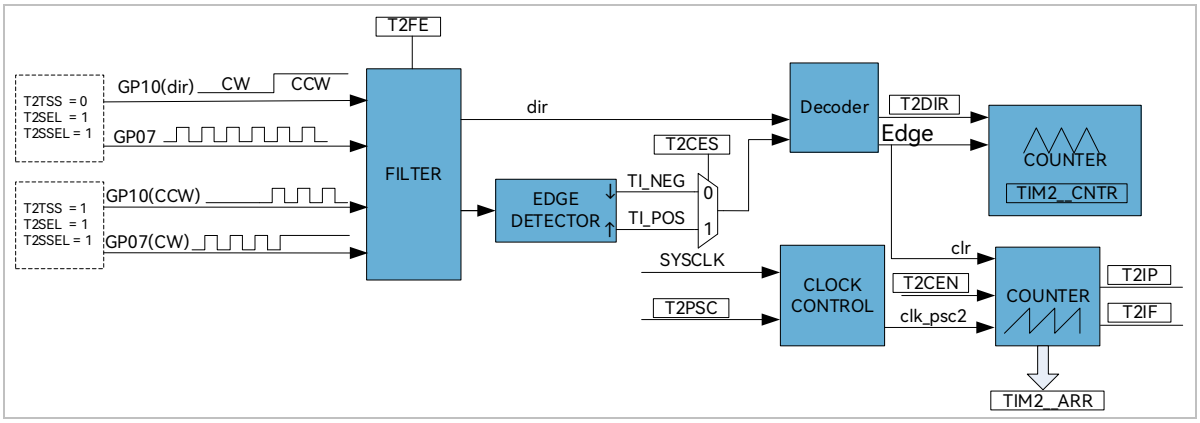


The Start of Sampling (“SoS”) time delay and End of Sampling (“EoS”) time advance must be set in order to sample correct BEMF comparison signals in RSD Sampling mode.

See section 31.1.4 for details.

16.1.8 Step Mode

Figure 16-12 Schematic Diagram of Step Mode



In step mode, relative position, direction and speed of the step motor are obtained by detecting inputs of the two channel: P1.0 for direction input and P0.7 for pulse input. Setting TIM2_CR0[T2CES] selects the rising edge or falling edge as the active edge. The input signals are sent to decoding module from the filtering module to obtain the active edge and direction TIM2_CR1[T2DIR].

Note:

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

Figure 16-13 Timing Diagram of Step Mode

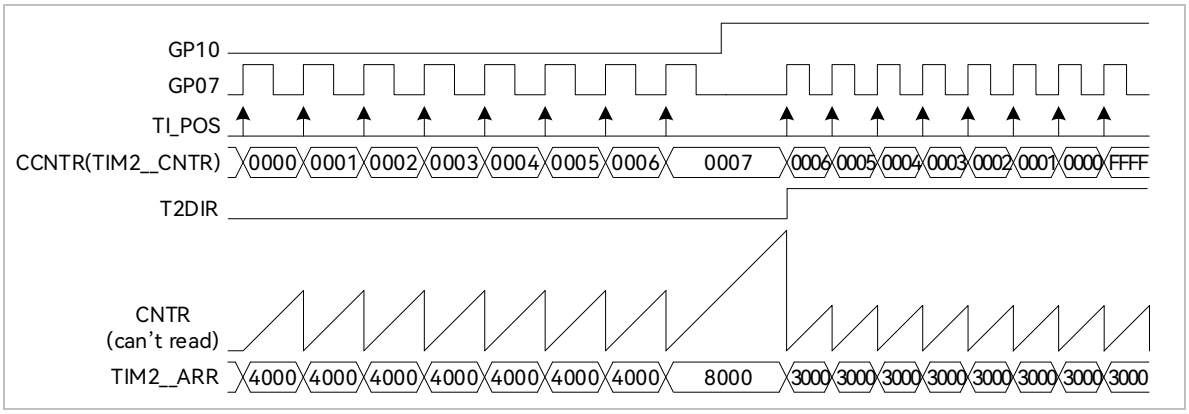
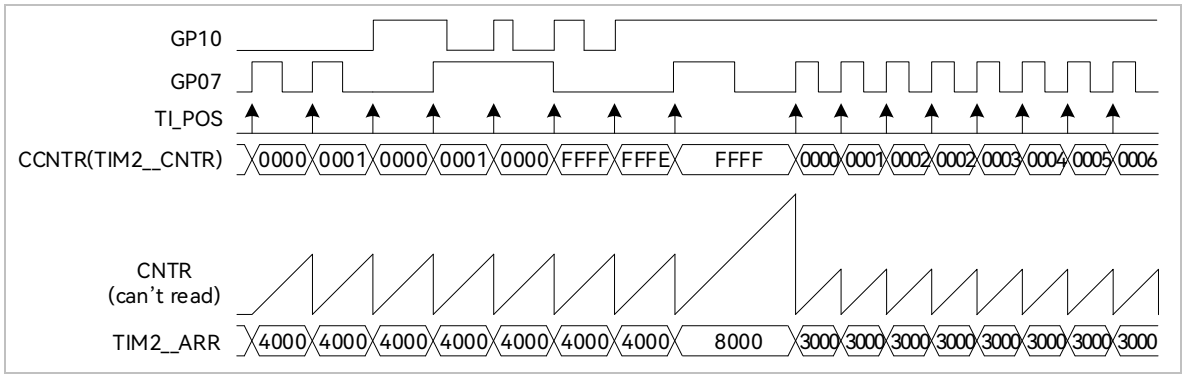


Figure 16-14 Timing Diagram of Positive + Negative Pulse Input State in Step Mode

(Raising Edge Selected as Active Edge)



The special timer is an up/down-counter, and the signal source is active edge of the encoding module. When P1.0 = 0, TIM2_CR1[T2DIR] = 0, the direction is forward. If active edge of P0.7 arrives, the special timer CCNTR increases by 1. When P1.0 = 1, TIM2_CR1[T2DIR] = 1 and the direction is reverse. If active edge of P0.7 arrives, CCNTR decreases by 1. If count value of the special timer reaches 65535 from 0, it is automatically cleared to "0". If it decreases from 65535 to 0, it is automatically reset to 65535. TIM2__CNTR is read to obtain the value of special time.

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 the interrupt flag bit TIM2_CR1[T2IP] is set to "1". When Base Timer counts to 0xFFFF, the count overflows and the interrupt flag TIM2_CR1[T2IF] is set to "1".

16.2 Timer2 Registers

16.2.1 TIM2_CR0 (0xA1)

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

Bit	Name	Description
[7:5]	T2PSC	Base Timer Clock Prescaler Selection It is configured to divide the system clock frequency and generate the clock source for Base Timer. The prescaled clock rates are configured as follows: 000: 24MHz 001: 12MHz 010: 6MHz 011: 3MHz 100: 1.5MHz 101: 750kHz 110: 375kHz 111: 187.5kHz
[4]	T2OCM	Output Mode: Output Mode Selection 0: Output “0” when TIM2__CNTR < TIM2__DR; output “1” when TIM2__CNTR ≥ TIM2__DR 1: Output “1” when TIM2__CNTR < TIM2__DR; output “0” when TIM2__CNTR ≥ TIM2__DR Input Counter Mode: No effect Input Capture Mode: TIM2__DR indicates the input level to be selected when timer TIM2__CNTR becomes overflowed. 0: TIM2__DR is reset to “0” by hardware for low level input upon an overflow interrupt and is set to “0xFFFF” for high level input upon an overflow interrupt. 1: TIM2__DR is reset to “0” by hardware for high level input upon an overflow interrupt and is set to “0xFFFF” for low level output upon an overflow interrupt. QEP&RSD Mode and Step Mode Selection 0: QEP&RSD Mode 1: Step Mode
[3]	T2IRE	Output Mode: Compare Match Interrupt Enable Input Capture Mode: Pulse Width Detection Interrupt Enable Input Counter Mode: No effect QEP Mode: QEP Encoder Z Signal Interrupt Enable 0: Disable 1: Enable
[2]	T2CES	Output Mode: No effect Input Capture Mode: Counting Edge Selection 0: The time between two adjacent raising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). The time between two adjacent falling edges forms one cycle, and the time from falling edge to raising

		edge forms the pulse width (LOW). Input Counter Mode: Active Edge Selection 0: Falling Edge Count 1: Raising Edge Count QEP&RSD Mode: Enable Pulse Counter Cleared upon Z Signal Interrupt INT1 0: Disable 1: Enable Step Mode: Active Edge Selection 0: Falling Edge Count 1: Raising Edge Count
[1:0]	T2MOD	Mode Selection 00: Input Capture Mode 01: Output Mode 10: Input Counter Mode 11: QEP&RSD Mode or Step Mode

16.2.2 TIM2_CR1 (0xA9)

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

Bit	Name	Description
[7]	T2IR	Output Mode: Compare Match Interrupt Flag Input Capture Mode: Pulse Width Detection Interrupt Flag Input Counter Mode: No effect QEP Mode: QEP Encoder Z Signal Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	T2IP	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag Input Counter Mode: PWM Input Counter Match Interrupt Flag QEP&RSD Mode or Step Mode: Active Edge Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0”. 1: No effect

[5]	T2IF	Output Mode: Base Timer Overflow Interrupt Flag, which is set to “1” when TIM2_CNTR matches TIM2_ARR. Input Capture Mode: Base Timer Overflow Interrupt Flag, which is set to “1” when the Timer has not detected an input PWM cycle but the timer TIM2_CNTR value reaches 0xFFFF. Input Counter Mode: Special-purpose Counter overflow Interrupt Flag, which is set to “1” when the input PWM cycle has not reached the preset TIM2_DR value but the Base Timer TIM2_CNTR value reaches 0xFFFF. QEP&RSD Mode or Step Mode: Base Timer Overflow Interrupt Flag, which is set to “1” and Basic Timer is cleared to “0” when Basic Timer reaches to 0xFFFF. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	T2IPE	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Enable Input Counter Mode: PWM Input Counter Match Interrupt Enable QEP&RSD Mode or Step Mode: Active Edge Detection Interrupt Enable 0: Disable 1: Enable
[3]	T2IFE	Output Mode: Base Timer Overflow Interrupt Enable Input Capture Mode: Base Timer Overflow Interrupt Enable Input Counter Mode: Base Timer Overflow Interrupt Enable QEP&RSD Mode or Step Mode: Base Timer Overflow Interrupt Enable 0: Disable 1: Enable
[2]	T2FE	Input Signal Filter Selection When TIM2_CR1[T2FE] = 1, input signals are filtered out as noise if the pulse width is less than 4 clock cycles. Assuming that the system clock runs at 24MHz (41.67ns) , then the pulse width for filtering is 166.67/333.34ns. 0: Signals are not filtered 1: Signals are filtered on every 4 clock cycles
[1]	T2DIR	QEP&RSD Mode: Indicator of Motor Rotation Direction Rotation direction of the motor is determined according to the phase relationship of the two input signals. Step Mode: Indicator of Motor Rotation Direction Rotation direction of the motor is determined according to the direction signal P1.0. 0: Forward 1: Backward

[0]	T2CEN	Base Timer Enable 0: Disable 1: Enable

16.2.3 TIM2_CNTR (0xAA, 0xAB)

TIM2_CNTRH(0xAB)								
Bit	15	14	13	12	11	10	9	8
Name	TIM2_CNTR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2_CNTRL(0xAA)								
Bit	7	6	5	4	3	2	1	0
Name	TIM2_CNTR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM2_CNTR	Output Mode/Input Capture Mode/Input Counter Mode: Count values held in the Base Timer QEP&RSD Mode or Step Mode: count values held in the special timer

16.2.4 TIM2_DR (0xAC, 0xAD)

TIM2_DRH(0xAD)								
Bit	15	14	13	12	11	10	9	8
Name	TIM2_DR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2_DRL(0xAC)								
Bit	7	6	5	4	3	2	1	0
Name	TIM2_DR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM2_DR	Output Mode: Compare match value (written by software) Input Capture Mode: Count value of the detected input pulse width (written by hardware) Input Counter Mode: PWM cycles to be counted (written by software) QEP Mode: Encoder value Step Mode: No effect

16.2.5 TIM2_ARR (0xAE, 0xAF)

TIM2_ARRH(0xAF)								
Bit	15	14	13	12	11	10	9	8
Name	TIM2_ARR[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TIM2_ARRL(0xAE)								
Bit	7	6	5	4	3	2	1	0
Name	TIM2_ARR[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	TIM2_ARR	Output Mode: PWM cycle (written by software) Input Capture Mode: Count value held in Base Timer of a PWM cycle (written by hardware) Input Counter Mode: Count value held in Base Timer when the input PWM count matches (written by hardware) QEP&RSD Mode or Step Mode: Count value held in Base Timer when the input signal is detected as an active edge (written by hardware)

17 Timer3/Timer4

17.1 Timer3/Timer4 Instructions

Timer3/Timer4 support output and input modes:

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

Timer3/Timer4 Features:

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

17.1.1 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 is: clk_pscx = SYSCLK/(2^TxPSC). The clock rate corresponding to different TIMx_CR0[TxPSC] value as shown in Table 17-1.

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

TIMx_CR0[TxPSC]	Division Factor	clk_pscx(Hz)	TIMx_CR0[TxPSC]	Division Factor	clk_pscx(Hz)
000	0x1	24M	100	0x10	1.5M

TIMx_CR0[TxPSC]	Division Factor	clk_pscx(Hz)	TIMx_CR0[TxPSC]	Division Factor	clk_pscx(Hz)
001	0x2	12M	101	0x20	750k
010	0x4	6M	110	0x40	375k
011	0x8	3M	111	0x80	187.5k

Note:

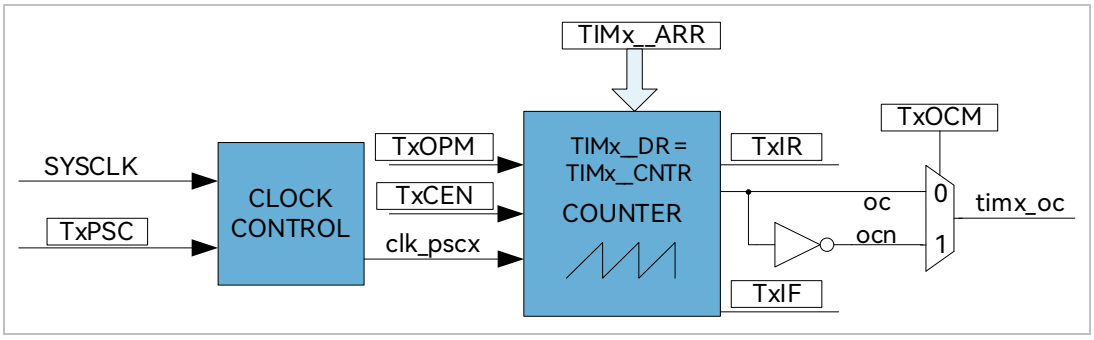
In Input Capture Mode, the clock rate of Timer3 is 48MHz when TIM3_CR0[T3PSC] = 111.

17.1.2 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 bytes first and then low-order bytes, and the hardware caches the low-order bytes simultaneously. When reading low-order bytes, the software reads the cached data.

17.1.3 Output Mode

Figure 17-1 Block Diagram of Output Mode



The output mode generate output signals according to TIMx_CR0[TxOCM], and the comparison results between TIMx_CNTR and registers TIMx_DR, TIMx_ARR. Meanwhile, corresponding interrupts is generated.

17.1.3.1 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.

17.1.3.2 PWM Generation

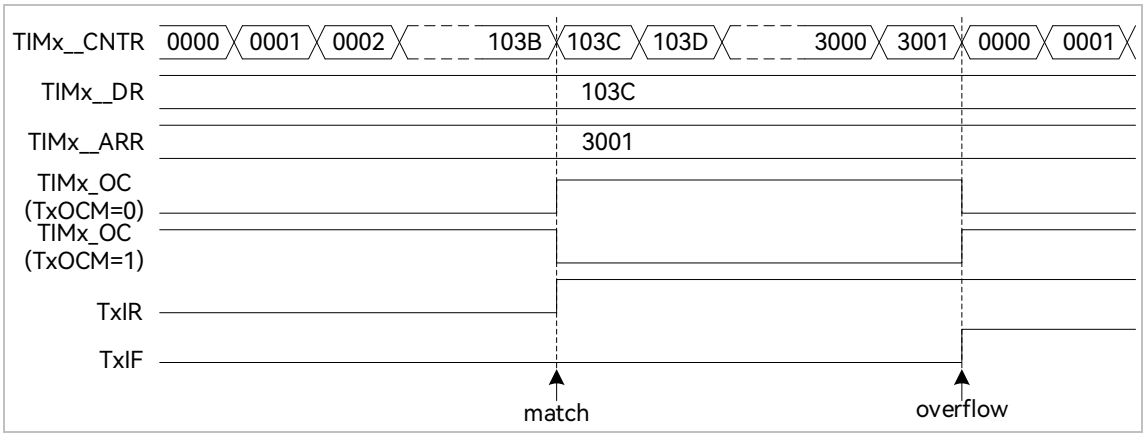
In PWM generation mode, TIMx_ARR determines PWM cycle, and TIMx_DR determines the duty cycle,

and duty cycle = TIMx_DR/TIMx_ARR*100%. If TIMx_CR0[TxOCM] = 0, the low level is output when TIMx_CNTR < TIMx_DR, and the high level is output when TIMx_CNTR ≥ TIMx_DR. If TIMx_CR0[TxOCM] = 1, the high level is output when TIMx_CNTR < TIMx_DR, and low level is output when TIMx_CNTR ≥ TIMx_DR. When TIMx_CNTR > TIMx_ARR, the output signal is reversed.

17.1.3.3 Interrupt Event

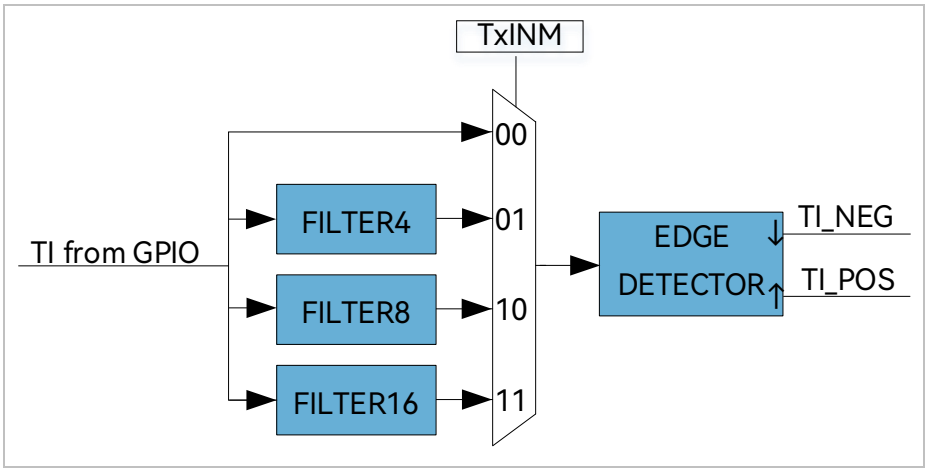
- When TIMx_CNTR = TIMx_DR, a compare match interrupt is generated. The interrupt flag TIMx_CR1[TxIR] is set to "1", and the timer continues.
- When TIMx_CNTR = TIMx_ARR, an overflow event is generated. The interrupt flag TIMx_CR1[TxIF] is set to "1", and the timer is cleared to "0". TIMx_CR0[TxOPM]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

Figure 17-2 Output Waveform of Output Mode



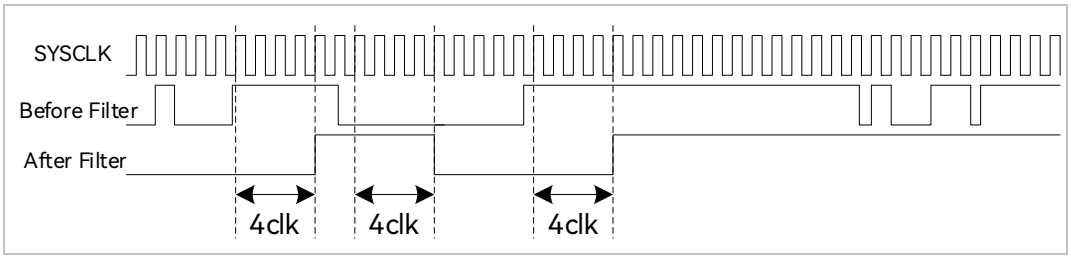
17.1.4 Input Signal Filtering and Edge Detection

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



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

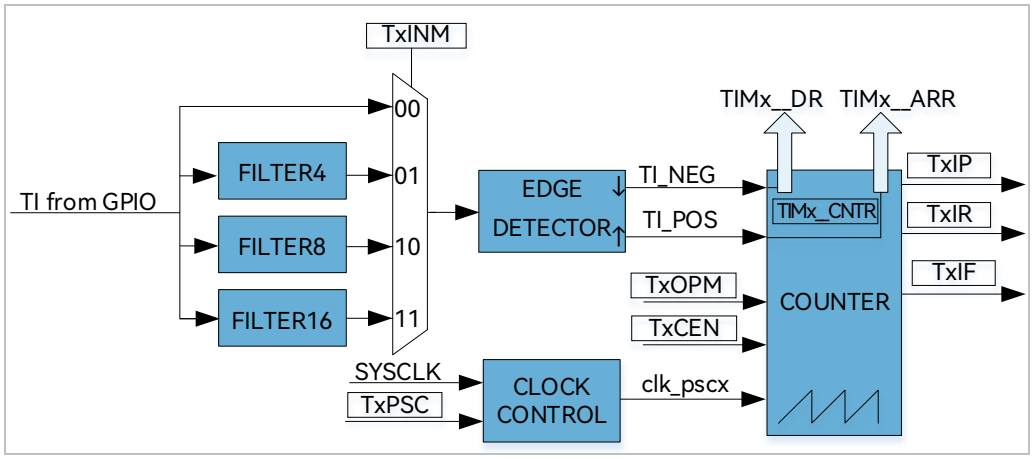
Figure 17-4 Timing Diagram of Filter Module



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

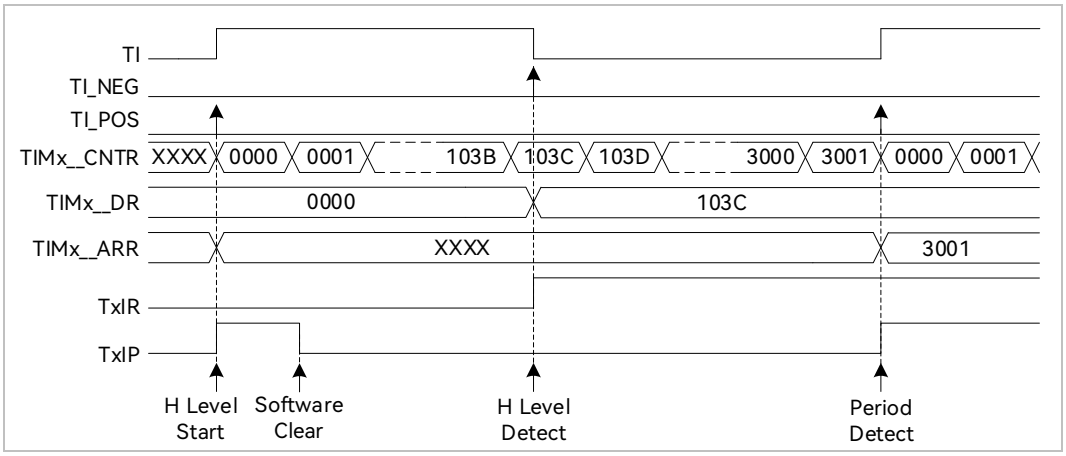
17.1.5 Input Capture Mode

Figure 17-5 Schematic Diagram of Input Capture Mode



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

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



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

17.1.6 Timer4 FG Output Mode

See FG Output Generation for details.

17.2 Timer3/Timer4 Registers

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

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

Bit	Name	Description
[7:5]	TxPSC	Base TimerClock Prescaler Selection It is configured to divide the system clock frequency and generate the clock source for Base Timer. The prescaled clock rates are configured as follows: 000: 24MHz 001: 12MHz 010: 6MHz 011: 3MHz 100: 1.5MHz 101: 750kHz 110: 375kHz 111: 187.5kHz Note: In Input Capture Mode, the clock rate of Timer3 is 48MHz when this bit is set to “111”.
[4]	TxOCM	Output Mode: Output Mode Selection 0: Output “0” when TIMx_CNTR < TIMx_DR; output “1” when TIMx_CNTR ≥ TIMx_DR 1: Output “1” when TIMx_CNTR < TIMx_DR; output “0” when TIMx_CNTR ≥ TIMx_DR Input Capture Mode: TIMx_DR indicates the input level to be selected when the active edge is detected or the timer becomes overflowed. Active Edge Selection 0: The time between two adjacent raising edges forms one cycle, and the time from rising edge to falling edge forms the pulse width (HIGH). 1: The time between two adjacent falling edges forms one cycle, and the time from falling edge to raising edge forms the pulse width (LOW). TIMx_DR indicates the input level to be selected when the timer becomes overflowed. 0: TIMx_DR is reset to “0” by hardware for low level input upon an overflow interrupt and is set to “0xFFFF” by hardware for high level input upon an overflow interrupt. 1: TIMx_DR is reset to “0” by hardware for high level input upon an overflow interrupt and is set to “0xFFFF” for low level input upon an overflow interrupt.
[3]	TxIRE	Output Mode: Compare Match Interrupt Enable Input Capture Mode: Pulse Width Detection Interrupt Enable

		0: Disable 1: Enable
[2]	RSV	Reserved
[1]	TxOPM	Single Mode Base Timer stops in any of the following events: Output Mode: Base Timer overflow Input Capture Mode: PWM Cycle Detection or Base Timer overflow 0: Base Timer does not stop 1: Base Timer stops (TIMx_CR1[TxEN] is reset to “0”)
[0]	TxMOD	Working Mode Selection 0: Input Capture Mode 1: Output Mode

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

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

Bit	Name	Description
[7]	TxIR	Output Mode: Compare Match Interrupt Flag Input Capture Mode: Pulse Width Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	TxIP	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[5]	TxIF	Output Mode: Base Timer Overflow Interrupt Flag, which is set to”1” when TIMx_CNTR matches TIMx_ARR. Input Capture Mode: Base Timer Overflow Interrupt Flag, which is set to “1” when the Timer does not detect an input PWM cycle but TIMx_CNTR reaches 0xFFFF. Read: 0: No Interrupt Pending

		1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[4]	TxIPE	Output Mode: No effect Input Capture Mode: PWM Cycle Detection Interrupt Enable 0: Disable 1: Enable
[3]	TxIFE	Output Mode: Base Timer Overflow Interrupt Enable Input Capture Mode: Base Timer Overflow Interrupt Enable 0: Disable 1: Enable
[2:1]	TxINM	Input Signal Filtering Pulse Width Selection Input signals are filtered as noise if pulse width is less than the defined value. 00: Signals are not filtered 01: Signals are filtered on every 4 system clock cycles 10: Signals are filtered on every 8 system clock cycles 11: Signals are filtered on every 16 system clock cycles
[0]	TxEN	Base Timer Enable 0: Disable 1: Enable

17.2.3 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

17.2.4 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 detected input pulse width (written by hardware)

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

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

Bit	Name	Description
[15:0]	TIMx_ARR	Output Mode: Reload value (written by hardware). See FG Output Generation for details. Input Capture Mode: Count value of a detected PWM cycle (written by hardware)

18 Systick

18.1 Systick Instructions

The chip can generate Systick interrupts at a fixed interval, and the interrupt cycle is controlled by SYST_ARR. Systick interrupt is enabled when DRV_SR[SYSTIE] is set to “1”, and the interrupt entry is accessed by 10.

18.2 Systick Registers

18.2.1 DRV_SR (0x4061)

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

Bit	Name	Description
[7]	SYSTIF	Systick Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	SYSTIE	Systick Interrupt Enable 0: Disable 1: Enable
[5]	FGIF	FG Interrupt Flag When FOC Drive/Square Wave Drive is enabled, an FGIF Interrupt is generated in each rotation cycle (electrical cycle). Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect

[4]	DCIF	Driver Compare Match Interrupt Flag When the Driver counter value is equal to DRV_COMR, the system decides whether to generate an interrupt according to the counting direction set by DRV_SR[DCIM]. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[3]	FGIE	FG Interrupt Enable When FOC Drive/Square Wave Drive is enabled, an FG Interrupt is generated in each rotation cycle (electrical cycle). 0: Disable 1: Enable
[2]	DCIP	Number of PWM cycles required to generate a Driver Compare Match Interrupt 0: 1 interrupt in 1 PWM cycle 1: 1 interrupt in 2 PWM cycles
[1:0]	DCIM	Compare Match Interrupt Mode Selection The system decides whether to generate an interrupt according to DRV_SR[DCIM] when Driver counter value is equal to DRV_COMR. 00: No interrupt is generated. 01: An interrupt is generated when the counter counts up 10: An interrupt is generated when the counter counts down 11: An interrupt is generated when the counter counts up/down

18.2.2 SYST_ARR (0x4064, 0x4065)

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

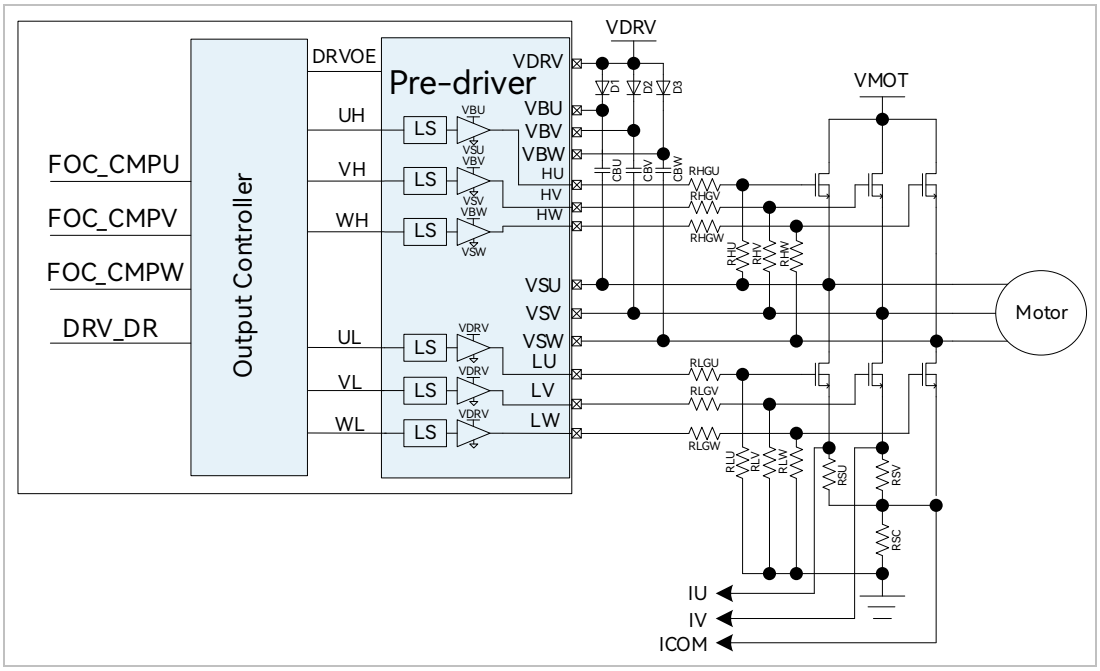
Bit	Name	Description
[15:0]	SYST_ARR	Systick Reloaded Value This bit decides the cycle when Systick interrupts are generated, which defaults to 1ms. Calculation formula: Systick interrupt rate = SYSCLK/(SYST_ARR[15:0] + 1) Range [0, 65535]

19 Driver

19.1 Driver Instructions

19.1.1 Driver Introduction

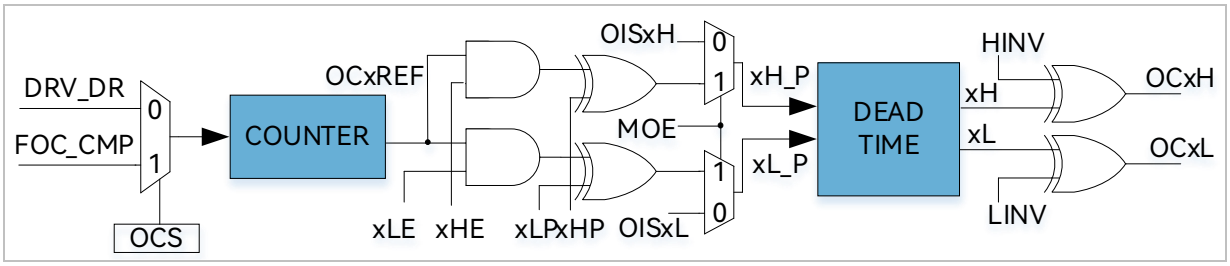
Figure 19-1 Block Diagram of FU6367Q Driver Module



FOC_CMPU/V/W is the three-way comparison value output by FOC module, and DRV_DR is the comparison value set by the software. The above comparison value outputs three sets of level signals U/V/W to pre-driver (FU6367Q) after passing through the output control module. The U/V/W three-way output is applied to DC brushless motor control, and the U/V/W/X four-way output to step motor control.

19.1.2 Output Control Module

Figure 19-2 Block Diagram of Output Control Module



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

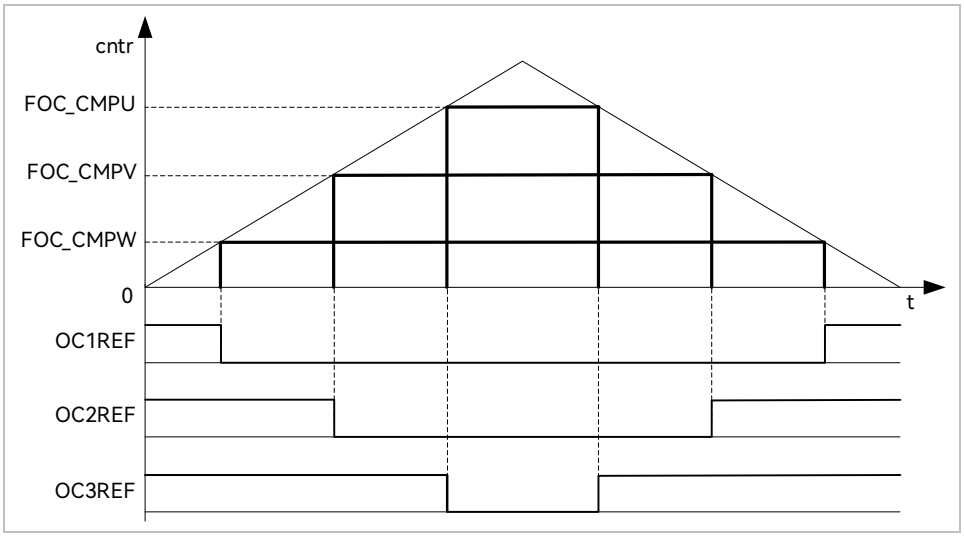
If DRV_CR[OCS] = 0, comparison value of PWM comes from DRV_DR. Otherwise, it comes from FOC_CMP and U/V/W/X output signals (OCxREF) are generated. DRV_CMR[xHE], DRV_CMR[xLE], polarity control bits DRV_CMR[xHP] and DRV_CMR[xLP] are configured for logic processing of OCxREF signal. Enabling DRV_OUT[MOE] outputs PWM waveform, otherwise, the idle level. xH_P and xL_P output signals are transferred to the deadtime module to generate xH and xL signals. The PI_CR[HINV] and PI_CR[LINV] bits are configured to output PWM drive signals required by the high and low levels.

19.1.2.1 Count and Compare Module

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

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

Figure 19-3 PMW Generation



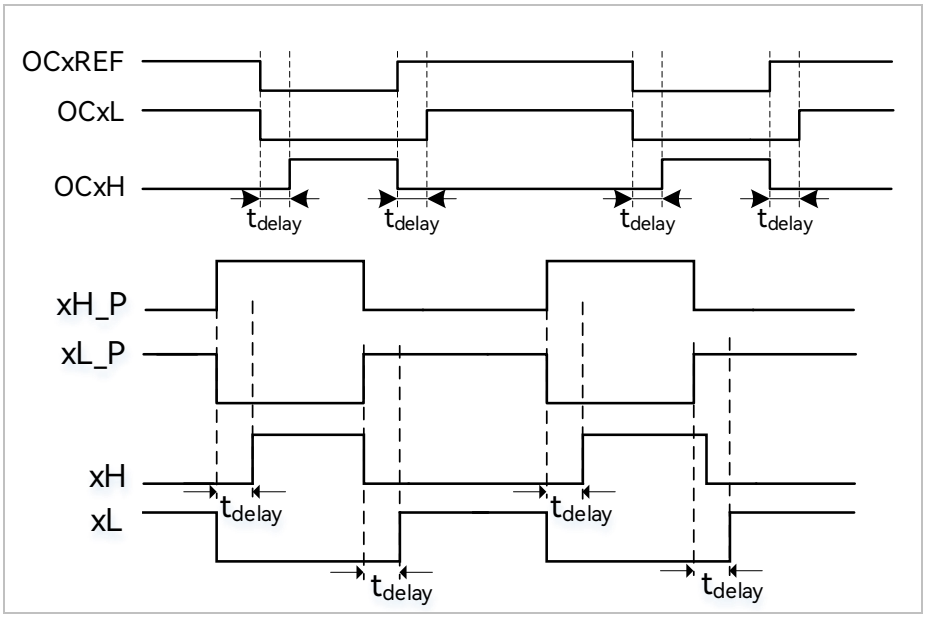
When DRV_CR[OCS] = 0, DRV_DR set by software is compared with the count value to generate

OC1REF/OC2REF/OC3REF with the same duty cycle. Duty cycle = DRV_DR/DRV_ARR*100%.

19.1.2.2 Deadtime Module

xH_P and xL_P signals are available for 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 three channels have the same deadtime, which is set by DRV_DTR. When rising edge signals are detected, output high level of xH and xL is delayed for a period of time set in DRV_DTR; if the delayed time is greater than the output pulse width, the associated channel pulse width is not delayed.

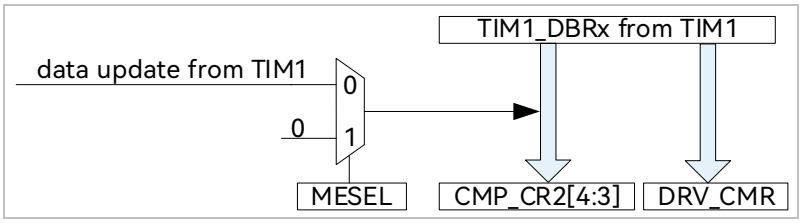
Figure 19-4 Complementary Outputs with Deadtime Insertion



19.1.2.3 Enable and Polarity of Output Signals

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

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



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

Figure 19-6 Pre-charge Waveform

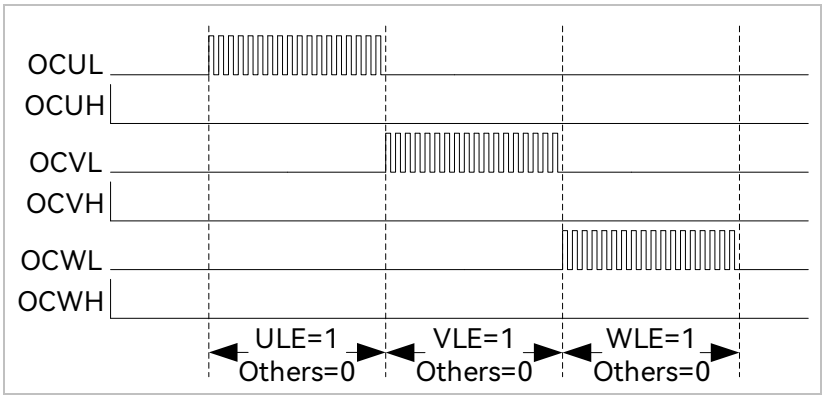
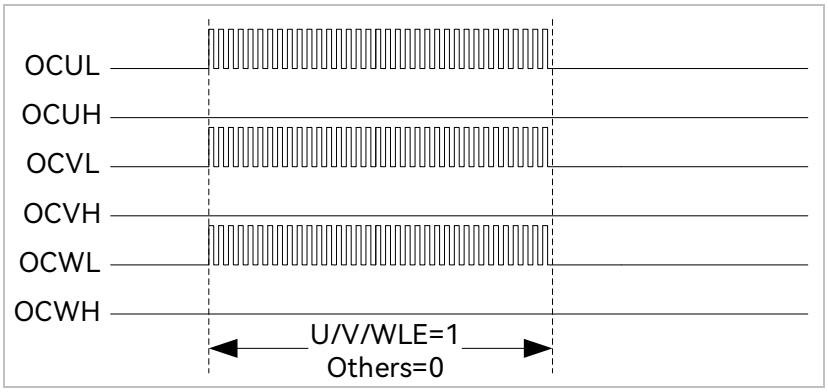


Figure 19-7 Brake Waveform



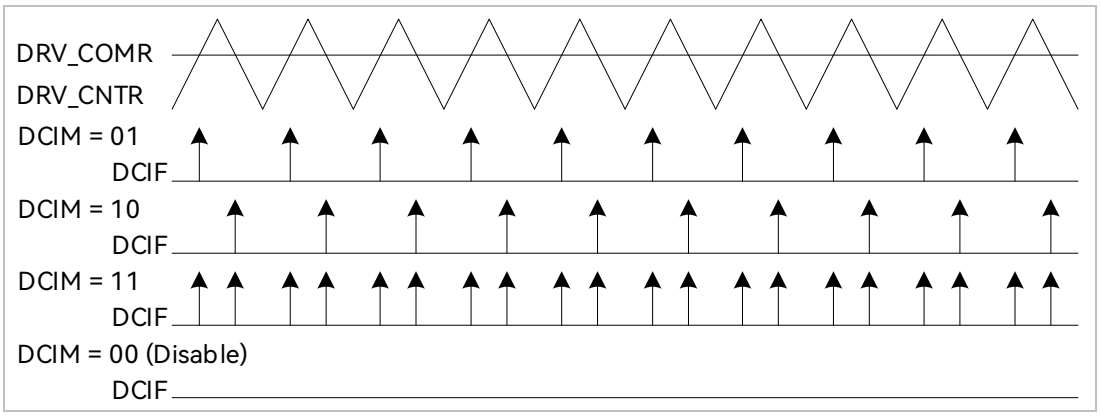
19.1.2.4 Interrupt

19.1.2.4.1 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.

Figure 19-8 Driver Compare Match Interrupt

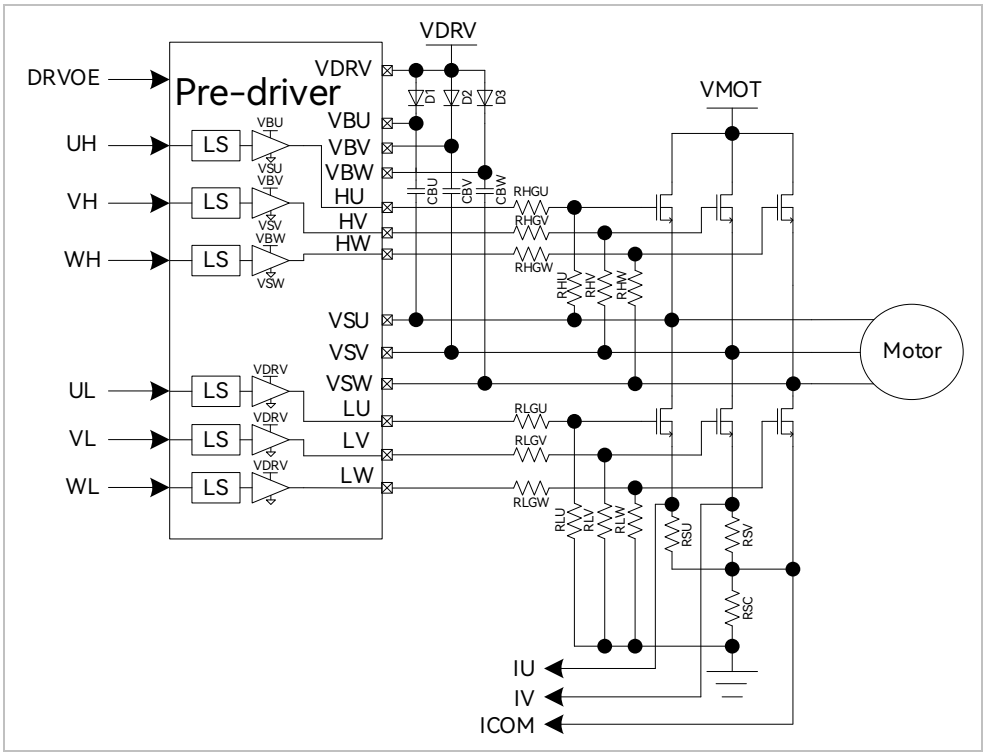


19.1.2.4.2 FG Interrupt

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

19.1.3 6N Pre-driver Mode

Figure 19-9 Block Diagram of 6N Pre-driver Mode



The features of 6N pre-driver are shown in Table 19-1. Configuring DRV_CR[DRVOE] enables pre-driver mode, where the pre-driver output is wired to 6 NMOS respectively to drive phase-U/V/W.

Table 19-1 Output Truth Values of FU6367Q Built-in Pre-driver

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

19.2 Driver Registers

19.2.1 PI_CR (0xF9)

Bit	7	6	5	4	3	2	1	0
Name	T2TSS	RSV				DRVMD	HINV	LINV
Type	R/W	-	-	-	-	R/W	R/W	R/W
Reset	0	-	-	-	-	0	0	0

Bit	Name	Description
[7]	T2TSS	Input Mode Selection of TIM2 Step Motor 0: Direction + Pulse Input Mode. P1.0 for direction input, and P0.7 for pulse input 1: Bidirectional Pulse Input Mode. P1.0 for backward pulse input, and P0.7 for forward pulse input
[6:3]	RSV	Reserved
[2]	DRVMD	Count Mode 0: Triangular Wave Mode 1: Sawtooth Wave Mode (FOC disabled)
[1]	HINV	High Side Reverse Enable 0: Disable 1: Enable
[0]	LINV	Low Side Reverse Enable 0: Disable 1: Enable

19.2.2 DRV_CR (0x4062)

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

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

19.2.3 DRV_SR (0x4061)

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

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

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

19.2.4 DRV_OUT (0xF8)

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

Bit	Name	Description
[7]	MOE	Main Output Enable This bit selects the sources of 3-phase output signals for high and low sides of the driver. It can be set to “1” and “0” by software. When bus current protection occurs, it is automatically cleared to “0” by hardware to turn off the output (see section 31.1.1.1). 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]	OISWL	Output idle level of WL/XL See descriptions on OISUH register Note: DRV_OUT[OISWL] bit is configured as WL/XL output in IDLE state
[4]	OISWH	Output idle level of WH/XH See the descriptions on OISUH register Note: DRV_OUT[OISWH] bit is configured as WH/XH output in IDLE state
[3]	OISVL	Output idle level of VL See descriptions on OISUH register
[2]	OISVH	Output idle level of VH See descriptions on OISUH register
[1]	OISUL	Output idle level of UL See descriptions on OISUH register
[0]	OISUH	Output idle level of UH This bit sets the UH output in IDLE state. When DRV_OUT[MOE] = 0, it outputs idle level to disable MOS.

		0: Low 1: High

19.2.5 DRV_CMR (0x405C, 0x405D)

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

Bit	Name	Description
[15]	XHP	High-side Polarity Control of Phase-X 0: Active High 1: Active Low
[14]	XLP	Low-side Polarity Control of Phase-X 0: Active High 1: Active Low
[13]	XHE	High-side Output of Phase-X Enable 0: Disable 1: Enable
[12]	XLE	Low-side Output of Phase-X Enable 0: Disable 1: Enable
[11]	WHP	High-side Polarity Control of Phase-W 0: Active High 1: Active Low
[10]	WLP	Low-side Polarity Control of Phase-W 0: Active High 1: Active Low
[9]	VHP	High-side Polarity Control of Phase-V 0: Active High 1: Active Low
[8]	VLP	Low-side Polarity Control of Phase-V 0: Active High 1: Active Low

[7]	UHP	High-side Polarity Control of Phase-U 0: Active High 1: Active Low
[6]	ULP	Low-side Polarity Control of Phase-U 0: Active High 1: Active Low
[5]	WHE	High-side Output of Phase-W Enable 0: Disable 1: Enable
[4]	WLE	Low-side Output of Phase-W Enable 0: Disable 1: Enable
[3]	VHE	High-side Output of Phase-V Enable 0: Disable 1: Enable
[2]	VLE	Low-side Output of Phase-V Enable 0: Disable 1: Enable
[1]	UHE	High-side Output of Phase-U Enable 0: Disable 1: Enable
[0]	ULE	Low-side Output of Phase-U Enable 0: Disable 1: Enable

Note:

- When DRV_CMR[W/V/ULE] and DRV_CMR[W/V/UHE] are set to “1”, high-side and low-side outputs of phase-W/V/U are complementary to generate PWM signals with deadtime insertion.
- For square-wave control, Timer1 automatically controls DRV_CMR register.

19.2.6 DRV_ARR (0x405E, 0x405F)

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

Bit	Name	Description
[15:14]	RSV	Reserved
[13:0]	DRV_ARR	Timer reload value, which determines PWM frequency (center-aligned) Driver timer up-counts from 0 to DRV_ARR/2 - 1 and an overflow event occurs. Then it down-counts to 0. Calculation formula: $f_{carrier}$ = 48MHz/DRV_ARR DRV_ARR value is calculated using 48MHz clock rate, which falls within the range [0, 16383]. Note: The LSB is always 0, and a write of "1" has no effect.

19.2.7 DRV_COMR (0x405A, 0x405B)

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

Bit	Name	Description
[15:12]	RSV	Reserved
[11:0]	DRV_COMR	Timer Compare-match Value The compare match interrupt is generated when the count value is equal to DRV_COMR and the conditions set in DRV_SR[DCIM] are met. The clock rate for the calculation is 12MHz. Duty cycle at the match point = DRV_COMR*4/DRV_ARR*100% DRV_COMR is calculated using 12MHz clock rate, which falls within the range [0, 4095].

19.2.8 DRV_DR (0x4058, 0x4059)

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

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

Bit	Name	Description
[15:14]	RSV	Reserved
[13:0]	DRV_DR	PWM Duty Cycle Setting in Software Duty cycle = DRV_DR/DRV_ARR*100% DRV_DR is calculated using 48MHz clock rate, which falls within the range [0, 16383]. Note: When this register is used as a comparison source, PWM is referenced to high side of the driver and a deadtime is inserted in the complementary output of low side of the driver.

19.2.9 DRV_DTR (0x4060)

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

Bit	Name	Description
[7:0]	DRV_DTR	Deadtime Setting Deadtime = (DRV_DTR + 1)*T Example: If DRV_DTR is configured to “11”, the deadtime = 12*41.67ns = 500ns. Note: If DRV_DTR is configured to “0”, deadtime insertion is disabled.

19.2.10 DRV_CNTR (0x4066, 0x4067)

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

Bit	Name	Description

[15:12]	RSV	Reserved
[11:0]	DRV_CNTR	Count Value of Timer The clock rate for the calculation is 12MHz, and Driver duty cycle = DRV_CNTR*4/ DRV_ARR*100% Range [0, 4095] Note: DRV_CNTR register can be written only when DRV_CR[DRVEN] = 1.

20 WDT

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

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

20.1 WDT Notes

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

20.2 WDT Operations

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

20.3 WDT Registers

20.3.1 WDT_CR (0x4026)

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

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

20.3.2 WDT_ARR (0x4027)

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

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

20.3.3 CCFG1 (0x401E)

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

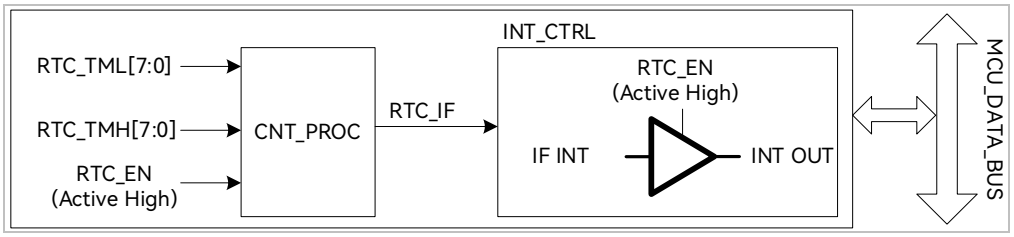
Bit	Name	Description
[7]	RSV	Reserved
[6]	LVWIE	LVW Detection Interrupt Enable 0: Disable 1: Enable
[5]	WDT_EN	WDT Enable 0: Disable 1: Enable
[4]	VBB_ENB	LDO VBB Enable 0: Enable 1: Disable

		Note: It can be configured by unchecking Vbb Disable (see Figure 32-2)
[3:1]	RSV	Reserved
[0]	FCK_MOD	System Clock Selection 0: Internal Fast Clock serves as the system clock 1: External Fast Clock serves as the system clock Note: It can be configured by checking Internal Fast Clk and External Fast Clk options (See Figure 32-2).

21 RTC

21.1 Functional Block Diagram

Figure 21-1 Functional Block Diagram of RTC Module



21.2 RTC Operations

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

21.3 RTC Registers

21.3.1 RTC_TM (0x402C, 0x402D)

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

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

21.3.2 RTC_STA (0x402E)

Bit	7	6	5	4	3	2	1	0
Name	RTC_EN	RTC_IF	ISOSCSEL	ISOSCEN	ESOAE	ESOEN	ESCLKSEL	ESCKEN
Type	R/W	R/W0	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]	RTC_EN	RTC Enable 0: Disable 1: Enable
[6]	RTC_IF	RTC Interrupt Flag This bit is set to “1” when the timer value matches RTC_TM setting. Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[5]	ISOSCSEL	Slow Clock Source Selection 0: Internal Slow Clock 1: External Slow Clock
[4]	ISOSCEN	Internal Slow Clock Enable 0: Disable 1: Enable
[3]	ESOAE	External Slow Clock Crystal Pin Enable 0: Disable 1: Enable
[2]	ESOEN	External Slow Clock Crystal Enable 0: Disable, with digital input for the external slow clock 1: Enable, with external slow clock crystal input
[1]	ESCLKSEL	External Slow Clock Digital Input Selection 0: P1.1 Input 1: P3.4 Input
[0]	ESCKEN	External Slow Clock Digital Input Enable 0: Disable 1: Enable

22 IO

22.1 IO Introduction

FU6367Q has up to 31 GPIO pins, including P0.0 ~ P0.2, P0.4 ~ P0.7, P1.0 ~ P1.7, P2.0 ~ P2.7, P3.0 ~ P3.6 and P4.5.

22.2 IO Operations

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

- The enable bits of pull-up resistors and pull-down resistors are configured to “1”. See 22.3.10 P0_PU (0x4053) ~ 22.3.14 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.
- The relevant bits of P1_AN, P2_AN, P3_AN and P4_AN registers are configured to “1”. See 22.3.6 P1_AN (0x4050) ~ 22.3.9 P4_AN(0x4047) 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, P3 and P4 registers.
- Pull-up resistors of P1.6 ~ P1.7, P2.0 ~ P2.7 and P3.0 ~ P3.5 are automatically disabled when the port pins are configured as analog mode.

22.3 IO Registers

22.3.1 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

22.3.2 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

22.3.3 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

22.3.4 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

22.3.5 P4_OE (0xE9)

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

Bit	Name	Description
[7]	RSV	Reserved
[6:1]	P4_OE	P4.1 ~ P4.6 Digital I/O Selection 0: Input 1: Output
[0]	RSV	Reserved

22.3.6 P1_AN (0x4050)

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

Bit	Name	Description
[7:4]	P1_AN	P1.4 ~ P1.7 Analog Mode Enable 0: Disable 1: Enable
[3]	HBMOD	P1.3 mode configuration, which determines the functional mode of P1.3 in combination with P1_OE[3], as shown in Table 22-1. Table 22-1 P1.3 Mode Setting

HBMOD	P1_OE[3]	P1.3 Pin Mode
0	0	Digital Input
0	1	Digital Output
1	0	Analog Mode

			1	1	Digital enhanced drive output mode. The maximum output current of high level output can be up to 20mA for Hall bias power supply. The drive mode of low level output is the same as that of the digital output mode.
[2]	RSV	Reserved			
[1]	ODE1	P0.1 Open-drain Enable 0: Disable 1: Enable			
[0]	ODE0	P0.0 Open-drain Enable 0: Disable 1: Enable			

22.3.7 P2_AN (0x4051)

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

Bit	Name	Description
[7:0]	P2_AN	P2.0 ~ P2.7 Analog Mode Enable 0: Disable 1: Enable

22.3.8 P3_AN (0x4052)

Bit	7	6	5	4	3	2	1	0
Name	P11_PL	P01_PL	P3_AN					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	P11_PL	P1.1 Pull-down Resistor Enable 0: Disable 1: Enable Note: Pull-up resistor and pull-down resistor of P1.1 cannot be enabled at the same time.
[6]	P01_PL	P0.1 Pull-down Resistor Enable 0: Disable 1: Enable

		Note: Pull-up resistor and pull-down resistor of P0.1 cannot be enabled at the same time
[5:0]	P3_AN	P3.0 ~ P3.5 Analog Mode Enable 0: Disable 1: Enable

22.3.9 P4_AN(0x4047)

Bit	7	6	5	4	3	2	1	0
Name	RSV		P45_AN	RSV				
Type	-	-	R/W	-	-	-	-	-
Reset	-	-	0	-	-	-	-	-

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	P45_AN	P4.5 Analog Mode Enable 0: Disable 1: Enable
[4:0]	RSV	Reserved

22.3.10 P0_PU (0x4053)

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

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

22.3.11 P1_PU (0x4054)

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

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

[1]	P11HV_EN	P1.1 High Voltage Input Enable 0: Disable. The reversed level is the same as that of general-purposed IO. See section 5.4 GPIO Electrical Characteristics. 1: Enable. The reversed level is related to VCC. See section 5.4 GPIO Electrical Characteristics. Note: P1.1 can withstand a voltage up to VCC regardless of the level of this bit.
[0]	P10HV_EN	P1.0 High Voltage Input Enable 0: Disable 1: Enable

22.3.12 P2_PU (0x4055)

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

Bit	Name	Description
[7:0]	P2_PU	P2.0 ~ P2.7 Pull-up Resistor Enable 0: Disable 1: Enable

22.3.13 P3_PU (0x4056)

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

Bit	Name	Description
[7:0]	P3_PU	P3.0 ~ P3.7 Pull-up Resistor Enable 0: Disable 1: Enable

22.3.14 P4_PU (0x4057)

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

Bit	Name	Description

[7]	RSV	Reserved
[6:1]	P4_PU	P4.1 ~ P4.6 Pull-up Resistor Enable 0: Disable 1: Enable
[0]	RSV	Reserved

22.3.15 PH_SEL (0x404C)

Bit	7	6	5	4	3	2	1	0
Name	SPITMOD	UART1EN	UART2EN	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]	UART1EN	Port Multiplexed as RXD, TXD and UART1 Enabled 0: Disable 1: P0.5 and P0.6 pins multiplexed as RXD and TXD pins and UART1 enabled
[5]	UART2EN	Port Multiplexed as RXD2, TXD2 and UART2 Enabled 0: Disable 1: P1.2 and P0.0 pins multiplexed as TXD2 pin; P3.6 and P0.1 pins multiplexed as RXD2 pin; and UART2 enabled.
[4]	T4SEL	Port Pins Multiplexed as Timer4 or Timer4S 0: Disable 1: P0.1 or P0.0 or P1.2 (PH_SEL1[T4CT1] = 1 and PH_SEL1[T4CT0] = 0) multiplexed as Timer4 I/O pins.
[3]	T3SEL	Port Pins Multiplexed as Timer3 or Timer3S 0: Disable 1: P1.1 or P0.1 pin multiplexed as Timer3 I/O pins
[2]	T2SEL	Port Pins Multiplexed as Timer2 0: Disable 1: P1.0 pin multiplexed as Timer2 I/O pins
[1]	T2SSEL	Port Pins Multiplexed as Timer 2S 0: Disable 1: P0.7 pin multiplexed as Timer2 I/O pins
[0]	XOE	Phase-X Output Enable 0: Disable 1: Enable, with P4.1 as the phase-X low-side PWM output pin and P4.2 as the phase-X high-side PWM output pin

22.3.16 PH_SEL1 (0x404D)

Bit	7	6	5	4	3	2	1	0
Name	UART2CH1	UART2CH0	CMPXO_P34	SPICT0	T4CT1	T4CT0	RSV	T3CT0
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
[7:6]	UART2CH	UART2/LIN Function Switching 00: P3.6 serving as RXD (P3.6 is an I/O pin of single-wire mode) 01: P1.2 serving as TXD (P1.2 is an I/O pin of single-wire mode) 1X: P0.1 serving as RXD, and P0.0 pin as TXD (P0.1 is an I/O pin in single-wire mode)
[5]	CMPXO_P34	Comparator Function Switching 0: Disable, with P0.1 serving as output 1: Enable, with P3.4 serving as output
[4]	SPICT0	Debug Function Switching of SPI in Single-wire Mode This bit works with PH_SEL2[SPICT1]. PH_SEL2[SPICT1] : PH_SEL1[SPICT0] 00: Disable, with P0.5 serving as SPI debug output X1: Enable, with P0.0 serving as SPI debug output 10: Enable, with P3.4 serving as SPI debug output
[3:2]	T4CT	Timer4 Function Switching 00: Timer4 I/O pin switched to P0.1 X1: Timer4 I/O pin switched to P0.0 10: Timer4 I/O pin switched to P1.2
[1]	RSV	Reserved
[0]	T3CT0	Timer3 Function Switching 0: Timer3 I/O pin switched to P1.1 1: Timer3 I/O pin switched to P0.1

22.3.17 PH_SEL2 (0x4049)

Bit	7	6	5	4	3	2	1	0
Name	RSV		ADC_SCYCH[7:4]				I2C_FS	SPICT1
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
[5:2]	ADC_SCYCH[7:4]	See descriptions on PH_SEL2 (0x4049) in chapter ADC.

[1]	I2C_FS	I²C Function Switching 0: Disable, with P0.0 as I²C SDA pin, and P0.1 serving as I²C SCL pin 1: Enable, with P0.5 as I²C SDA pin, and P0.6 serving as I²C SCL pin
[0]	SPICT1	Debug Function Switching of SPI in Single-wire Mode This bit works with PH_SEL1[SPICT0]. PH_SEL2[SPICT1] : PH_SEL1[SPICT0] 00: Disable, with P0.5 serving as SPI debug output X1: Enable, with P0.0 serving as SPI debug output 10: Enable, with P3.4 serving as SPI debug output

22.3.18 P0 (0x80)

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

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

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

22.3.19 P1 (0x90)

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

Bit	Name	Description
[7]	GP17	GP17 Pin
[6]	GP16	GP16 Pin
[5]	GP15	GP15 Pin
[4]	GP14	GP14 Pin
[3]	GP13	GP13 Pin
[2]	GP12	GP12 Pin

[1]	GP11	GP11 Pin
[0]	GP10	GP10 Pin

22.3.20 P2 (0xA0)

Bit	7	6	5	4	3	2	1	0
Name	GP27	GP26	GP25	GP24	GP23	GP22	GP21	GP20
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP27	GP27 Pin
[6]	GP26	GP26 Pin
[5]	GP25	GP25 Pin
[4]	GP24	GP24 Pin
[3]	GP23	GP23 Pin
[2]	GP22	GP22 Pin
[1]	GP21	GP21 Pin
[0]	GP20	GP20 Pin

22.3.21 P3 (0xB0)

Bit	7	6	5	4	3	2	1	0
Name	GP37	GP36	GP35	GP34	GP33	GP32	GP31	GP30
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	GP37	GP37 Pin
[6]	GP36	GP36 Pin
[5]	GP35	GP35 Pin
[4]	GP34	GP34 Pin
[3]	GP33	GP33 Pin
[2]	GP32	GP32 Pin
[1]	GP31	GP31 Pin
[0]	GP30	GP30 Pin

22.3.22 P4 (0xB8)

Bit	7	6	5	4	3	2	1	0
Name	RSV	GP46	GP45	GP44	GP43	GP42	GP41	RSV
Type	-	R/W	R/W	R/W	R/W	R/W	R/W	-
Reset	-	0	0	0	0	0	0	-

Bit	Name	Description
[7]	RSV	Reserved
[6]	GP46	GP46 Pin
[5]	GP45	GP45 Pin
[4]	GP44	GP44 Pin
[3]	GP43	GP43 Pin
[2]	GP42	GP42 Pin
[1]	GP41	GP41 Pin
[0]	RSV	Reserved

Table 22-2 RMW Commands

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

23 System Clock

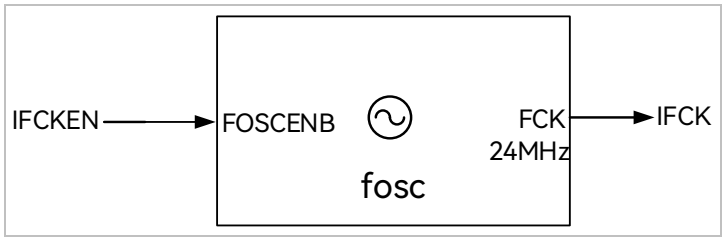
23.1 Introduction

The clock consists of four modules: Internal Fast Clock, External Fast Clock, Internal Slow Clock and External Slow Clock. The system clock operates either as an Internal Fast Clock or an External Fast Clock, which is selected by the associated register or switched by software program. External Fast Clock is configured as external clock input or external crystal oscillator source, and Internal Slow Clock is used as WDT for configuring overflow time of the watchdog. External Slow Clock is used for RTC counts.

23.2 Clock Operations

23.2.1 Internal Fast Clock

Figure 23-1 Input/Output of Internal Fast Clock



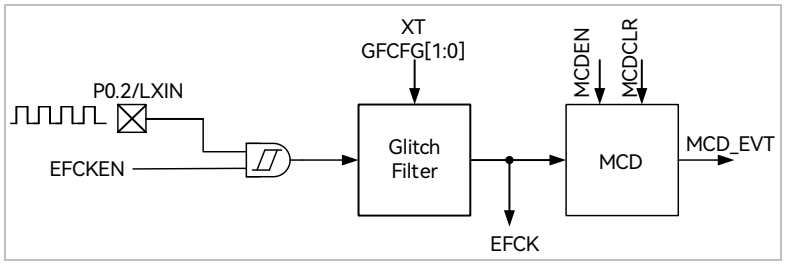
The I/O pins of the Internal Fast Clock are shown in Figure 23-1. Internal Fast Clock is used to produce an accurate clock running at a rate of 24MHz. It is enabled after Internal Fast Clk option in Figure 23-5 is checked or OSC_CFG[IFCKEN] is set to “1”. In Sleep mode (PCON[STOP] = 1), Internal Fast Clock is disabled.

23.2.2 External Fast Clock

External fast clock supports external clock input mode.

23.2.2.1 External Clock Input Mode

Figure 23-2 External Clock Input Mode of External Fast Clock



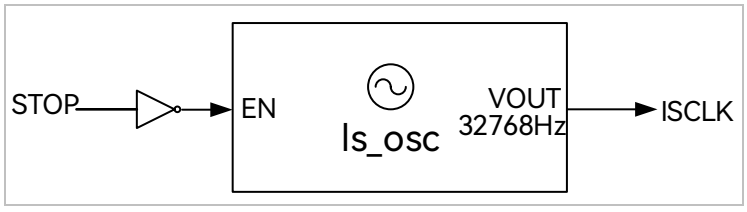
External Clock Input Mode of the External Fast Clock is shown in Figure 23-2. External clock is input through the pad port of LXIN. External Fast Clock is enabled after External Fast Clk option in Figure 23-5 is checked or OSC_CFG[EFCKEN] is set to “1”. When Oscillator Mode is checked, External Fast Clock operates in the External Clock Input Mode.

CLR_OST is the signal to clear OST timer (active High). When the transistor circuit runs in Crystal Input Mode, OST timer counts 1024 clocks and EFCK outputs clock signals after it stabilizes.

OST does not work when transistor circuit operates in External Clock Input Mode. OST timer is automatically cleared to “0” by hardware when transistor circuit is not enabled. When the clock is missing, the system automatically switches the clock to the internal clock, and then the software switches the clock to the external clock again. At this time, it needs to send a high level to CLR_OST to clear OST timer.

23.2.3 Internal Slow Clock

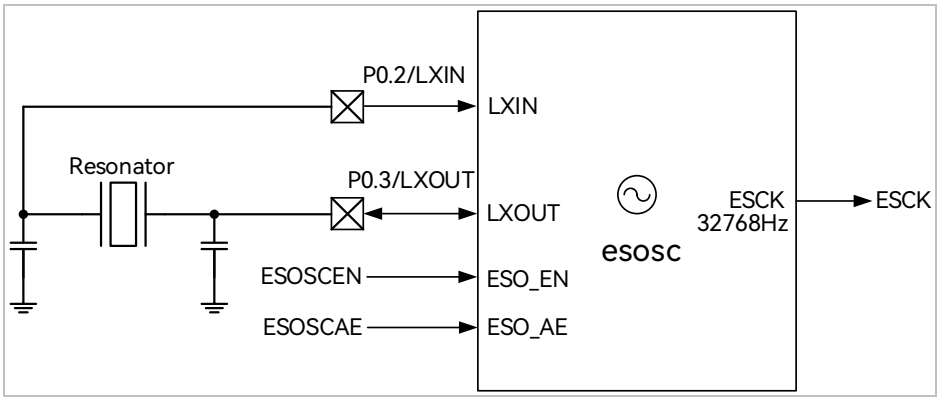
Figure 23-3 Input/Output of Internal Slow Clock on ISOSC Pin



The I/O pins of the Internal Slow Clock are shown in Figure 23-3. Internal Slow Clock is used to produce clock signals at a rate of 32768Hz. In Sleep mode (PCON[STOP] = 1), Internal Slow Clock is disabled.

23.2.4 External Slow Clock

Figure 23-4 Input/Output of External Slow Clock on esosc Pin



The I/O pins of the External Slow Clock are shown in Figure 23-4. External Slow Clock is used to produce clock signals with a rate of 32768Hz. It is enabled when RTC_STA[ESOEN] = 1 and RTC_STA[ESOAE] = 1.

23.3 MCD Module and Two-speed Mode

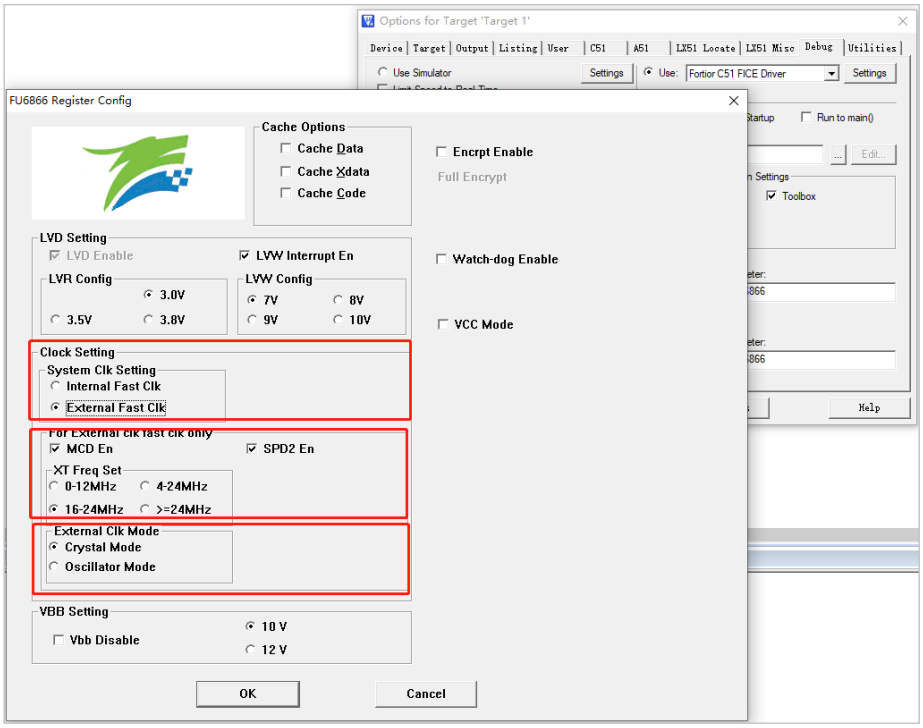
MCD (Missing Clock Detect) is used to detect the missing of the clock. When the system is running in Crystal Input Mode, the External Fast Clock is easily disturbed by various noises, and strong interference may even cause oscillation stop. In this case, MCD module is used to reduce such adverse effects on the system.

MCD En in Figure 23-5 is checked when the system is running with External Fast Clock. If the external crystal stops vibrating, the chip automatically detects the clock missing event, and enables and switches to the internal fast clock. When IE[MCDIE] = 1 (interrupt enable is set in advance), the system sends MCD interrupt after switching to internal fast clock. The software clears the MCD interrupt flag bit TCON[MCDIF] in the MCD interrupt service program, and reads OSC_CFG[CK_FLAG/MCD_RET] to determine whether the current system clock is running with an internal fast clock or an external fast clock. If the external fast clock is still required at this time, OSC_CFG[CK_FLAG/MCD_RET] must be set to “1”, that is, the system automatically switches from Internal Fast Clock mode back to External Fast Clock mode, and enables the External Fast Clock again.

Note:

SPD2 En in Figure 23-5 is checked to set the clock to two-speed mode, which means that during clock switching, the original clock keeps working until the target clock works stably, so as to reduce the impact of clock excitation/switching failure on system reliability. If SPD2 EN is not checked, the system clock directly switches to External Fast Clock, regardless of whether the external fast clock exists.

Figure 23-5 FU6367Q System Clock Configurations



23.4 Clock Registers

23.4.1 OSC_CFG (0xF1)

Bit	7	6	5	4	3	2	1	0
Name	EFCKEN	IFCKEN	CK_FLAG/ MCD_RET	RSV				
Type	R/W	R/W	R/W	-	-	-	-	-
Reset	0	0	0	-	-	-	-	-

Bit	Name	Description
[7]	EFCKEN	External Fast Clock Enable 0: Disable 1: Enable
[6]	IFCKEN	Internal Fast Clock Enable 0: Disable 1: Enable
[5]	CK_FLAG/ MCD_RET	Read: System Clock Flag 0: System operates in Internal Fast Clock mode 1: System operates in External Fast Clock mode Note: When CKMOD is configured, the system clock may not switch immediately. It may be affected by the two-speed mode and clock missing status. The CK_FLAG flag correctly reflects the system clock state. Write: 0: No effect 1: Exit the Internal Fast Clock mode that is forced to be activated due to the missing of External Fast Clock
[4:0]	RSV	Reserved

23.4.2 CCFG3 (0x401C)

Bit	7	6	5	4	3	2	1	0
Name	MCD_EN	SPD2	RSV				EC_MOD	RSV
Type	R/W	R/W	-	-	-	-	R/W	-
Reset	0	0	-	-	-	-	0	-

Bit	Name	Description
[7]	MCD_EN	MCD Enable 0: Disable 1: Enable

		Note: MCE En in Figure 23-5 is checked for the configuration.
[6]	SPD2	Two-speed Mode Enable 0: Disable 1: Enable Note: SPD2 En in Figure 23-5 is checked for the configuration.
[5:2]	RSV	Reserved
[1]	EC_MOD	Transistor Circuit Mode Selection 0: Crystal Input Mode 1: External Clock Input Mode Note: Crystal Mode and Oscillator Mode in Figure 23-5 are checked for the configuration.
[0]	RSV	Reserved

23.5 Clock Calibration

23.5.1 Introduction

Clock calibration is a feature that uses internal slow clock to calibrate the internal fast clock. Working principles: A 13-bit counter 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.

23.5.2 Clock Calibration Registers

23.5.2.1 CAL_CR0 (0x4044) CAL_CR1 (0x4045)

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

Bit	Name	Description
[15]	CAL_STA/ CAL_BUSY	Clock Calibration Enable Read: 0: Calibration is completed. 1: Calibration is in progress. Write: 0: No effect 1: Clock Calibration starts
[14:13]	RSV	Reserved
[12:0]	CAL_ARR	Calibration Counts The count values of the fast clock to continuously count eight slow clock cycles Note: When this value is 0, it indicates that no corresponding slow clock input, and when this value is 0x1FFF, it indicates that a count overflows (slow clock is too slow or fast clock is too fast).

24 Temperature Sensor

The chip integrates temperature sensor. TSD_CR[TSEN_EN] is configured as “1” to enable temperature sensor, read the value in register, and view the table to obtain current internal temperature of the chip. The sensed temperature ranges from 70℃ ~ 150℃, with its mapping code shown in Table 24-1.

Table 24-1 Mapping between Output Codes and Temperature Levels

TSEN_DR	Temperature (℃)	TSEN_DR	Temperature (℃)
000000	≤ 70	100000	110
000001	71	100001	111
000010	72	100010	112
000011	73	100011	114
000100	74	100100	115
000101	75	100101	117
000110	76	100110	118
000111	77	100111	120
001000	79	101000	121
001001	80	101001	123
001010	81	101010	124
001011	82	101011	126
001100	83	101100	128
001101	84	101101	129
001110	86	101110	131
001111	87	101111	132
010000	88	110000	134
010001	89	110001	136
010010	91	110010	137
010011	92	110011	139
010100	93	110100	141
010101	95	110101	142
010110	96	110110	144
010111	97	110111	146
011000	99	111000	148
011001	100	111001	149
011010	101	111010 ~ 111111	> 150
011011	102		

TSEN_DR	Temperature (℃)	TSEN_DR	Temperature (℃)
011100	104		
011101	105		
011110	107		
011111	108		

24.1 Temperature Sensor Registers

24.1.1 TSD_CR (0x402F)

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

Bit	Name	Description
[7]	RSV	Reserved
[6]	TSEN_EN	Temperature Sensor Enable 0: Disable 1: Enable
[5]	TSEN_HYS	Filter Width of Temperature Sensor Output 0: Filter width is 1 code value 1: Filter width is 2 code value
[4:0]	RSV	Reserved

24.1.2 TSEN_DR (0x4048)

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

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	TSEN_DR	Temperature Sensor Output Data Read the register value, and find out the temperature of the chip according to Table 24-1.

25 ADC

25.1 ADC Introduction

The ADC module is a 12-bit successive approximation register ADC with 16 channels inside. The sampling mode supports sequential sampling (that is, from ADC Channel 0 to ADC channel 15 in sequence) and trigger sampling (including FOC triggered sampling mode and Timer1 triggered sampling mode). The results of sequential sampling are stored in ADCx_DR (x = 0 ~ 15) in a right-aligned or left-second-highest-bit-aligned format. The result of triggered sampling is sent to FOC module or Timer1 module instead of ADCx_DR for motor control. The relevant registers of the FOC module or Timer1 module are always left-second-highest-bit-aligned to store the triggered sample results. Triggered sampling is done automatically by hardware, and sequential sampling is controlled by software. The priority of triggered sampling is higher than that of sequential sampling. If both triggered sampling and sequential sampling are applied at the same time, the triggered sampling is performed first, and ADC automatically stores sequential sampling mode upon completion of triggered sampling.

The clock source of ADC is at a rate of 12MHz and the sampling time is set by ADC_SCYC, PH_SEL2(0x4049) and DAC_CR[5:2]. See ADC Electrical Characteristics for sample time and conversion time.

25.2 ADC Block Diagram

Figure 25-1 ADC Multiplexer Block Diagram

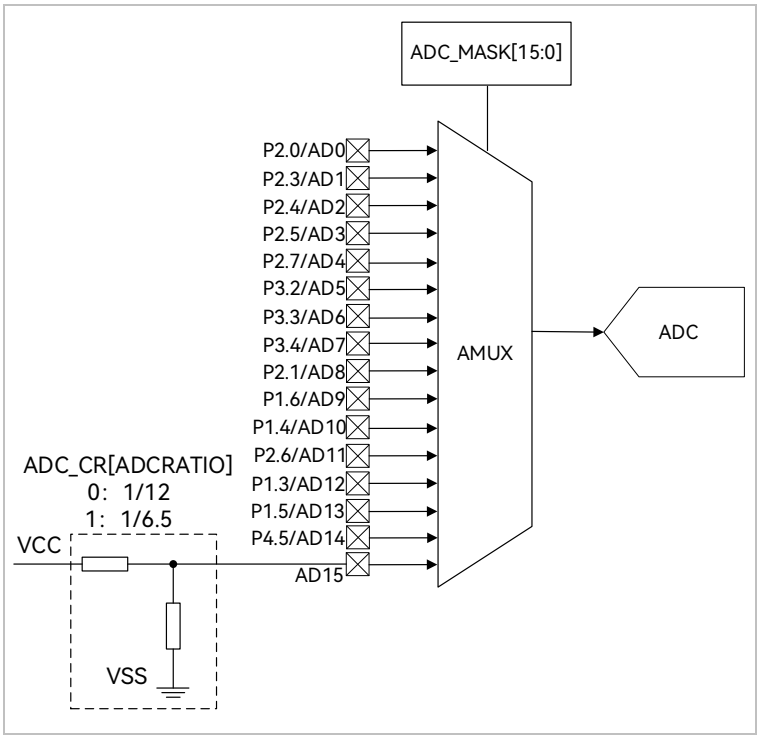
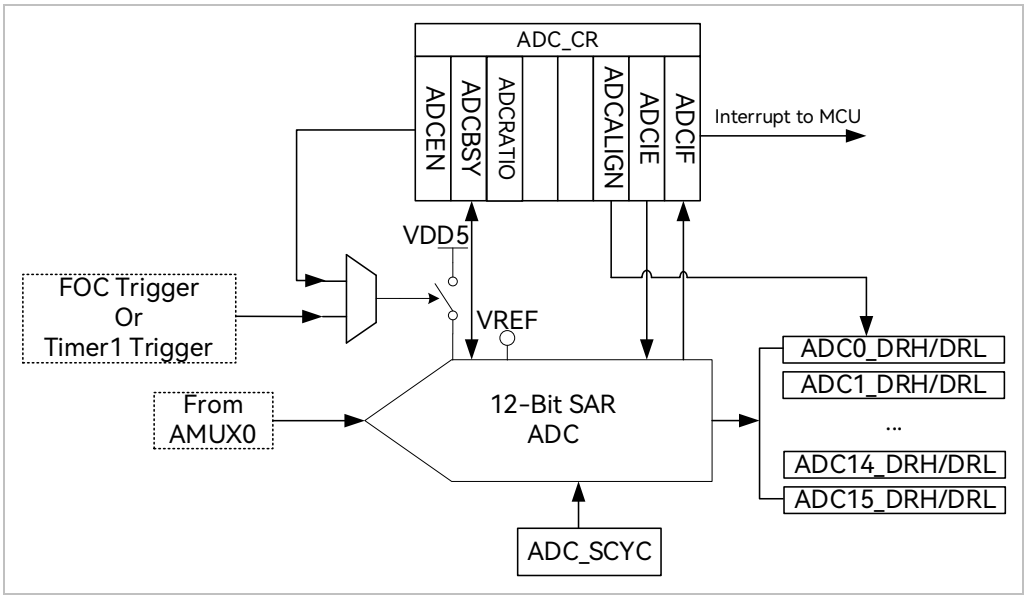


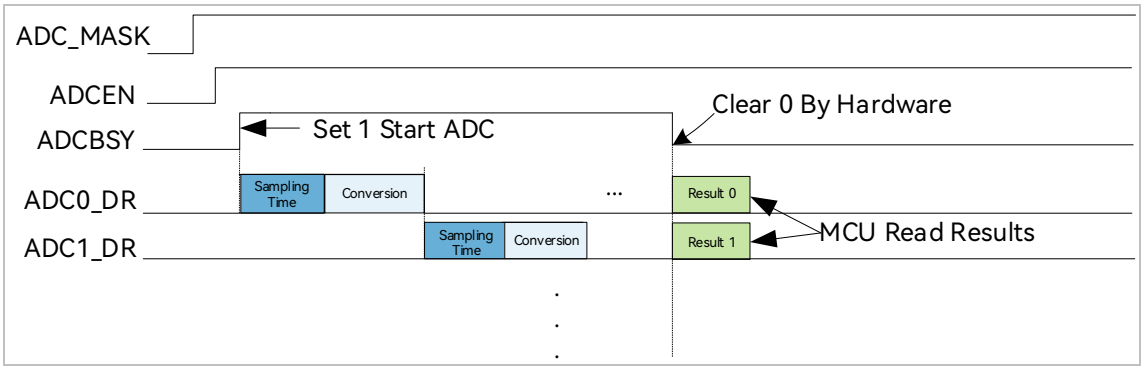
Figure 25-2 Functional Block Diagram of ADC Module



25.3 ADC Operations

25.3.1 Sequential Sampling Mode

Figure 25-3 ADC Sequential Sampling Timing



ADC operations:

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

Note:

The ADC conversion sequence is from low to high based on the enabled channel (i.e., when channel 2/3/4 is enabled, the signal is sampled in order of 2/3/4, and a single conversion result is read after confirming ADC_CR[ADCBSY] = 0).

25.3.2 Triggered Sampling Mode

When FOC module is enabled, ADC channel 0/1/2/4/15 can be used to FOC triggered sampling. Channel 2 or channel 15 is selected by FOC_CR0[UCSEL] for bus voltage triggered sampling. In single-shunt current sampling mode, channel 4 is used for itrip sampling. In dual-shunt current sampling mode, channel 0 is used for ia sampling and channel 1 for ib sampling. In triple-shunt current sampling mode, channel 0 is used for is ia sampling, channel 1 for ib sampling, and channel 4 for ic sampling.

When timer1 is enabled, channel 4 is used for bus current triggered sampling. TIM1_CR3[T1TIS] is configured to select ADC as the input source of position detection. When CMP0_CR4[CMP0FS] = 0, channel 10 is used for phase-U voltage sampling, channel 9 for phase-V voltage sampling, and channel 8 for phase-W voltage sampling. When CMP0_CR4[CMP0FS] = 1, channel 10 is used for phase-U voltage sampling, channel 12 for phase-V voltage sampling, and channel 13 for phase-W voltage sampling.

25.3.3 Output Data Format

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

Table 25-1 Relation between Output Voltage and Result Data

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

25.4 ADC Registers

25.4.1 ADC_CR (0x4039)

Bit	7	6	5	4	3	2	1	0
Name	ADCEN	ADCBSY	ADCRATIO	RSV		ADCALIGN	ADCIE	ADCIF
Type	R/W	R/W1	R/W	-	-	R/W	R/W	R/W0
Reset	0	0	0	-	-	0	0	0

Bit	Name	Description
[7]	ADCEN	ADC Enable 0: Disable 1: Enable
[6]	ADCBSY	ADC Start & ADC Busy Flag Read: 0: ADC is idle 1: ADC is busy Write: 0: No effect 1: ADC conversion starts Note: Writing “1” to this bit has no effect when ADC_MASK = 0
[5]	ADCRATIO	Division Ratio of VCC Sampling by ADC Channel 15 0: 1/12 1: 1/6.5
[4:3]	RSV	Reserved
[2]	ADCALIGN	ADC Data Format Selection 0: ADC output is right-aligned, and ADC result = ADCx_DR[11:0] 1: ADC output is left-second-highest-bit-aligned, and ADC result = ADCx_DR[14:3] Note: The results of triggered sampling mode are always left-second-highest-bit-aligned
[1]	ADCIE	ADC Interrupt Enable (excluding triggered sampling mode interrupt) 0: Disable 1: Enable
[0]	ADCIF	ADC Interrupt Flag This bit is set to “1” by hardware when ADC conversion is completed Read: 0: No Interrupt Pending 1: Interrupt Pending

		Write: 0: This bit is cleared to “0” 1: No effect

25.4.2 ADC_MASK (0x4036, 0x4037)

ADC_MASKH(0x4036)								
Bit	15	14	13	12	11	10	9	8
Name	CH15EN	CH14EN	CH13EN	CH12EN	CH11EN	CH10EN	CH9EN	CH8EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
ADC_MASKL(0x4037)								
Bit	7	6	5	4	3	2	1	0
Name	CH7EN	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15]	CH15EN	ADC Channel 15 Enable
[14]	CH14EN	ADC Channel 14 Enable
[13]	CH13EN	ADC Channel 13 Enable
[12]	CH12EN	ADC Channel 12 Enable
[11]	CH11EN	ADC Channel 11 Enable
[10]	CH10EN	ADC Channel 10 Enable
[9]	CH9EN	ADC Channel 9 Enable
[8]	CH8EN	ADC Channel 8 Enable
[7]	CH7EN	ADC Channel 7 Enable
[6]	CH6EN	ADC Channel 6 Enable
[5]	CH5EN	ADC Channel 5 Enable
[4]	CH4EN	ADC Channel 4 Enable
[3]	CH3EN	ADC Channel 3 Enable
[2]	CH2EN	ADC Channel 2 Enable
[1]	CH1EN	ADC Channel 1 Enable
[0]	CH0EN	ADC Channel 0 Enable

Note:

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

25.4.3 PH_SEL2 (0x4049)

Bit	7	6	5	4	3	2	1	0
Name	RSV		ADC_SCYCH[7:4]		I2C_FS		SPICT1	
Type	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	1	1	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:4]	ADC_SCYCH [7:4]	ADC Sampling Cycle for ADC Channel 2 and 15 ADC_SCYCH[7] = 0: The sampling cycle is ADC_SCYCH[6:4] ADC clock cycles ADC_SCYCH[7] = 1: The sampling cycle is (ADC_SCYCH[6:4]*8 + 7) ADC clock cycles
[3:2]	I2C_FS	See section PH_SEL2 (0x4049) in chapter IO
[1:0]	SPICT1	See section PH_SEL2 (0x4049) in chapter IO

25.4.4 DAC_CR (0x4035)

Bit	7	6	5	4	3	2	1	0
Name	DAC0_1EN	DACMOD	ADC_SCYCH[3:0]				DAC2EN	RSV
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
Reset	0	0	0	0	1	1	0	-

Bit	Name	Description
[7]	DAC0_1EN	See section DAC_CR (0x4035) in chapter DAC
[6]	DACMOD	See section DAC_CR (0x4035) in chapter DAC
[5:2]	ADC_SCYCH [3:0]	ADC Sampling Cycle for ADC Channel 8 ~ 13 ADC_SCYCH[3] = 0: The sampling cycle is ADC_SCYCH[2:0] ADC clock cycles ADC_SCYCH[3] = 1: The sampling cycle is (ADC_SCYCH[2:0]*8 + 7) ADC clock cycles
[1]	DAC2EN	See section DAC_CR (0x4035) in chapter DAC
[0]	RSV	Reserved

25.4.5 ADC_SCYC (0x4038)

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

Bit	Name	Description
[7:4]	ADC_SCYC [7:4]	ADC Sampling Cycle for ADC Channel 5 ~ 7 and 14 ADC_SCYC[7] = 0: The sampling cycle is ADC_SCYC[6:4] ADC clock cycles

		ADC_SCYC[7] = 1: The sampling cycle is (ADC_SCYC[6:4]*8 + 7) ADC clock cycles
[3:0]	ADC_SCYC [3:0]	ADC Sampling Cycle for ADC Channel 0, 1, 3 and 4 ADC_SCYC[3] = 0: The sampling cycle is ADC_SCYC[2:0] ADC clock cycles ADC_SCYC[3] = 1: The sampling cycle is (ADC_SCYC[2:0]*8 + 7) ADC clock cycles

25.4.6 ADC0_DR (0x0FD8, 0x0FD9)

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

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

25.4.7 ADC1_DR (0x0FDA, 0x0FDB)

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

Bit	Name	Description

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

25.4.8 ADC2_DR (0x0FDC, 0x0FDD)

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

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

25.4.9 ADC3_DR (0x0FDE, 0x0FDF)

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

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

25.4.10 ADC4_DR (0x0FE0, 0x0FE1)

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

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

25.4.11 ADC5_DR (0x0FE2, 0x0FE3)

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

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

25.4.12 ADC6_DR (0x0FE4, 0x0FE5)

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

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

25.4.13 ADC7_DR (0x0FE6, 0x0FE7)

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

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

25.4.14 ADC8_DR (0x0FE8, 0x0FE9)

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

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

25.4.15 ADC9_DR (0x0FEA, 0x0FEB)

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

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

25.4.16 ADC10_DR (0x0FEC, 0x0FED)

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

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

25.4.17 ADC11_DR (0x0FEE, 0x0FEF)

ADC11_DRH(0x0FEE)								
Bit	15	14	13	12	11	10	9	8
Name	ADC11_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC11_DRL(0x0FEF)								
Bit	7	6	5	4	3	2	1	0
Name	ADC11_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC11_DR	The conversion results of ADC channel 11 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.18 ADC12_DR (0x0FF0, 0x0FF1)

ADC12_DRH(0x0FF0)								
Bit	15	14	13	12	11	10	9	8
Name	ADC12_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC12_DRL(0x0FF1)								
Bit	7	6	5	4	3	2	1	0
Name	ADC12_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC12_DR	The conversion results of ADC channel 12 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.19 ADC13_DR (0x0FF2, 0x0FF3)

ADC13_DRH(0x0FF2)								
Bit	15	14	13	12	11	10	9	8
Name	ADC13_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC13_DRL(0x0FF3)								
Bit	7	6	5	4	3	2	1	0
Name	ADC13_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC13_DR	The conversion results of ADC channel 13 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.20 ADC14_DR (0x0FF4, 0x0FF5)

ADC14_DRH(0x0FF4)								
Bit	15	14	13	12	11	10	9	8
Name	ADC14_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC14_DRL(0x0FF5)								
Bit	7	6	5	4	3	2	1	0
Name	ADC14_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC14_DR	The conversion results of ADC channel 14 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

25.4.21 ADC15_DR (0x0FF6, 0x0FF7)

ADC15_DRH(0x0FF6)								
Bit	15	14	13	12	11	10	9	8
Name	ADC15_DR[15:8]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
ADC15_DRL(0x0FF7)								
Bit	7	6	5	4	3	2	1	0
Name	ADC15_DR[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[15:0]	ADC15_DR	The conversion results of ADC channel 15 upon completion of ADC conversion in the Sequential Sampling Mode. The data is aligned according to ADC_CR[ADCALIGN]. Note: ADC results of Triggered Sampling Mode are not updated to this register.

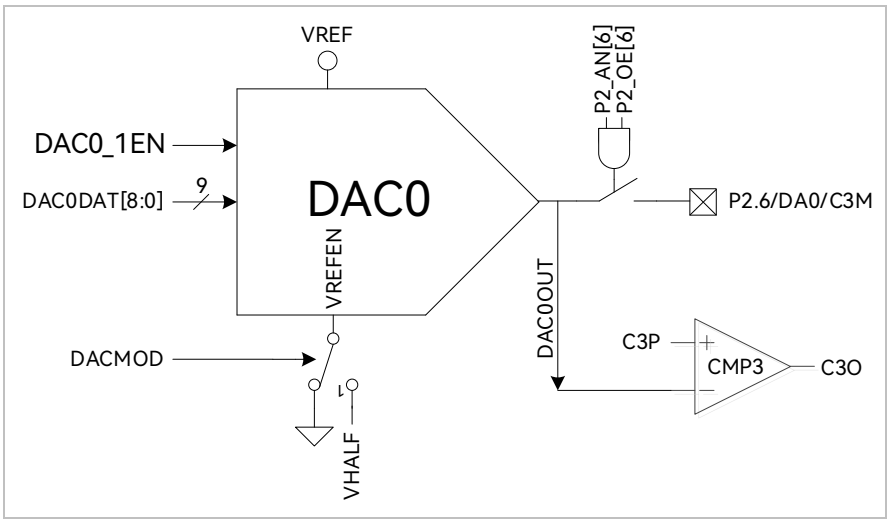
26 DAC

26.1 DAC Introduction

The chip integrates two DAC modules, where DAC0 is a 9-bit digital-to-analog converter and DAC1 is a 6-bit digital-to-analog converter.

26.2 DAC0

Figure 26-1 Functional Block Diagram of DAC0 Module



As shown in Figure 26-1, DAC0 converts 9-bit digital data into analog voltage and sends the voltage to CMP3 negative input of for bus overcurrent protection. P2.6 pin can be configured as the analog output.

Note:

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.

DAC0 operations are as follows:

1. Configure P2_AN[6] = 1 and P2_OE[6] = 1, and DAC0 outputs to P2.6 pin;
2. Configure VREF_VHALF_CR[VREFEN] = 1 and DAC_CR[DAC0_1EN] = 1, and VREF is used as DAC0 reference voltage;

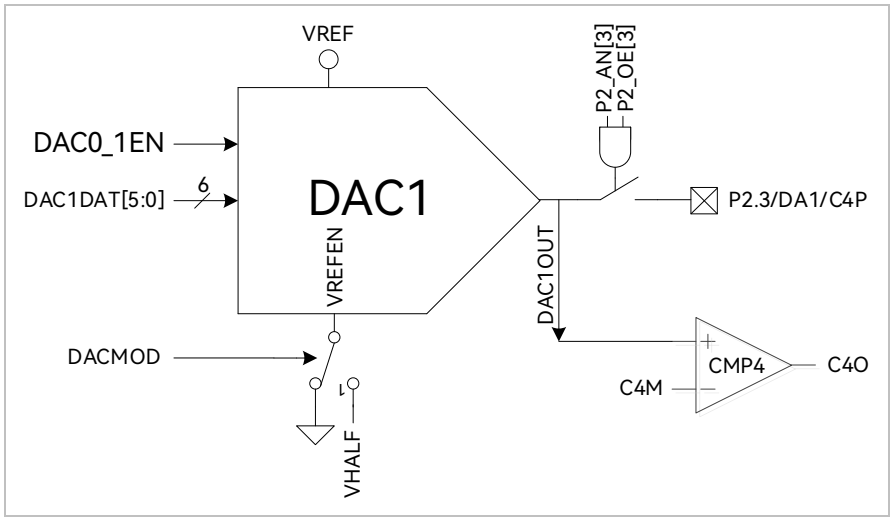
3. The range of output voltage is set by DAC_CR[DACMOD]. When DAC_CR[DACMOD] = 0, full-voltage output mode is active, and the range of output voltage is 0 ~ VREF. When DAC_CR[DACMOD] = 1, half-voltage output mode is active, the range of output voltage is VHALF ~ VREF. Output voltage of DAC0DAT under different configurations is shown in Table 26-1.

Table 26-1 Voltage Output of DAC0 under Different Configurations

DAC0DAT[8:0]	DAC Output Voltage (DAC_CR[DACMOD] = 0)	DAC Output Voltage (DAC_CR[DACMOD] = 1)
0x000	0	VHALF
0x100	VREF/2	(VREF - VHALF)/2 + VHALF
0x1FF	VREF*511/512	(VREF - VHALF)*511/512 + VHALF

26.3 DAC1

Figure 26-2 Functional Block Diagram of DAC1 Module



As shown in Figure 26-2, DAC1 converts 6-bit digital data into analog voltage, and sends the voltage to CMP4 positive input for cycle-by-cycle current limiting. P2.3 pin can be configured as the analog output.

Note:

DAC1 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.

DAC1 operations are as follows:

1. Configure P2_AN[3] = 1 and P2_OE[3] = 1, and DAC1 outputs to P2.3 pin;

2. Configure VREF_VHALF_CR[VREFEN] = 1 and DAC_CR[DAC0_1EN] = 1, and VREF is used as DAC1

reference voltage;

3. The range of output voltage is set by DAC_CR[DACMOD]. When DAC_CR[DACMOD] = 0, full-voltage output mode is active, and the range of output voltage is 0 ~ VREF. When DAC_CR[DACMOD] = 1, half-voltage output mode is active, and the range of output voltage is VHALF ~ VREF. Output voltage of DAC1 under different configurations is shown in Table 26-2.

Table 26-2 DAC1 Voltage Output under Different Configurations

DAC1DAT[5:0]	DAC Output Voltage (DAC_CR[DACMOD] = 0)	DAC Output Voltage (DAC_CR[DACMOD] = 1)
0x00	0	VHALF
0x20	VREF/2	(VREF - VHALF)/2 + VHALF
0x3F	VREF*63/64	(VREF - VHALF)*63/64 + VHALF

26.4 DAC Registers

26.4.1 DAC_CR (0x4035)

Bit	7	6	5	4	3	2	1	0
Name	DAC0_1EN	DACMOD	ADC_SCYCH[3:0]				RSV	
Type	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	1	1	-	-

Bit	Name	Description
[7]	DAC0_1EN	DAC0 and DAC1 Enable 0: Disable 1: Enable
[6]	DACMOD	DAC Mode Setting 0: Full-voltage Output Mode 1: Half-voltage Output Mode
[5:2]	ADC_SCYCH[3:0]	See section DAC_CR (0x4035) in chapter ADC
[1:0]	RSV	Reserved

26.4.2 DAC0_DR (0x404B)

Bit	7	6	5	4	3	2	1	0
Name	DAC0DAT[8:1]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:0]	DAC0DAT[8:1]	8 high-order bits input of DAC0 controller

26.4.3 DAC1_DR (0x404A)

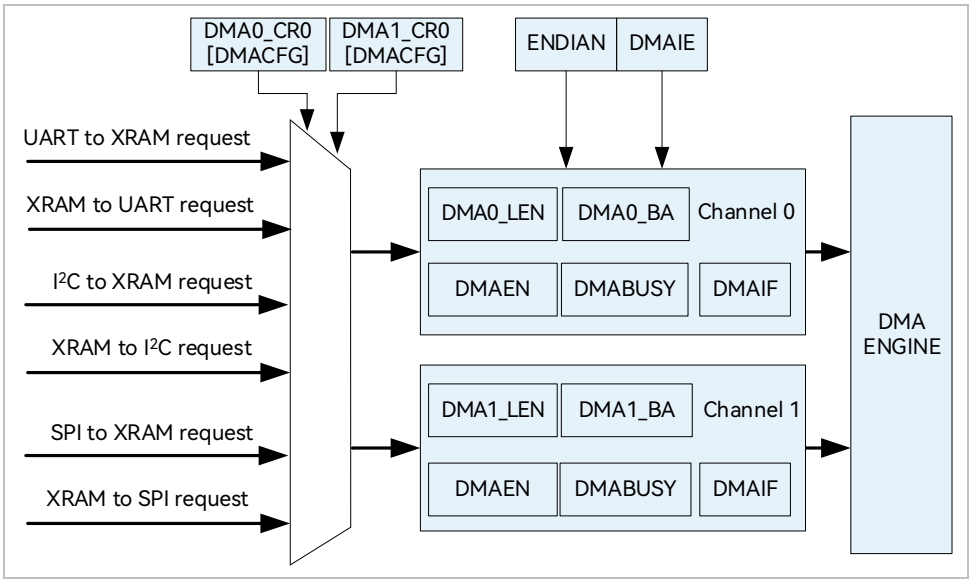
Bit	7	6	5	4	3	2	1	0
Name	DAC0_DR_0	RSV	DAC1DAT					
Type	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	0

Bit	Name	Description
[7]	DAC0_DR_0	LSB input of DAC0 controller
[6]	RSV	Reserved
[5:0]	DAC1DAT	6-bit data input of DAC1 controller

27 DMA

27.1 DMA Instructions

Figure 27-1 Functional Block Diagram of DMA Module



The DMA module is a dual-channel DMA controller, which performs direct data transfer between peripherals (SPI, UART, I²C) and XRAM (IRAM data invalid). DMA accessing to XRAM does not interfere with the normal CPU read/write operation to XRAM. The length of the transferred data and the start address of XRAM access is configurable. Data transfer mode is configurable and interrupt can be enabled.

DMA operations are as follows

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

27.2 DMA Registers

27.2.1 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	DMA Channel 0 Peripherals and Transfer Direction Selection 000: From UART1 to XRAM 001: From XRAM to UART1 010: From I²C to XRAM 011: From XRAM to I²C 100: From SPI to XRAM 101: From XRAM to SPI 110: From UART2 to XRAM 111: From XRAM to UART2 Note: It cannot be configured when channel 0 is busy. LIN multiplexes UART2 channels, and the read/write direction is determined by LIN_CR[LINRW].
[2]	DMAIE	DMA Channel 0 Interrupt Enable 0: Disable 1: Enable
[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 Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: The interrupt event is generated

27.2.2 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 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	DMA Channel 1 Peripherals and Direction Selection 000: From UART1 to XRAM 001: From XRAM to UART1 010: From I^2C to XRAM 011: From XRAM to I^2C 100: From SPI to XRAM 101: From XRAM to SPI 110: From UART2 to XRAM 111: From XRAM to UART2 Note: It cannot configured when channel 1 is busy. LIN multiplexes UART2 channels, and the read/write direction is determined by LIN_CR[LINRW].
[2]	DBGSW	Sector Targeted in Debug Mode 0: XSFR as the Debug area (export address space: 0x4020 ~ 0x40FF) 1: XRAM as the Debug area (export address space: 0x0000 ~ 0x0317)

[1]	DBGEN	Debug Mode Enable DMA works in Debug mode when DMA1_CR0[DMACFG] is set to “101” and DMA1_CR0[DBGEN] to “1”. After SPI is enabled, DMA automatically sends relevant data in the sector defined by DMA1_CR0[DBGSW] via MOSI. DMA1_BA/DMA1_LEN defines the start address and range of the relevant data. 0: Disable 1: Enable Note: DMA Channel 1 Interrupt is automatically disabled in Debug mode
[0]	DMAIF	DMA Channel 1 Transfer Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: The interrupt event is generated Note: Setting DMA1_CR0[2:1] = 10 enables DMA Channel 1 Interrupt

27.2.3 DMA0_LEN (0x403C)

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

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	DMA0_LEN	Transfer Length of DMA Channel 0 Read: The number of the byte that is currently transferred by DMA Channel 0 (0 denotes the first byte) Write: XRAM data transfer length of DMA Channel 0 Note: It cannot be configured when channel 0 is busy. When DMA0_CR0[ENDIAN] = 1 (low bytes are received or transmitted first), it is recommended that DMA0_LEN be set to an odd number.

27.2.4 DMA0_BA (0x403E, 0x403F)

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

Bit	Name	Description
[15:12]	RSV	Reserved
[11:0]	DMA0_BA	Start address of data transfer by DMA Channel 0 Start address of XRAM data transfer by DMA Channel 0 It cannot be configured when channel 0 is busy Note: XRAM address space for data transfer by channel 0: DMA0_BA[11:0] ~ (DMA0_BA[11:0] + DMA0_LEN[5:0])

27.2.5 DMA1_LEN (0x403D)

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

Bit	Name	Description
[7:6]	RSV	Reserved
[5:0]	DMA1_LEN	Transfer Length of DMA Channel 1 Read: The number of the bytes that is currently transferred by DMA Channel 1 (0 denotes the first byte) Write: XRAM data transfer length of DMA Channel 1 Note: It cannot be configured when channel 1 is busy. When DMA0_CR0[ENDIAN] = 1 (low bytes are received or transmitted first), it is recommended that DMA1_LEN be set to an odd number.

27.2.6 DMA1_BA (0x4040, 0x4041)

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

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

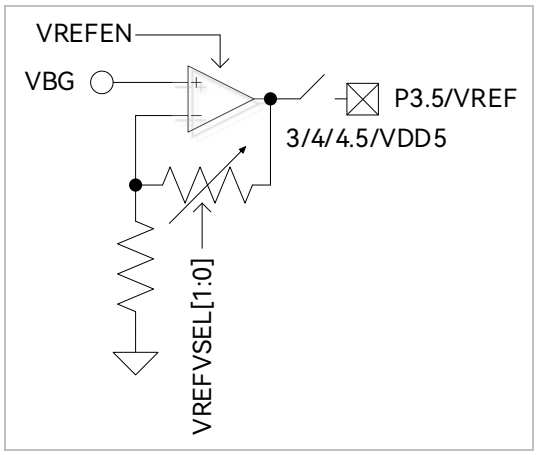
Note:

When I²C is selected as DMA channel peripherals (including from I²C to XRAM and from XRAM to I²C) , START + Address interrupt of I²C communication still requires to be cleared 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.

28 VREF

28.1 VREF Instructions

Figure 28-1 I/O Pins of VREF Module



The input and output ports of the VREF module are shown in Figure 28-1. VREF is the voltage reference generation block that provides internal voltage reference to ADC and DAC modules. VBG is the voltage supplied by the chip internally.

VREF is enabled when VREF_VHALF_CR[VREFEN] is set to “1”. The output voltage is selected by configuring VREF_VHALF_CR[VREFVSEL]. When P3_AN[5] = 1, VREF is output to P3.5.

28.2 VREF Register

28.2.1 VREF_VHALF_CR (0x404F)

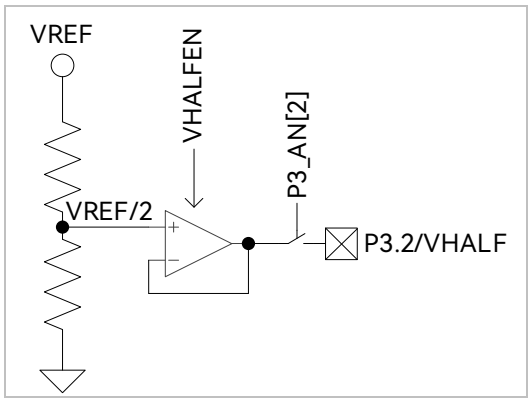
Bit	7	6	5	4	3	2	1	0
Name	VREFVSEL		RSV	VREFEN	RSV	VHALFSEL		VHALFEN
Type	R/W	R/W	-	R/W	-	R/W	R/W	R/W
Reset	0	0	-	0	-	1	1	0

Bit	Name	Description
[7:6]	VREFVSEL	VREF Module Output Voltage Selection 00: 4.5V 01: VDD5 10: 3V 11: 4V
[5]	RSV	Reserved
[4]	VREFEN	VREF Module Enable 0: Disable. P3_AN[5] is set to “1”, and external VREF is input from P3.5. 1: Enable. P3_AN[5] is set to “1”, and internal VREF is output to P3.5. A 0.1μF ~ 1μF external capacitor is added to improve the stability of VREF.
[3]	RSV	Reserved
[2:1]	VHALFSEL	VHALF Operating Voltage Selection (VREF Coefficient) 00:1/8 01:1/4 10:25/64 11:1/2 (default)
[0]	VHALFEN	VHALF Enable 0: Disable 1: Enable

29 VHALF

29.1 VHALF Instructions

Figure 29-1 I/O Pins of VHALF Module



The input and output ports of VHALF module are shown in Figure 29-1. This module generates the voltage reference. Its voltage value is controlled by register VREF_VHALF_CR[VHALFSEL], with the configurations 00: VREF/8; 01: VREF/4; 10: VREF*25/64; 11: VREF/2 (default)

VHALF is enabled when VREF_VHALF_CR[VHALFEN] is set to “1”, and the voltage is output to P3.2. A 1μF capacitor is added.

29.2 VHALF Register

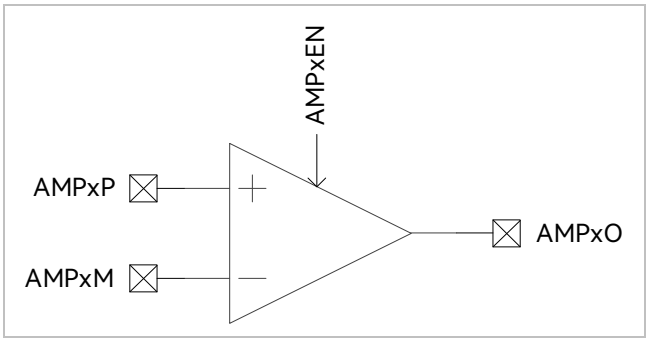
See VREF_VHALF_CR (0x404F) for details.

30 Operational Amplifiers

30.1 Operational Amplifier Introduction

The chip integrates four high-speed independent operational amplifiers, AMP0, AMP1, AMP2 and AMP3. Each operational amplifier has a separate enable bit, and can be configured as PGA.

Figure 30-1 Schematic Diagram of Operational Amplifier Module



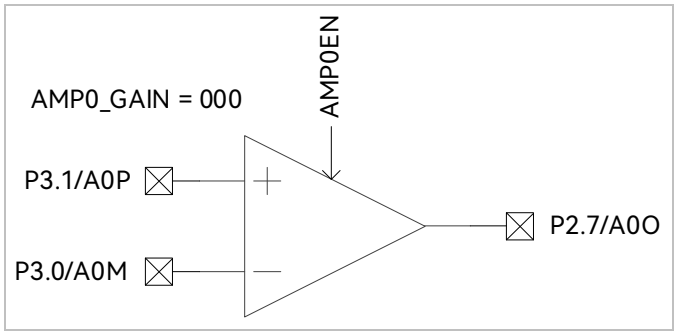
30.2 Operational Amplifier Instructions

30.2.1 Bus Current Sampling Operational Amplifier (AMP0)

AMP0 operates in three modes: normal mode, PGA differential input mode and PGA single-ended input mode.

30.2.1.1 AMP0 Normal Mode

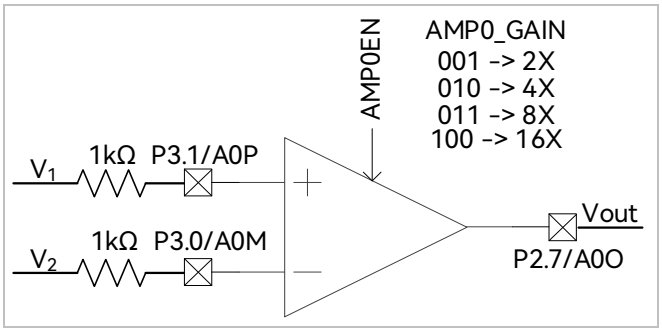
Figure 30-2 Bus Current AMP0



The I/O pins of AMP0 are shown in Figure 30-2. AMP0 is enabled when AMP_CR0[AMP0EN] = 1, and P2.7, P3.0 and P3.1 are automatically configured to analog signal mode by the hardware.

30.2.1.2 AMP0 PGA Differential Input Mode

Figure 30-3 AMP0 Operating in PGA Differential Input Mode

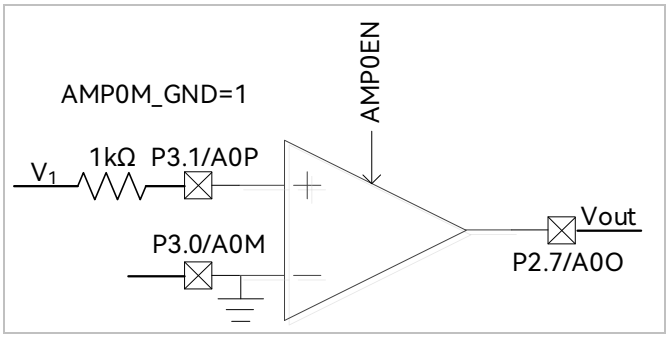


As shown in Figure 30-3, positive and negative inputs of AMP0 are connected with a 1kΩ resistor in the external circuit respectively.

When PGA differential Input Mode is selected for AMP0, the amplification gain is set by AMP_CR1[AMP0_GAIN], and AMP0 is enabled when AMP_CR0[AMP0EN] = 1. The relation between output and input of operational amplifier: Vout = VHALF + (V1 - V2)*AMP0_GAIN.

30.2.1.3 AMP0 PGA Single-ended Input Mode

Figure 30-4 AMP0 Operating in PGA Single-ended Input Mode



As shown in Figure 30-4, positive input of AMP0 is connected with a 1kΩ resistor in the external circuit, and negative input is connected to ground inside the MCU.

When PGA Single-ended Input Mode is selected for AMP0, the amplification gain is set by AMP_CR1 [AMP0_GAIN], and AMP0 is enabled when AMP_CR0[AMP0EN] = 1. The relation between output and input of operational amplifier:

When AMP_CR1[AMP0_GAIN] is set as 2x, Vout = 7/6*VHALF + 7/3*V1

When AMP_CR1[AMP0_GAIN] is set as 4x, Vout = 6/5*VHALF + 24/5*V1

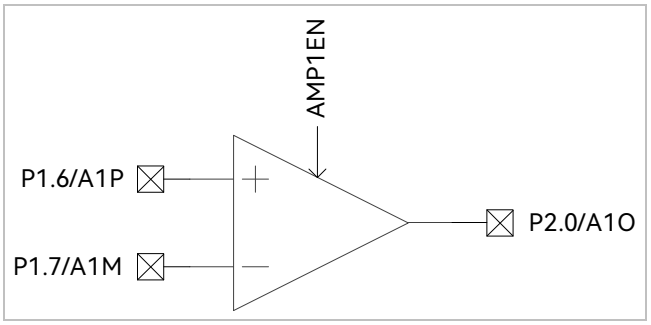
When AMP_CR1[AMP0_GAIN] is set as 8x, Vout = 11/9*VHALF + 88/9*V1

When AMP_CR1[AMP0_GAIN] is set as 16x, Vout = 21/17*VHALF + 336/17*V1

30.2.2 Phase Current Operational Amplifier (AMP1/AMP2)

30.2.2.1 AMP1 Normal Mode

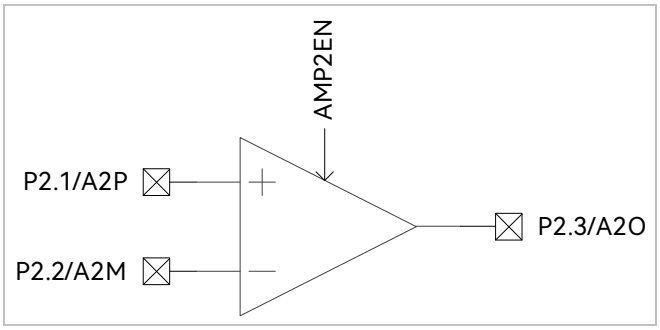
Figure 30-5 I/O Pins of AMP1



The I/O pins of AMP1 are shown in Figure 30-5. AMP1 is enabled when AMP_CR0[AMP1EN] = 1. P1.6, P1.7 and P2.0 pins are automatically configured to analog signal mode by the hardware. P1_AN[7:6] is set to “11” and P2_AN[0] to “1”.

30.2.2.2 AMP2 Normal Mode

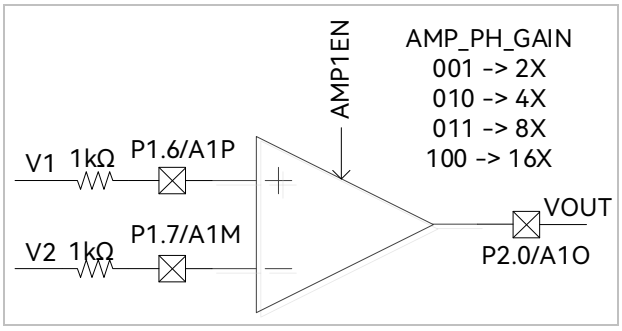
Figure 30-6 I/O Pins of AMP2



The I/O pins of AMP2 are shown in Figure 30-6. AMP2 is enabled when AMP_CR0[AMP2EN] = 1. P2.1, P2.2, and P2.3 are automatically configured to analog signal mode by the hardware. P2_ AN[3:1] is set to “111”.

30.2.2.3 AMP1 PGA Differential Input Mode

Figure 30-7 AMP1 Operating in PGA Differential Input Mode

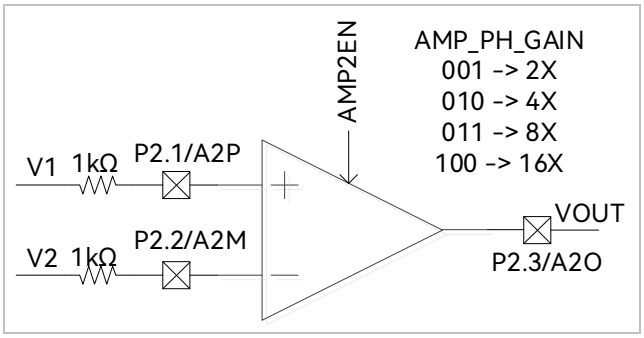


As shown in Figure 30-7, positive and negative inputs of AMP1 are connected with a 1kΩ resistor in the external circuit respectively.

When PGA Differential Input Mode is selected for AMP1, the amplification gain is set by AMP_CR1 [AMP_PH_GAIN], and AMP1 is enabled when AMP_CR0[AMP1EN] = 1. The relation between output and input of operational amplifier: Vout = VHALF + (V1 - V2)*AMP_PH_GAIN.

30.2.2.4 AMP2 PGA Differential Input Mode

Figure 30-8 AMP2 Operating in PGA Differential Input Mode

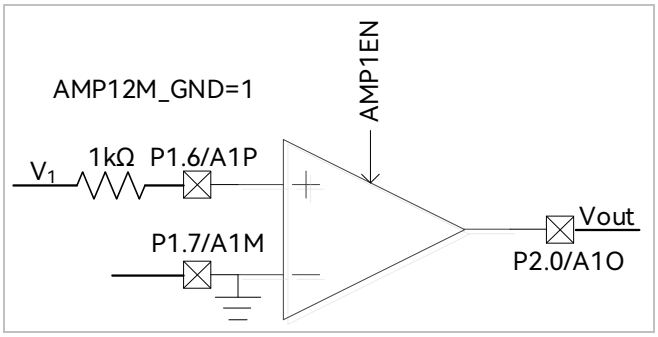


As shown in Figure 30-8, positive and negative inputs of AMP2 are connected with a 1kΩ resistor in the external circuit respectively.

When PGA Differential Input Mode is selected for AMP2, the amplification gain is set by AMP_CR1 [AMP_PH_GAIN]. AMP2 is enabled when AMP_CR0[AMP2EN] = 1. The relation between output and input of operational amplifier: Vout = VHALF + (V1 - V2)*AMP_PH_GAIN.

30.2.2.5 AMP1 PGA Single-ended Input Mode

Figure 30-9 AMP1 Operating in PGA Single-ended Input Mode



As shown in Figure 30-9, positive input of AMP1 is connected with a 1kΩ resistor in the external circuit, and negative input is connected to ground inside the MCU. When PGA Single-ended Input Mode is selected for AMP1, the amplification gain is set by AMP_CR1[AMP_PH_GAIN]. AMP1 is enabled by configuring AMP_CR0 [AMP12M_GND] = 1 and AMP_CR0[AMP1EN] = 1. The relation between output and input of operational amplifier:

When AMP_CR1[AMP_PH_GAIN] is set as 2x, Vout = 7/6*VHALF + 7/3*V1

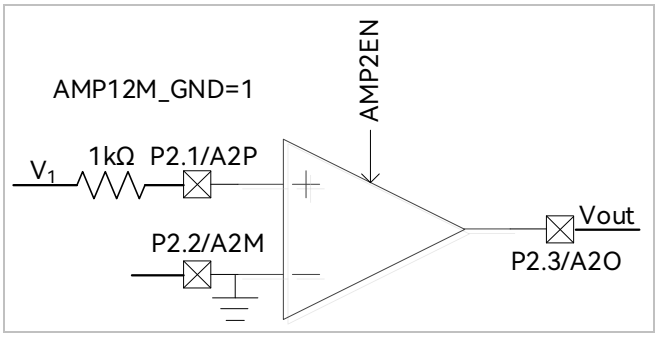
When AMP_CR1[AMP_PH_GAIN] is set as 4x, Vout = 6/5*VHALF + 24/5*V1

When AMP_CR1[AMP_PH_GAIN] is set as 8x, Vout = 11/9*VHALF + 88/9*V1

When AMP_CR1[AMP_PH_GAIN] is set as 16x, Vout = 21/17*VHALF + 336/17*V1

30.2.2.6 AMP2 PGA Single-ended Input Mode

Figure 30-10 AMP2 Operating in PGA Single-ended Input Mode



As shown in Figure 30-10, positive input of AMP2 is connected with a 1kΩ resistor in the external circuit, and negative input is connected to ground inside the MCU. When PGA Single-ended Input Mode is selected for AMP2, the amplification gain is set by AMP_CR1[AMP_PH_GAIN], and AMP2 is enabled by configuring

AMP_CR0[AMP12M_GND] = 1 and AMP_CR[AMP2EN] = 1. The relation between output and input of operational amplifier:

When AMP_CR1[AMP_PH_GAIN] is set as 2x, Vout = 7/6*VHALF + 7/3*V1

When AMP_CR1[AMP_PH_GAIN] is set as 4x, Vout = 6/5*VHALF + 24/5*V1

When AMP_CR1[AMP_PH_GAIN] is set as 8x, Vout = 11/9*VHALF + 88/9*V1

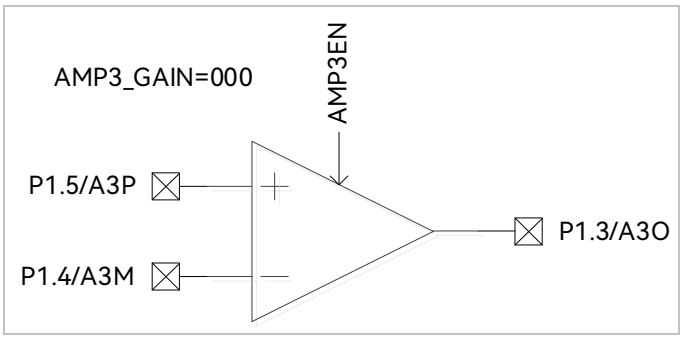
When AMP_CR1[AMP_PH_GAIN] is set as 16x, Vout = 21/17*VHALF + 336/17*V1

30.2.3 Operational Amplifier AMP3

AMP3 works in two modes: normal mode and PGA differential input mode.

30.2.3.1 AMP3 Normal Mode

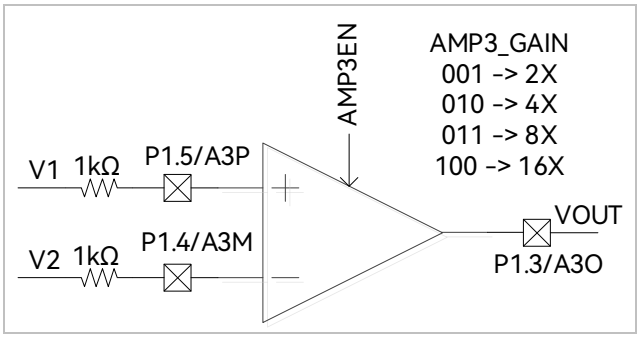
Figure 30-11 I/O Pins of AMP3



The I/O pins of AMP3 are shown in Figure 30-11. AMP3 is enabled when AMP_CR0[AMP3EN] = 1, and P1.5, P1.4 and P1.3 are automatically configured to analog input mode by the hardware. P1_AN[5:4] is set to “11”, P1_AN[HBMOD] to “1”, and P1_OE[3] to “04”.

30.2.3.2 AMP3 PGA Differential Input Mode

Figure 30-12 AMP3 Operating in PGA Differential Input Mode



As shown in Figure 30-12, positive and negative inputs of AMP3 are connected with a 1kΩ resistor in the external circuit respectively.

When PGA Differential Input Mode is selected for AMP3, the amplification gain is set by AMP_CR2 [AMP3_GAIN], and AMP3 is enabled when AMP_CR0[AMP3EN] = 1. The relation between output and input of operational amplifier: Vout = VHALF + (V1 - V2)*AMP3_GAIN.

30.3 Operational Amplifier Registers

30.3.1 AMP_CR0 (0x404E)

Bit	7	6	5	4	3	2	1	0
Name	CP_SEL	CP_EN	AMP12M_GND	AMP0M_GND	AMP3EN	AMP2EN	AMP1EN	AMP0EN
Type	R/W	R/W	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]	CP_SEL	Constant Power Channel Selection It is valid only when AMP_CR0[CP_EN] = 1. 0: AMP0 output on P2.7 pin is routed to P3.4 via the 16kΩ resistor 1: AMP0 output on P2.7 pin is routed to P3.3 via the16kΩ resistor Note: When AMP_CR0[CP_EN] = 1, AMP0 output (P2.7) is routed to P3.4/P3.3 via the 16kΩ internal resistor. P3.4/P3.3 pin must be provided with a 1μF external capacitor for bus average current sampling.
[6]	CP_EN	Constant Power Configuration Enable 0: Disable 1: Enable
[5]	AMP12M_GND	Enable the negative input of AMP1&2, after which the built-in P1.7 & P2.2 are wired to GND 0: Disable 1: Enable
[4]	AMP0M_GND	Enable the negative input of AMP0, after which the built-in P3.0 is wired to GND 0: Disable 1: Enable
[3]	AMP3EN	AMP3 Enable 0: Disable 1: Enable

[2]	AMP2EN	AMP2 Enable 0: Disable 1: Enable
[1]	AMP1EN	AMP1 Enable 0: Disable 1: Enable
[0]	AMP0EN	AMP0 Enable 0: Disable 1: Enable

30.3.2 AMP_CR1 (0x4034)

Bit	7	6	5	4	3	2	1	0
Name	AMP_PH_GAIN			RSV		AMP0_GAIN		
Type	R/W	R/W	R/W	-	-	R/W	R/W	R/W
Reset	0	0	0	-	-	0	0	0

Bit	Name	Description
[7:5]	AMP_PH_GAIN	AMP1&AMP2 Amplification Gain Setting. See descriptions on AMP_CR1[AMP0_GAIN].
[4:3]	RSV	Reserved
[2:0]	AMP0_GAIN	AMP0 Amplification Gain Setting 000: The gain is configured by external circuit 001: 2x 010: 4x 011: 8x 100: 16x 101: Reserved 110: Reserved 111: Reserved Note: The built-in amplification is isotropic amplification. When the difference of input voltage is 0, the output voltage is VHALF. For other applications, AMP0_GAIN is set to “000” to select external circuit to configure the gain.

30.3.3 AMP_CR2 (0x4046)

Bit	7	6	5	4	3	2	1	0
Name	RSV					AMP3_GAIN		
Type	-	-	-	-	-	R/W	R/W	R/W
Reset	-	-	-	-	-	0	0	0

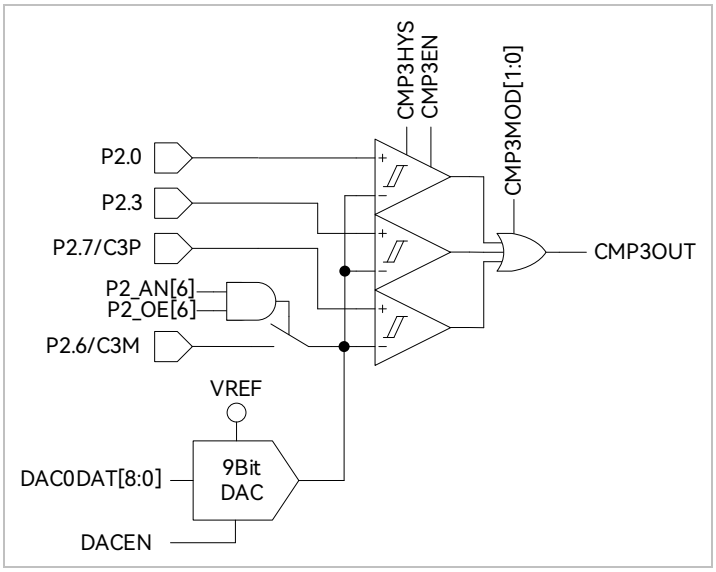
Bit	Name	Description
[7:3]	RSV	Reserved
[2:0]	AMP3_GAIN	AMP3 amplification gain setting. See descriptions on AMP_CR1[AMP0_GAIN] in section AMP_CR1 (0x4034)

31 Comparators

31.1 Comparator Operations

31.1.1 CMP3

Figure 31-1 I/O Pins of CMP3



The I/O pins of CMP3 are shown in Figure 31-1.

CMP3 configurations:

1. Configure P2_AN[6] and P2_OE[6] to “1” to enable VREF on the negative input of CMP3. 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 and GND (the recommended capacitance value is 100pF, and the output voltage stabilizes after DAC0 output for a period of time);

2. Configure CMP_CR1[CMP3MOD] to select single comparator input, dual comparator input, or triple comparator input mode;

   - When CMP_CR1[CMP3MOD] = 00, CMP3 works in Single-comparator Input Mode. The connection of input and output pins are shown in Figure 31-2.
   - When CMP_CR1[CMP3MOD] = 01, CMP3 works in Dual-comparator Input Mode. The connection

of input and output pins are shown in Figure 31-3.

> When CMP_CR1[CMP3MOD] = 1X, CMP3 works in Triple-comparator Input Mode. The connection of input and output pins are as shown in Figure 31-4.

3. Configure CMP_CR1[CMP3HYS] to enable or disable hysteresis;

4. Set CMP_CR1[CMP3EN] = 1 to enable CMP3.

Figure 31-2 Single-comparator Input Mode

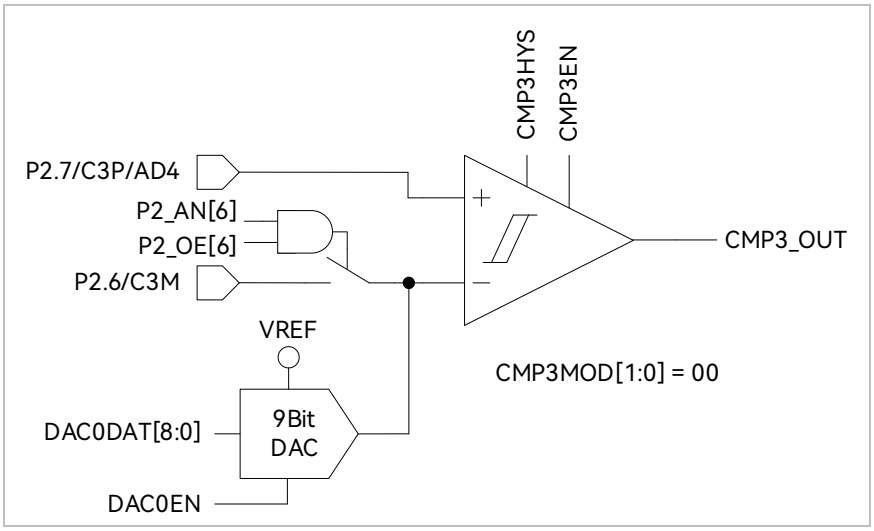


Figure 31-3 Dual-comparator Input Mode

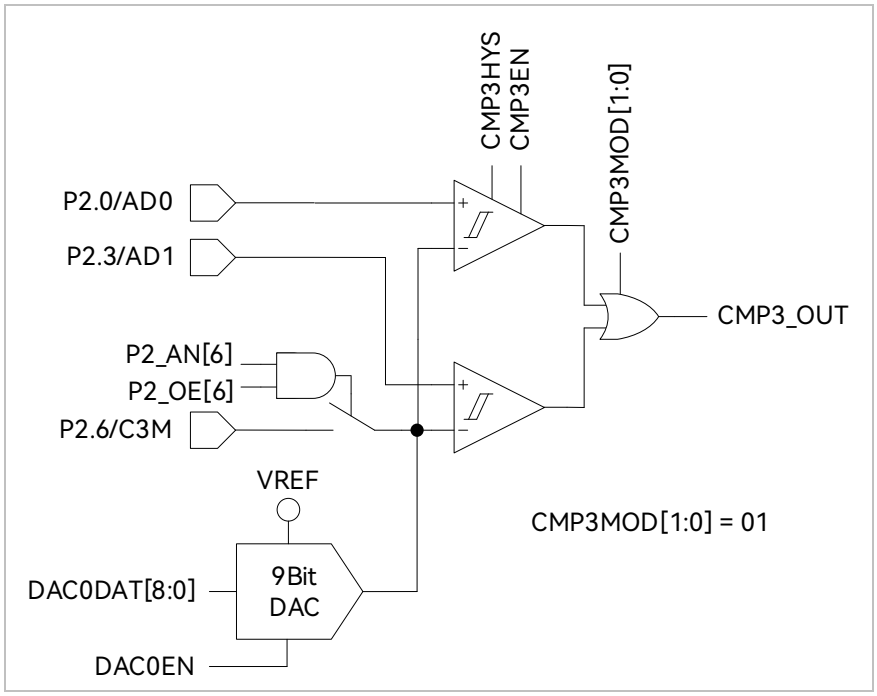
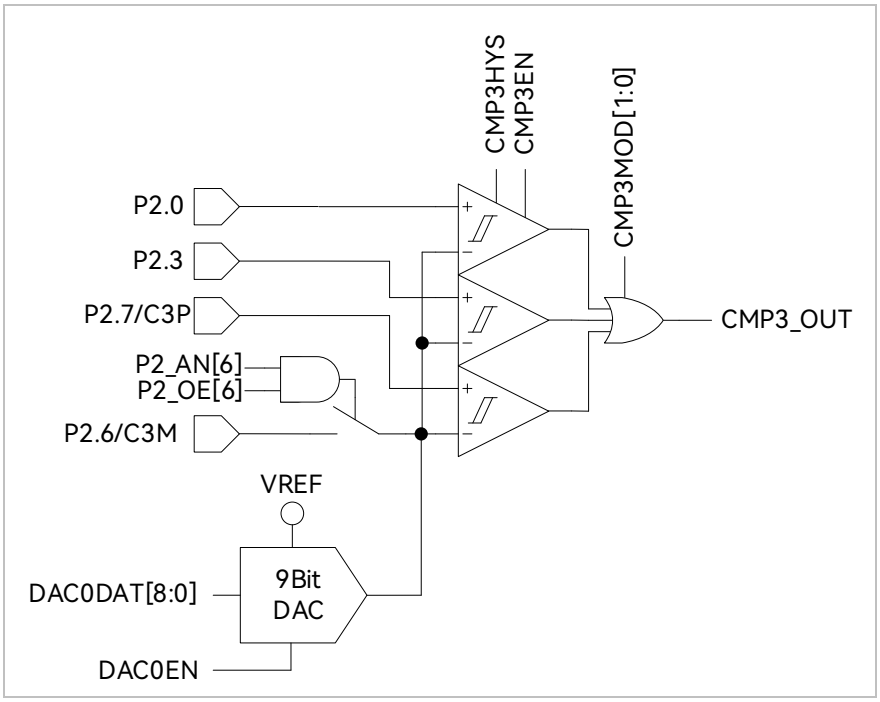


Figure 31-4 Triple-comparator Input Mode



31.1.1.1 Overcurrent Protection (OCP)

When an overcurrent protection signal is generated, DRV_OUT[MOE] is automatically cleared to output idle voltage and stop motor drive for chip and motor protection. OCP feature is enabled when EVT_FILT[MOEMD] = 01, which automatically turns off the output and generates an OCP interrupt request if the current exceeds the threshold. When EVT_ FILT [MOEMD] = 00, the output is not automatically turned off if the current exceeds the threshold. However, an OCP request is generated.

The source of OCP interrupt is selected by configuring EVT_FILT[MOEMD] ≠ 00 or EVT_FILT [INT0_MOE_EN] = 1, namely CMP3 interrupt or external interrupt INT0. When EVT_ FILT[INT0_MOE_EN] = 1, TCON[IT0] 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[INT0_MOE_EN] = 0 and CMP_CR0[CMP3IM] = 11, the OCP output is generated on the raising edge of CMP3. At this time, the source of OCP interrupt is CMP3. In triple-shunt current sampling mode, CMP_CR1[CMP3MOD] is configured to select triple-comparator input mode. When current of any phase is over the threshold, CMP3 generates an OCP signal. For other sampling modes, CMP_CR1[CMP3MOD] is configured to choose single-comparator input mode. When bus current is over the threshold, CMP3 generates an OCP signal.

Configuring EVT_FILT[EFDIV] enables the filtering of interrupt signals for OCP, and programming EVT_FILT[EFDIV] = 01/10/11 selects filter width of 6/12/24 clock cycles. When the filtering feature is enabled, the filtered signal is delayed by 6/12/24 clock cycles compared to the signal before filtering.

31.1.1.2 Cycle-by-cycle Current Limiting

The cycle-by-cycle current limiting feature is applied to square-wave-based drive control of BLDC motors. When an OCP event occurs, DRV_OUT[MOE] is set to "1" by hardware after it has been cleared to "0" for a period of time, so that the motor drive is automatically restored. When CMP_CR0[CMP3IM] = 11, DRV_OUT[MOE] is cleared to "0" on the rising edge of CMP3OUT to protect the motor. When EVT_FILT[MOEMD] = 10, the outputs are automatically turned off upon an OCP interrupt and DRV_OUT[MOE] is enabled automatically upon Driver timer overflow/underflow events or after 10μs to restore motor drive. When EVT_FILT[MOEMD] = 11, the outputs are automatically turned off upon an OCP interrupt and DRV_OUT[MOE] is enabled automatically upon Driver timer overflow/underflow events or after 5μs to restore motor drive.

Figure 31-5 Cycle-by-cycle Current Limiting Waveform
(t2 - t1 = 10μs) when EVT_FILT[MOEMD] = 10

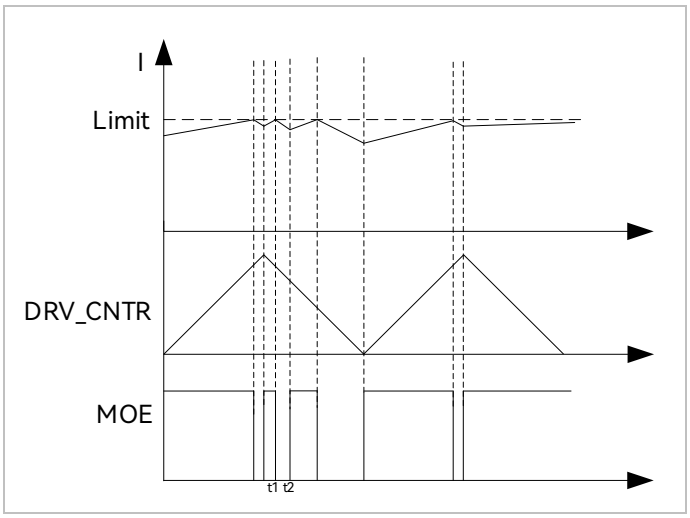
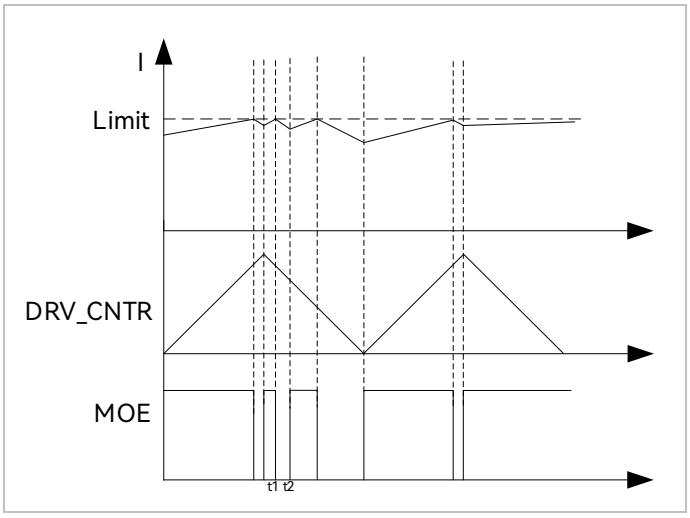


Figure 31-6 Cycle-by-cycle Current Limiting Waveform

(t2 - t1 = 5μs) when EVT_FILT[MOEMD] = 11



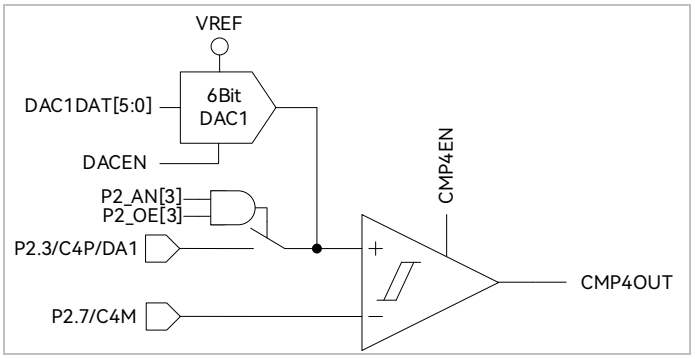
31.1.2 CMP4

CMP4OUT can be read by software or reversed on external interrupt INT0. When CMP3 is used for cycle-by-cycle current limiting protection, CMP4 is used for bus current protection. When bus current OCP feature of CMP4 is triggered, output must be turned off by software.

CMP4 configurations:

1. Configure P2_AN[3] and P2_OE[3] to “1” to enable VREF on the positive input of CMP4. The VREF source can be on-chip DAC1 output voltage or external circuit input voltage. Select DAC1 output, and place an external capacitor between P2.3 and GND (the recommended capacitance value is 100pF, and the output voltage stabilizes after DAC1 output for a period of time);
2. Configure P2_AN[7] = 1 to assign P2.7 to analog signal;
3. Configure CMP_CR2[CMP4EN] = 1 to enable CMP4;
4. Clear INT0 flag bit to enable INT0;
5. Set LVSR[EXT0CFG] = 111 to select CMP4 as the source of INT0;
6. Configure TCON[IT0] = 01 to select falling edge triggered INT0.

Figure 31-7 Schematic Diagram of CMP4



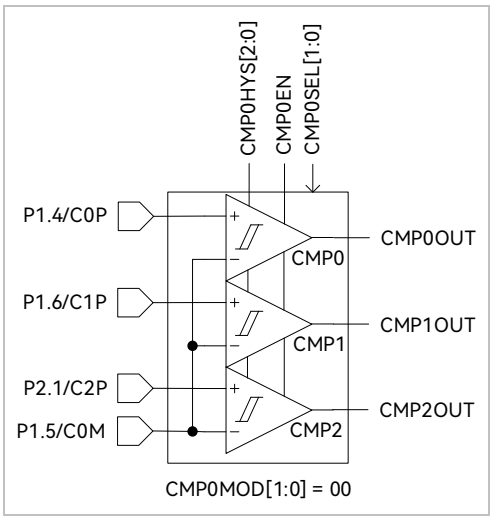
31.1.3 Comparator Group (CMPG)

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

When CMP_CR2[CMP0MOD] = 00, CMPG works with three comparators without built-in resistor. The I/O pins are shown in Figure 31-8. It is used for BEMF detection with the external virtual neutral point resistors. The negative inputs of the three comparators are connected together to P1.5, and the positive inputs are connected to P1.4, P1.6 and P2.1 respectively. The outputs are CMP0OUT, CMP1OUT and CMP2OUT respectively. The number of comparators working in this mode is defined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0, CMP1 and CMP2 work simultaneously, which is the recommended configuration. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works. When CMP_CR2[CMP0SEL] = 11, only CMP2 works.

Figure 31-8 CMPG with Three Built-in Comparators

(without Built-in Resistor)

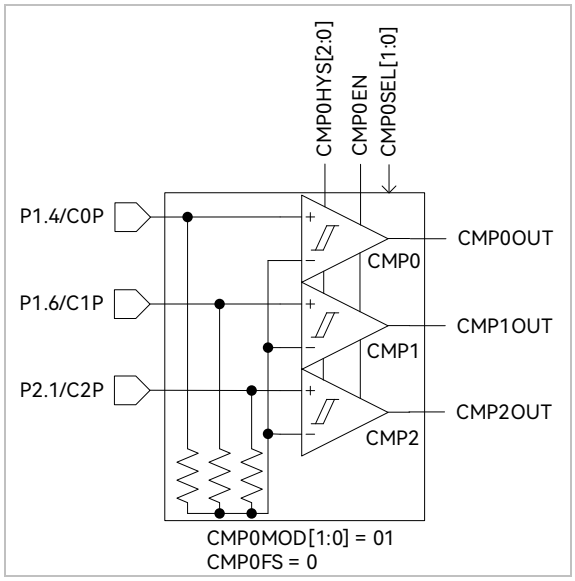


When CMP_CR2[CMP0MOD] = 01, CMPG works with three comparators and built-in resistors. It is used for BEMF detection with the internal virtual neutral point resistors. The input port is selected by setting the function switching bit CMP_CR4[CMP0FS]. The number of comparators operating in this mode is defined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0, CMP1 and CMP2 work simultaneously, which is the recommended configuration. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works. When CMP_CR2[CMP0SEL] = 11, only CMP2 works.

When CMP_CR4[CMP0FS] = 0, the I/O pins are shown in Figure 31-9. The negative inputs of the three comparators are connected together to the center point of the built-in resistor. The positive inputs are connected to P1.4, P1.6 and P2.1 respectively, and the outputs are CMP0OUT, CMP1OUT and CMP2OUT respectively.

Figure 31-9 CMPG with Built-in Three Comparators and Resistors

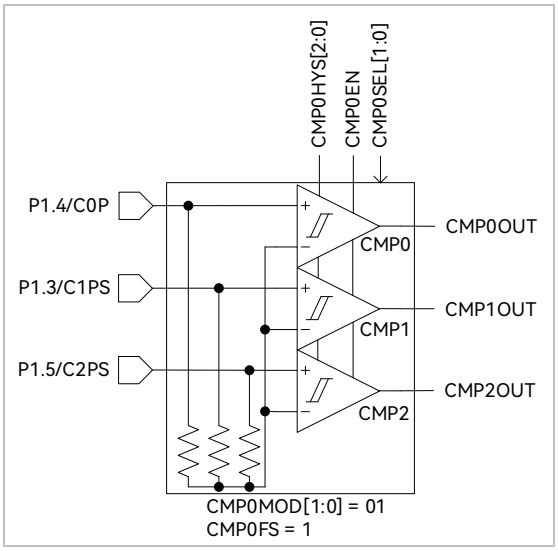
(without Function Switching)



When CMP_CR4[CMP0FS] = 1, the I/O pins are shown in Figure 31-10. The negative inputs of the three comparators are connected together to the center point of the built-in resistor. The positive inputs are connected to P1.4, P1.3 and P1.5 respectively, and the outputs are CMP0OUT, CMP1OUT and CMP2OUT respectively.

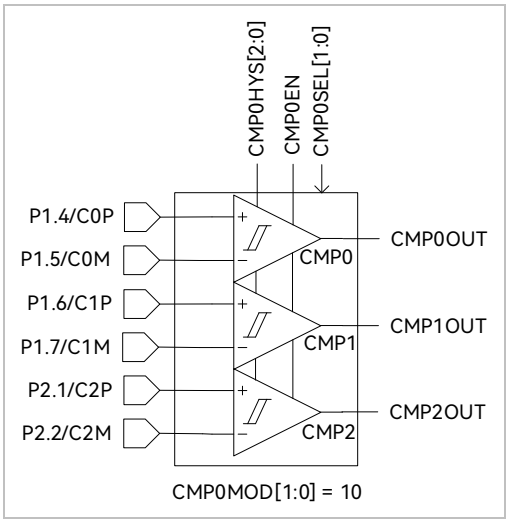
Figure 31-10 CMPG with Built-in Three Comparators and Resistors

(with Function Switching)



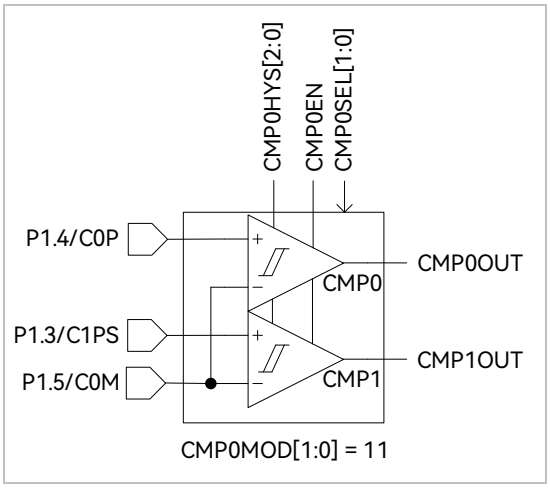
When CMP_CR2[CMP0MOD] = 10, CMPG works in triple-differential-comparator mode, where the motor rotor position is detected by differential Hall signals. The input and output pins are shown in Figure 31-11. The negative inputs of the three comparators are respectively connected to P1.5, P1.7 and P2.2, and the positive inputs are respectively connected to P1.4, P1.6 and P2.1. The outputs are CMP0OUT, CMP1OUT and CMP2OUT respectively. The number of comparators working in this mode is defined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0, CMP1 and CMP2 work simultaneously, which is the recommended configuration. When CMP_CR2[CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works. When CMP_CR2[CMP0SEL] = 11, only CMP2 works.

Figure 31-11 CMPG in Triple-differential-comparator Mode



When CMP_CR2[CMP0MOD] = 11, CMPG works in dual-comparator mode. The I/O pins are shown in Figure 31-12. The negative inputs of the two comparators are connected together to P1.5, and the positive inputs are connected to P1.4 and P1.3 respectively. The outputs are CMP0OUT and CMP1OUT respectively. The number of comparators in this mode is defined by CMP_CR2[CMP0SEL]. When CMP_CR2[CMP0SEL] = 00, CMP0 and CMP1 work simultaneously, which is the recommended configuration. When CMP_CR2 [CMP0SEL] = 01, only CMP0 works. When CMP_CR2[CMP0SEL] = 10, only CMP1 works.

Figure 31-12 CMPG in Dual-comparator Mode

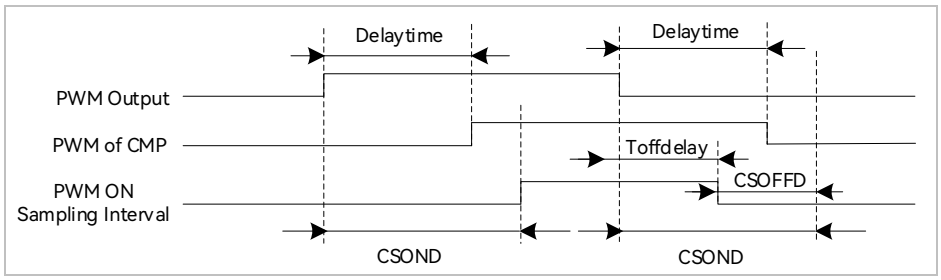


The output signals of CMP0/CMP1/CMP2 are sent to Timer1 after filtering and sampling modules.

31.1.4 Comparator Sampling

The comparator sampling feature is mainly used for the square-wave control and RSD, which eliminates the on-off interference from driving circuit. See section 15.1.2.3 for square-wave control and section 16.1.7.1 for RSD.

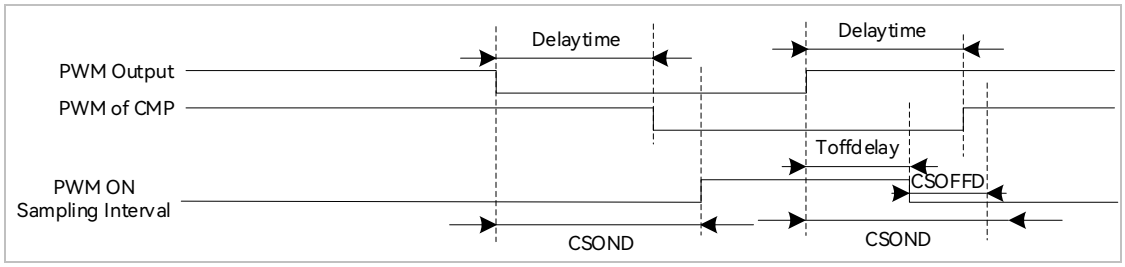
Figure 31-13 PWM ON Sampling Mode



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, switching speed of the power device, and input delay

and hysteresis settings of the comparator. As shown in Figure 31-13, the delay time is from the chip output to the comparator output. When high-level sampling is performed, the sampling interval shall be enveloped by actual high-level output of the comparator. First, the sampling ON-delayed time CMP_SAMR[CSOND] is set to overcome the output delay and the oscillation interval of the power device. At the end of the sampling interval, CMP_SAMR[CSOND] is delayed after the falling edge of PWM, at which time the actual sampling window has exceeded the corresponding high-level interval. The sampling OFF-lead time CMP_SAMR[CSOFFD] is set to stop sampling Toffdelay after the PWM output falling edge, where Toffdelay = CMP_SAMR[CSOND] − CMP_SAMR[CSOFFD]. By configuring CMP_SAMR[CSOND] and CMP_SAMR [CSOFDD], the sampling interval can be located in the high-level interval of the actual output of the comparator.

Figure 31-14 PWM OFF Sampling Mode



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

Method for measuring the delay of PWM output to comparator: Set CMP_CR3[SAMSEL] = 00 to disable the comparator sampling delay feature. Set CMP_CR3[CMPSEL] to select the corresponding comparator output to test pin P0.1. Enable the PWM output and comparator, manually rotate the motor to change the comparator value, and measure the delay between the PWM output and the comparator output.

31.1.5 Comparator Output

CMP_CR3[CMPSEL] is configured to output results of one comparator to P0.1 or select function switching to P3.4.

31.2 Comparator Registers

31.2.1 CMP_CR0 (0xD5)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IM		CMP2IM		CMP1IM		CMP0IM	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7:6]	CMP3IM	CMP3 Interrupt Mode 00: No interrupt is generated. 01: An interrupt is generated upon rising edge 10: An interrupt is generated upon falling edge 11: When a rising edge is detected, DRV_OUT[MOE] is cleared to “0”, and the interrupt event flag bit CMP_SR[CMP3IF] is set to “1”. However, the interrupt is disabled. Note: In Cycle-by-cycle Current Limiting mode, EVT_FILT[MOEMD] must be set to 10/11.
[5:4]	CMP2IM	CMP2 Interrupt Mode See descriptions on CMP_CR0[CMP0IM]
[3:2]	CMP1IM	CMP1 Interrupt Mode See descriptions on CMP_CR0[CMP0IM]
[1:0]	CMP0IM	CMP0 Interrupt Mode 00: No interrupt 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

31.2.2 CMP_CR1 (0xD6)

Bit	7	6	5	4	3	2	1	0
Name	RSV	CMP3MOD		CMP3EN	CMP3HYS	RSV	CMP0HYS	
Type	-	R/W	R/W	R/W	R/W	-	R/W	R/W
Reset	-	0	0	0	0	-	0	0

Bit	Name	Description
[7]	RSV	Reserved

[6:5]	CMP3MOD	CMP3 Mode Selection Negative input connected to P2.6 or DAC0 output 00: Single-comparator mode, where P2.7 is connected to the positive input, as shown in Figure 31-2. 01: Dual-comparator mode, where P2.0 and P2.3 are connected to the positive inputs, as shown in Figure 31-3. 1X: Triple-comparator mode, where P2.0, P2.3 and P2.7 are connected to the positive inputs, as shown in Figure 31-4.
[4]	CMP3EN	CMP3 Enable 0: Disable 1: Enable
[3]	CMP3HYS	CMP3 Hysteresis Voltage Selection 0: No hysteresis 1: 15mv
[2]	RSV	Reserved
[1:0]	CMP0HYS	CMP0/1/2 Hysteresis Voltage Selection 00: No hysteresis 01: ± 3mV 10: + 6mV 11: ± 12mV

31.2.3 CMP_CR2 (0xDA)

Bit	7	6	5	4	3	2	1	0
Name	CMP4EN	CMP0MOD		CMP0SEL		RSV		CMP0EN
Type	R/W	R/W	R/W	R/W	R/W	-	-	R/W
Reset	0	0	0	0	0	-	-	0

Bit	Name	Description
[7]	CMP4EN	CMP4 Enable 0: Disable 1: Enable
[6:5]	CMP0MOD	CMPG Mode Setting 00: CMPG with built-in three comparators (without built-in resistor), as shown in Figure 31-8. 01: CMPG with built-in three comparators and resistors, where function switching is selected by configuring CMP_CR4[CMP0FS], as shown in Figure 31-9 and Figure 31-10. 10: CMPG in triple-differential-comparator mode, as shown in Figure 31-11 11: CMPG in dual-comparator mode , where only CMP0 and CMP1 work, as shown in Figure 31-12.
[4:3]	CMP0SEL	CMPG Pin Combination Selection, used with CMP_CR2[CMP0MOD] bit. It is set to 00 by default. In square-wave drive applications, TIM1_DBRx[T1CPE] automatically

controls CMP_CR2[CMP0SEL] to enable or disable each comparator.

Table 31-1 Function Description of CMPG Port and CMP_CR2[CMP0MOD] Combination

CMP0MOD	CMP0SEL	Description
00	00	CMP0/1/2 work simultaneously, as shown in Figure 31-8. The negative input of these comparators are connected to C0M. The hardware automatically compares the positive inputs C0P, C1P and C2P with C0M, and the output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.
	01	Only CMP0 works. The positive input is connected to C0P, and the negative input to C0M. The output results are transferred to CMP0OUT.
	10	Only CMP1 works. The positive input is connected to C1P, and the negative input to C0M. The output results are transferred to CMP1OUT.
	11	Only CMP2 works. The positive input is connected to C2P, and the negative input to C0M. The output results are transferred to CMP2OUT.
01	00	CMP0/1/2 work simultaneously, as shown in Figure 31-9 and Figure 31-10. The negative inputs of these 3 comparators are connected to the center of built-in resistor. When CMP_CR4[CMP0FS] = 0, the hardware automatically compares the positive inputs C0P, C1P and C2P with C0M. When CMP_CR4 [CMP0FS] =1, the hardware automatically compares the positive inputs C0P, C1PS and C2PS with C0M. The output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT respectively.
	01	Only CMP0 works. The positive input is connected to C0P, and the negative input to the center of built-in BEMF resistor. The output results are transferred to CMP0OUT.
	10	Only CMP1 works. When CMP_CR4[CMP0FS] = 0, the positive input is connected to C1P, and when CMP_CR4[CMP0FS] = 1, it is connected to C1PS. The negative input is connected to the center of built-in BEMF resistor. The output results are transferred to CMP1OUT.
	11	Only CMP2 works. When CMP_CR4[CMP0FS] = 0, the positive input is connected to C2P, and when CMP_CR4[CMP0FS] = 1, it is connected to C2PS. The negative input is connected to the center of built-in

				BEMF resistor. The output results are transferred to CMP2OUT.
		10	00	CMP0/1/2 work simultaneously, as shown in Figure 31-11. The positive inputs of these comparators are connected to C0P, C1P and C2P respectively, and the negative inputs are connected to C0M, C1M and C2M respectively. The output results are transferred to CMP0OUT, CMP1OUT and CMP2OUT.
			01	Only CMP0 works. The positive input is connected to C0P, and the negative input to C0M, and the output results are transferred to CMP0OUT.
			10	Only CMP1 works. The positive input is connected to C1P, and the negative input to C1M. The output results are transferred to CMP1OUT.
			11	Only CMP2 works. The positive input is connected to C2P, and the negative input to C2M. The output results are transferred to CMP2OUT.
		11	00	CMP0/1 work simultaneously, as shown in Figure 31-12. The positive inputs are connected to C0P and C1PS respectively, and the negative inputs to C0M. The outputs results are transferred to CMP0OUT and CMP1OUT respectively.
			01	Only CMP0 works. The positive input is connected to C0P, and the negative input to C0M. The output results are transferred to CMP0OUT.
			10	Only CMP1 works. The positive input is connected to C1PS, and the negative input to C0M. The output results are transferred to CMP1OUT.
			11	Reserved
[2:1]	RSV	Reserved		
[0]	CMP0EN	CMP0/1/2 Enable 0: Disable 1: Enable		

31.2.4 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 00: Debug Output Disable 01: Freewheeling shielding is completed and ZCP signal is detected 10: ADC Triggered Signal 11: Comparator Sampling Interval
[4:3]	SAMSEL	Sampling Delay 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 only, with time delay according to CMP_SAMR 10: Sampling at PWM ON only, with time delay according to CMP_SAMR 11: Sampling at both PWM ON and OFF, with time delay according to CMP_SAMR
[2:0]	CMPSEL	Comparator Output Selection Output signals of one selected comparator to P0.1, which can be transferred to P3.4. 000: No output 001: CMP0 010: CMP1 011: CMP2 100: CMP3 101: CMP4 110: Reserved 111: Omega Startup Flag (Estimator Output Angle Flag, see section 14.1.9.3)

31.2.5 CMP_CR4 (0xE1)

Bit	7	6	5	4	3	2	1	0
Name	CMP4OUT	RSV		CMP3P4M_FS	RSV	FAEN	CMP0FS	RSV
Type	R	-	-	R/W	-	R/W	R/W	-
Reset	1	-	-	0	-	0	0	-

Bit	Name	Description
[7]	CMP4OUT	CMP4 comparison results
[6:5]	RSV	Reserved
[4]	CMP3P4M_FS	CMP3P and CMP4M are switched to output at P3.4. It is used to directly transfer the sampled bus current (bus current operational amplifier AMP0O output to P3.4) to OCP comparator. 0: Disable. 1: Enable, with output to P3.4, where CMP3 has only one input channel at the positive input.
[3]	RSV	Reserved

[2]	FAEN	Filtered Signal Sampling Coefficient Scale-up Enable After this bit is enabled, the base clock rates of TIM1_CR3[T1INM] and CMP_SAMR are scaled up by 4 times. 0: Disable 1: Enable
[1]	CMP0FS	CMP1/CMP2 Function Switching 0: Disable, as shown in Figure 31-9 1: Enable, valid when CMP_CR2[CMP0_MOD] = 01, see Figure 31-10
[0]	RSV	Reserved

31.2.6 CMP_SAMR (0x40AD)

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

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

31.2.7 CMP_SR (0xD7)

Bit	7	6	5	4	3	2	1	0
Name	CMP3IF	CMP2IF	CMP1IF	CMP0IF	CMP3OUT	CMP2OUT	CMP1OUT	CMP0OUT
Type	R/W0	R/W0	R/W0	R/W0	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Description
[7]	CMP3IF	CMP3 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect
[6]	CMP2IF	CMP2 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0”. 1: No effect
[5]	CMP1IF	CMP1 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0”. 1: No effect
[4]	CMP0IF	CMP0 Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0”. 1: No effect
[3]	CMP3OUT	CMP3 comparison results
[2]	CMP2OUT	CMP2 comparison results
[1]	CMP1OUT	CMP1 comparison results
[0]	CMP0OUT	CMP0 comparison results

31.2.8 HALL_CR (0xE2)

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

Bit	Name	Description
[7]	HALL_IF	Hall Interrupt Flag 0: No Hall edge change is detected 1: Hall edge change is detected
[6]	HALL_IE	Hall Interrupt Enable 0: Disable 1: Enable
[5:4]	RSV	Reserved
[3]	HALLSEL	Hall Input Selection 0: P0.2/P3.6 1: P1.4/P1.6/P2.1
[2]	HALL2	Hall2 Level 0: Hall2 level = 0 1: Hall2 level = 1
[1]	HALL1	Hall1 Level 0: Hall1 level = 0 1: Hall1 level = 1
[0]	HALL0	Hall0 Level 0: Hall0 level = 0 1: Hall0 level = 1

31.2.9 EVT_FILT (0xD9)

Bit	7	6	5	4	3	2	1	0
Name	RSV			MOEMD		INT0_MOE_EN	EFDIV	
Type	-	-	-	R/W	R/W	R/W	R/W	R/W
Reset	-	-	-	0	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4:3]	MOEMD	MOE Cleared and Enabled by Hardware MOE is cleared and enabled by hardware upon current protection. 00: MOE is not automatically cleared. 01: MOE is automatically cleared. 10: MOE is automatically cleared and enabled by hardware upon Driver timer overflow/underflow events or after 10 µs (for square-

		wave drive). 11: MOE is automatically cleared and enabled automatically upon Driver timer overflow/underflow events or after 5 μs (for square-wave drive).
[2]	INT0_MOE_EN	MOE OFF Triggered by INT0 Interrupt 0: Disable 1: Enable
[1:0]	EFDIV	Filter Width for Current Protection 00: Disable 01: 6 system clock cycles 10: 12 system clock cycles 11: 24 system clock cycles

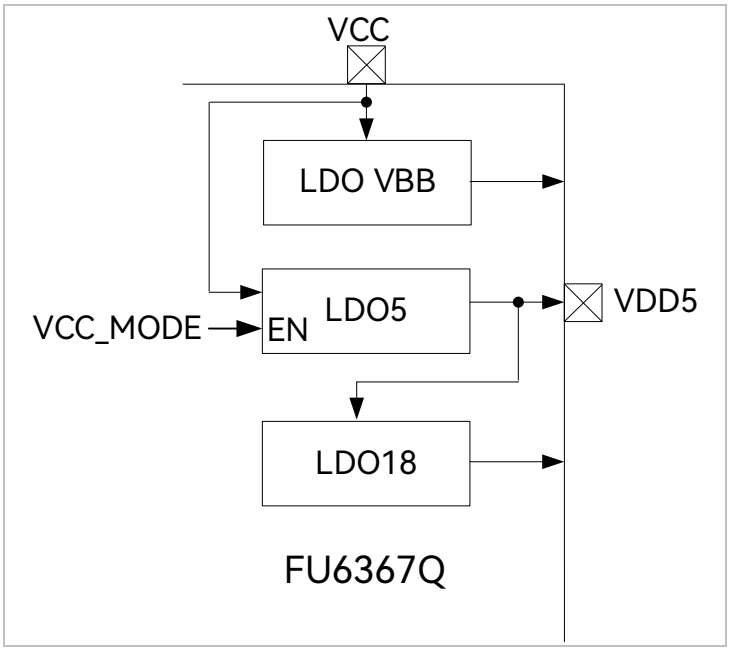
32 Power Supply

32.1 LDO

FU6367Q contains three built-in LDO output modules: LDO VBB, LDO5 and LDO18.

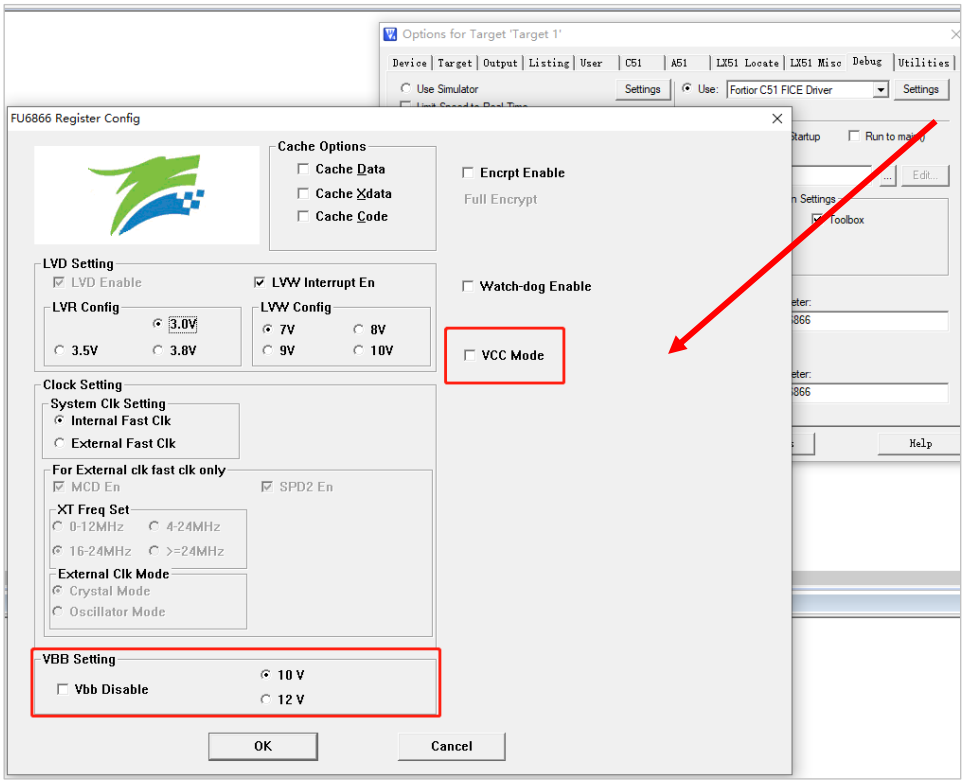
32.1.1 LDO Operations

Figure 32-1 Functional Block Diagram of Power Supply Module



The I/O pins of LDO module is shown in Figure 32-1. The LDO module converts the input supply voltage to VBB and 5V (VDD5) as the power supply for built-in analog module. As shown in Figure 32-2, if Vbb Disable is checked, LDO VBB does not work, and if it is unchecked, LDO VBB supplies 10V or 12V voltage. Internal LDO5 or external supply for VDD5 is selected by configuring VCC_MODE. As shown in Figure 32-2, VCC_MODE = 0 if VCC_MODE is unchecked, where internal LDO supplies VDD5 voltage; and VCC_MODE = 1 if VCC_MODE is checked, where external 5V power supply is connected to VDD5 pin.

Figure 32-2 VCC_MODE Configurations



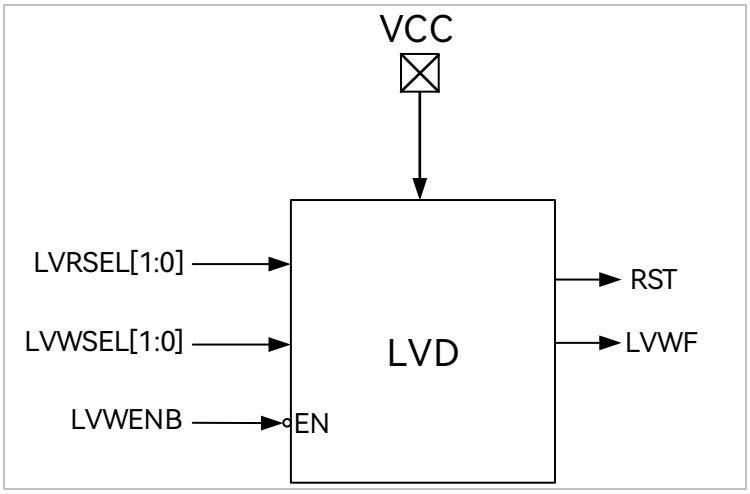
32.2 Low Voltage Detection (LVD)

32.2.1 LVD Introduction

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

32.2.2 LVD Operations

Figure 32-3 LVD Module



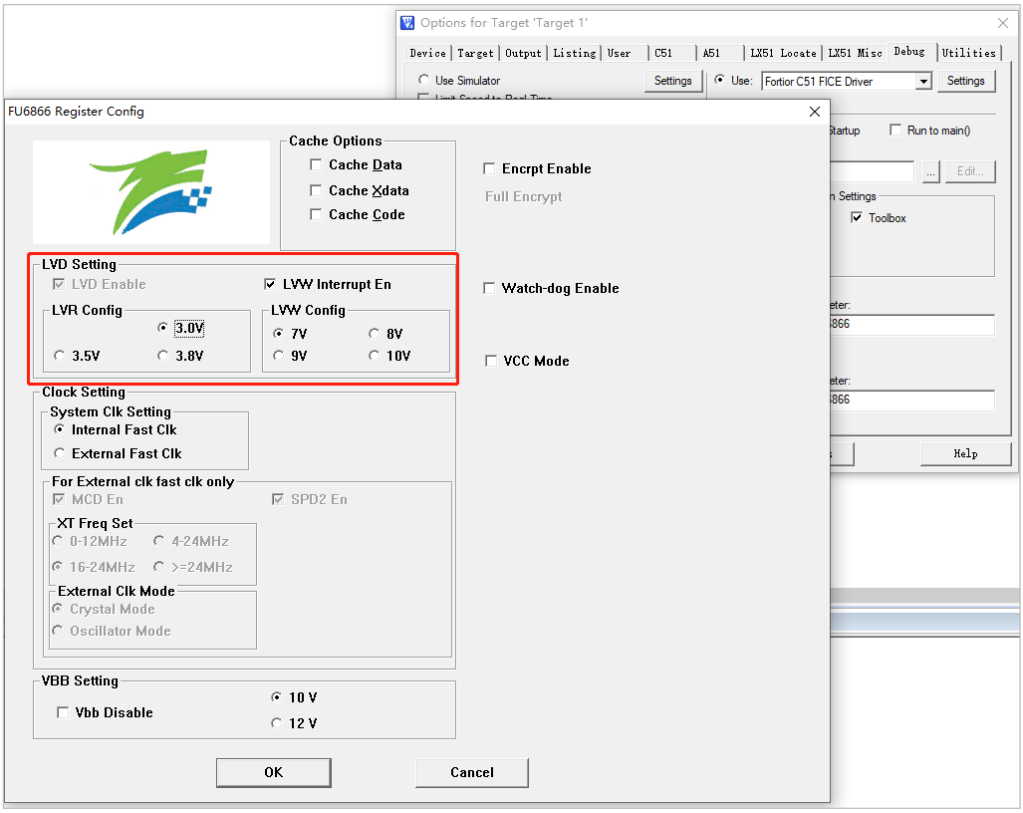
The operating instructions for LVD module are as follows:

- LV warning and LV reset are always enabled by default.
- 7/8/9/10V can be selected for LV warning threshold. When the interrupt feature is enabled, an interrupt is triggered if VCC voltage is lower than the LV warning threshold.
- 3.0/3.5/3.8V can be selected for the LV reset threshold. The chip resets when VCC voltage is lower than the LV reset voltage threshold.

LV warning threshold, interrupt settings and LV reset threshold are configured through IDE, as shown in Figure 32-2.

LVR Config sets low voltage reset threshold, LVW Interrupt En enables low voltage interrupt, and LVW Config sets low voltage warning threshold.

Figure 32-4 Configurations of LV Reset Threshold, LV Interrupt and LV Warning Threshold



32.2.3 LVD Registers

32.2.3.1 LVSR (0xDB)

Bit	7	6	5	4	3	2	1	0
Name	RSV		EXT0CFG			RSV	LVWF	LVWIF
Type	-	-	R/W	R/W	R/W	-	R	R/W0
Reset	-	-	0	0	0	-	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5:3]	EXT0CFG	INT0 Pin Selection 000: P0.0 001: P0.1 010: P0.2 100: P1.1 101: P0.5 110: P0.6 111: CMP4 Output
[2]	RSV	Reserved
[1]	LVWF	VCC Low Voltage Flag This bit indicates whether the chip is in the low voltage state. 0: The chip is not in LV warning state. 1: The chip is in LV warning state.
[0]	LVWIF	VCC Low Voltage Interrupt Flag Read: 0: No Interrupt Pending 1: Interrupt Pending Write: 0: This bit is cleared to “0” 1: No effect Note: This bit is not set to “1” by hardware when LVD interrupt is disabled.

32.2.3.2 CCFG2 (0x401D)

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

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

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	VBB_VSEL	VBB LDO Output Voltage 0: 10V 1: 12V
[3:0]	RSV	Reserved

33 Flash

33.1 Flash Introduction

The chip provides 32k bytes of Flash space. It supports page erase, page pre-programing and write.

Main features:

- 128 sectors in total, each with a size of 256 bytes
- 16 pages in total, each with 8 sectors
- Last sector (address range: 0x7F00 ~ 0x7FFF) cannot be erased at any time
- 120ms ~ 150ms for page erase
- Programming is enabled when FLA_CR[FLAEN] is set to “1”, where page pre-programming, page erase or write and other Flash operations are activated with MOVX instructions.

33.2 Flash Operations

- Flash memory must be unlocked before erase and programming operations. The Flash software programming feature is activated after “0x5A” and “0x1F” are written to register FLA_KEY in sequence. If the sequence is incorrect or other values are written, Flash space is frozen until the next reset. After unlocking, any write to the FLA_CR register causes the FLA_KEY to be locked again.
- CRC results change if Flash memory is rewritten during program execution.
- Page pre-programming must be done before page erase.
- Configuring FLA_CR = 0x23 enables page erase, FLA_CR = 0x25 enables page pre-programming and FLA_CR = 0x21 enables write operations.

Note:

All interrupts must be disabled before self-programming to ensure the security of Flash operations and avoid mis-operation of Flash using MOVX instruction during interrupt processing.

33.3 Flash Registers

33.3.1 FLA_CR (0x85)

Bit	7	6	5	4	3	2	1	0
Name	RSV		FLAPAGE	FLAERR	RSV	FLAPRE	FLAERS	FLAEN
Type	-	-	R/W	R	-	R/W	R/W	R/W
Reset	-	-	0	0	-	0	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	FLAPAGE	Page Operation Enable 0: Disable 1: Enable
[4]	FLAERR	Programming Error Flag 0: Programming or pre-programming succeeds. 1: Programming or pre-programming fails.
[3]	RSV	Reserved
[2]	FLAPRE	Pre-programming Enable 0: Disable 1: Enable Note: FLA_CR[FLAPRE] is valid only when FLA_CR[FLAEN] = 1.
[1]	FLAERS	Erase Enable 0: Disable 1: Enable Note: FLA_CR[FLAERS] is valid only when FLA_CR[FLAEN] = 1.
[0]	FLAEN	Programming Enable 0: Disable 1: Enable

33.3.2 FLA_KEY (0x84)

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

Bit	Name	Description
[7:0]	FLA_KEY	Write: Write “0x5A” and “0x1F” in sequence to unlock Flash operations. Write any value to FLA_CR to lock Flash operations.

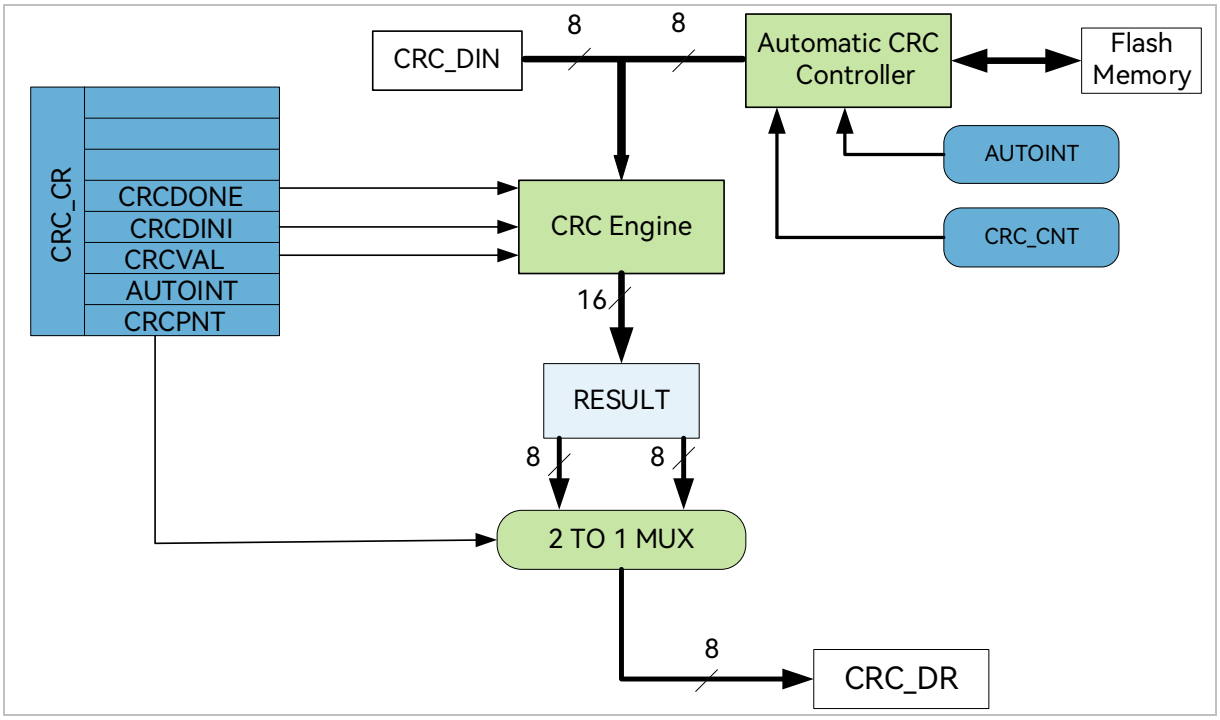
Bit	7	6	5	4	3	2	1	0
Name	RSV						FLAKSTA	
Type	-	-	-	-	-	-	R	R
Reset	-	-	-	-	-	-	0	0

Bit	Name	Description
[7:2]	RSV	Reserved
[1:0]	FLAKSTA	Read: Flash Release Status 00: Locked 01: 0x5A is written, waiting for 0x1F 10: Frozen 11: Unlocked

34 CRC

34.1 Functional Block Diagram

Figure 34-1 Functional Block Diagram of CRC Module



CRC module outputs the result of CRC calculation for any 8-bit data based on a fixed polynomial. As shown in Figure 34-1, CRC receives the 8-bit data from CRC_DIN and sends the 16-bit result to the internal register after the calculation is completed. The result can be indirectly accessed through CRC_CR[CRCPNT] and CRC_DR.

34.2 CRC16 Polynomial

The chip uses CRC16-CCITT-FALSE polynomial.

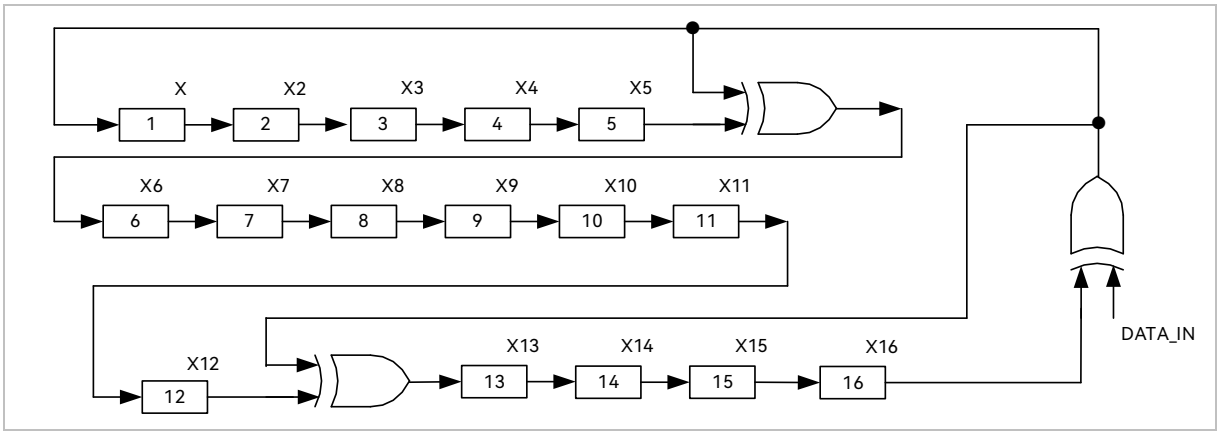
Table 34-1 CRC Criteria and Polynomial

CRC Criteria	Polynomial	Hexadecimal Representation
CRC16-CCITT-FALSE	x^16+x^12+x^5+1	0x1021

34.3 CRC16 Logic Diagram

The schematic diagram of CRC16 is shown in Figure 34-2. The chip implementation is based on parallel algorithm. For each input byte, MCU calculates the results within one system clock cycle.

Figure 34-2 CRC16 Schematic Diagram



34.4 CRC Operations

34.4.1 CRC Calculation of Single Byte

CRC of a single byte is calculated as follows:

1. Initialize CRC_DR with two options: Configure CRC_CR[CRCVAL] and set CRC_CR[CRCDINI] to “1”, with an initial value of 0x0000 or 0xFFFF. Or configure CRC_CR[CRCPNT] and CRC_DR, where any initial value can be set.
2. Write data to CRC_DIN, and the CRC calculation is completed in the next clock cycle;
3. Read CRC results: Configure CRC_CR[CRCPNT] = 1, and read off CRC_DR in software to gets the high bytes. Configure CRC_CR[CRCPNT] = 0, and read off CRC_DR to get the low bytes.

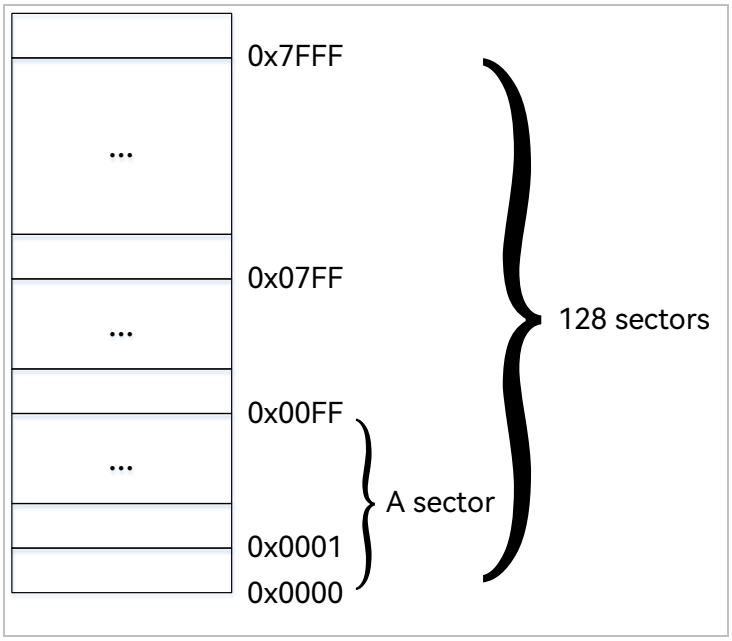
34.4.2 CRC Calculation of ROM Sector

CRC of a continuous area of data in the ROM is calculated as follows:

1. Initialize CRC_DR in the same way as that of single-byte CRC calculation;
2. Configure CRC_BEG to define starting sector of the ROM to be calculated;

3. Configure CRC_CNT to set the offset from the starting sector to the ending sector;
4. Write “1” to CRC_CR[AUTOINT] and keep other bits unchanged. The calculation starts automatically;
5. Read the CRC results.

Figure 34-3 ROM Sectors



As shown in Figure 34-3, ROM contains 32k bytes and is divided into 128 sectors, numbered from sector0 to sector127. Each sector contains 256 bytes. For CRC calculation of sectors, the value of CRC_BEG (the starting sector) can be any value falling between 0x00 and 0xFF, including 0x00 and 0x7F. The CRC_CNT (total number of sectors to be calculated) can be any value between 0x00 ~ 0x7F, including 0x00 and 0xFF.

As CRC_BEG increases, CRC_CNT decreases accordingly. For example, if CRC_BEG is 0x7F, CRC_CNT can be 0x00 only, i.e., the CRC value of the data in the last sector is calculated. In this case, if CRC_CNT is large, CRC controller will automatically limits the number of sectors to be calculated. Finally, CRC module only calculates CRC value of the last sector.

34.5 CRC Registers

34.5.1 CRC_CR (0x4022)

Bit	7	6	5	4	3	2	1	0
Name	RSV			CRCDONE	CRCDINI	CRCVAL	AUTOINT	CRCPNT
Type	-	-	-	R	W1	R/W	W1	R/W
Reset	-	-	-	1	0	0	0	0

Bit	Name	Description
[7:5]	RSV	Reserved
[4]	CRCDONE	CRC Sector Calculation Completion Flag During the calculation, this bit is automatically set to “0” and the software program stops. In other cases, this bit is automatically set to “1” by the hardware, so the software always returns “1” when reading this bit.
[3]	CRCDINI	CRC Result Initialization Trigger 0: No effect 1: CRC result initialization is triggered
[2]	CRCVAL	CRC Result Initialization Selection 0: CRC result is initialized to 0x0000 1: CRC result is initialized to 0xFFFF
[1]	AUTOINT	CRC Sector Calculation Launch 0: No effect 1: Launch CRC batch calculation See section CRC Calculation of ROM Sector
[0]	CRCPNT	CRC Result Pointer 0: Read CRC_DR to access 8 low-order bits of the 16-bit CRC result 1: Read CRC_DR to access 8 high-order bits of the 16-bit CRC result

Note:

CRC_CR[AUTOINT] is set to “0” to perform single-byte CRC checksum.

34.5.2 CRC_DIN (0x4021)

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

Bit	Name	Description
[7:0]	CRC_DIN	CRC Input Data When 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:  Note: It is a virtual register, so the written data is not saved. 0x00 is returned when the address is accessed.

34.5.3 CRC_DR (0x4023)

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

Bit	Name	Description
[7:0]	CRC_DR	CRC Result Output When this register is read or written, CRC_CR[CRCPNT] setting determines to access 8 high-order bits or 8 lower-order bits of the CRC result.

34.5.4 CRC_BEG (0x4024)

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

Bit	Name	Description
[7]	RSV	Reserved
[6:0]	CRC_BEG	First ROM Sector Pending Auto CRC Calculation Example: If CRC_BEG is set to “1”, CRC calculation starts from location 1*256 = 256, or rather from the first byte of sector 2.

34.5.5 CRC_CNT (0x4025)

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

Bit	Name	Description
[7]	RSV	Reserved
[6:0]	CRC_CNT	Offset of Sector Pending Automatic CRC Calculation This bit defines the offset of ROM sector for CRC calculation and determines the last sector pending CRC calculation.

35 Sleep Mode

35.1 Introduction

The chip operates in three modes: normal mode, standby mode and sleep mode. These modes are selected by setting PCON[IDLE] and PCON[STOP].

The operating states of the module under different power modes are summarized in Table 35-1.

Table 35-1 Power Consumption Modes

Power Mode	Description	Wakeup Source	Power Consumption Performance
Normal	All modules work at full speed except for peripherals that are disabled	NA	High power consumption with best performance
Standby	CPU clock stops and other functional modules are either enabled or disabled depending on their control bit setting. WDT stops.	Any interrupt Reset/Debug on external interrupt	Low power performance with flexible performance
Sleep	Flash Deep Sleep. The analog fast clock circuit is disconnected and software shall be operated to ensure that ADC, FOC, and driver modules are idle 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_();
_nop_();
```

35.2 Sleep Mode Register

35.2.1 PCON(0x87)

Bit	7	6	5	4	3	2	1	0
Name	RSV		GF3	GF2	GF1	RSV	STOP	IDLE
Type	-	-	R/W	R/W	R/W	-	R/W	R/W
Reset	-	-	0	0	0	-	0	0

Bit	Name	Description
[7:6]	RSV	Reserved
[5]	GF3	General-purpose flag bit 3
[4]	GF2	General-purpose flag bit 2
[3]	GF1	General-purpose flag bit 1
[2]	RSV	Reserved
[1]	STOP	A write of “1” makes the chip enter the sleep mode. This bit is automatically cleared to “0” by hardware after wakeup.
[0]	IDLE	A write of “1” makes the chip enter the standby mode. This bit is automatically cleared to “0” by hardware after wakeup.

Power Consumption Mode PCON[STOP : IDLE]:

00: Normal

01: Standby

1X: Sleep

36 Code Protection

36.1 Introduction

The chip supports Flash space encryption to protect your software intellectual property and avoid unauthorized access. When Flash memory is encrypted, the data inside cannot be read, and data consistency can be verified by CRC check module only.

36.2 Operating Instructions

Figure 36-1 Code Protection Configurations

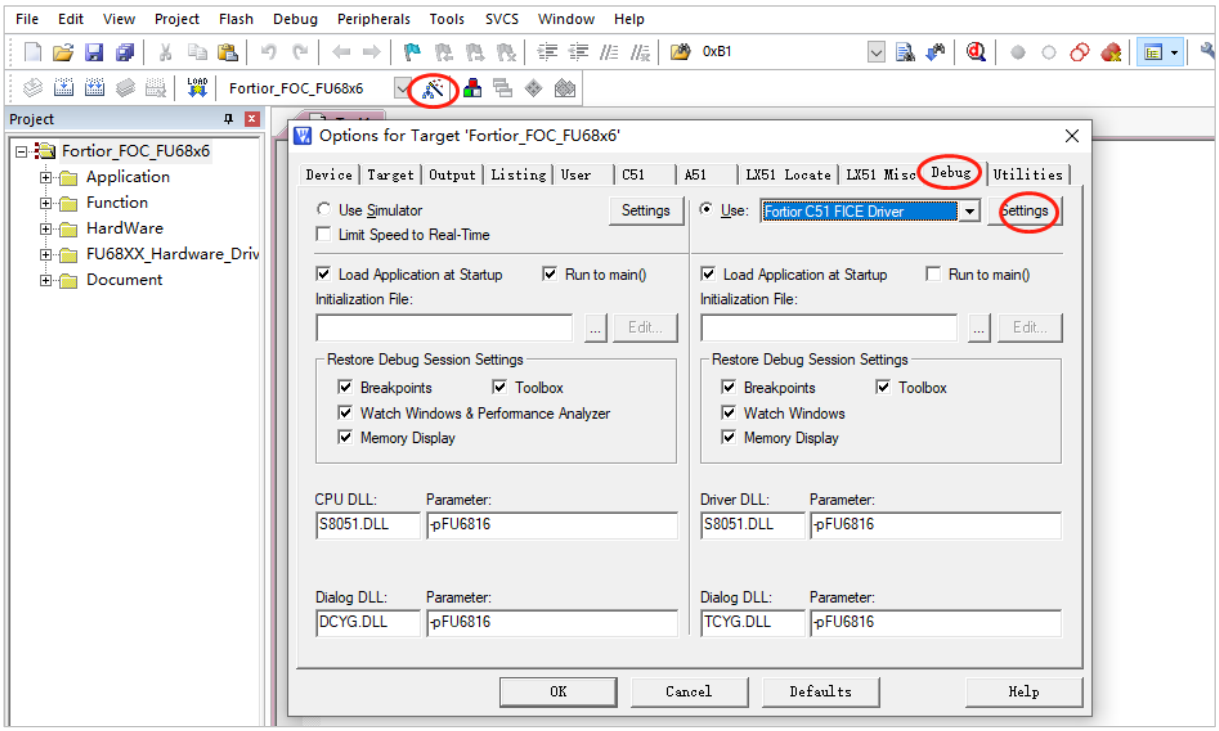
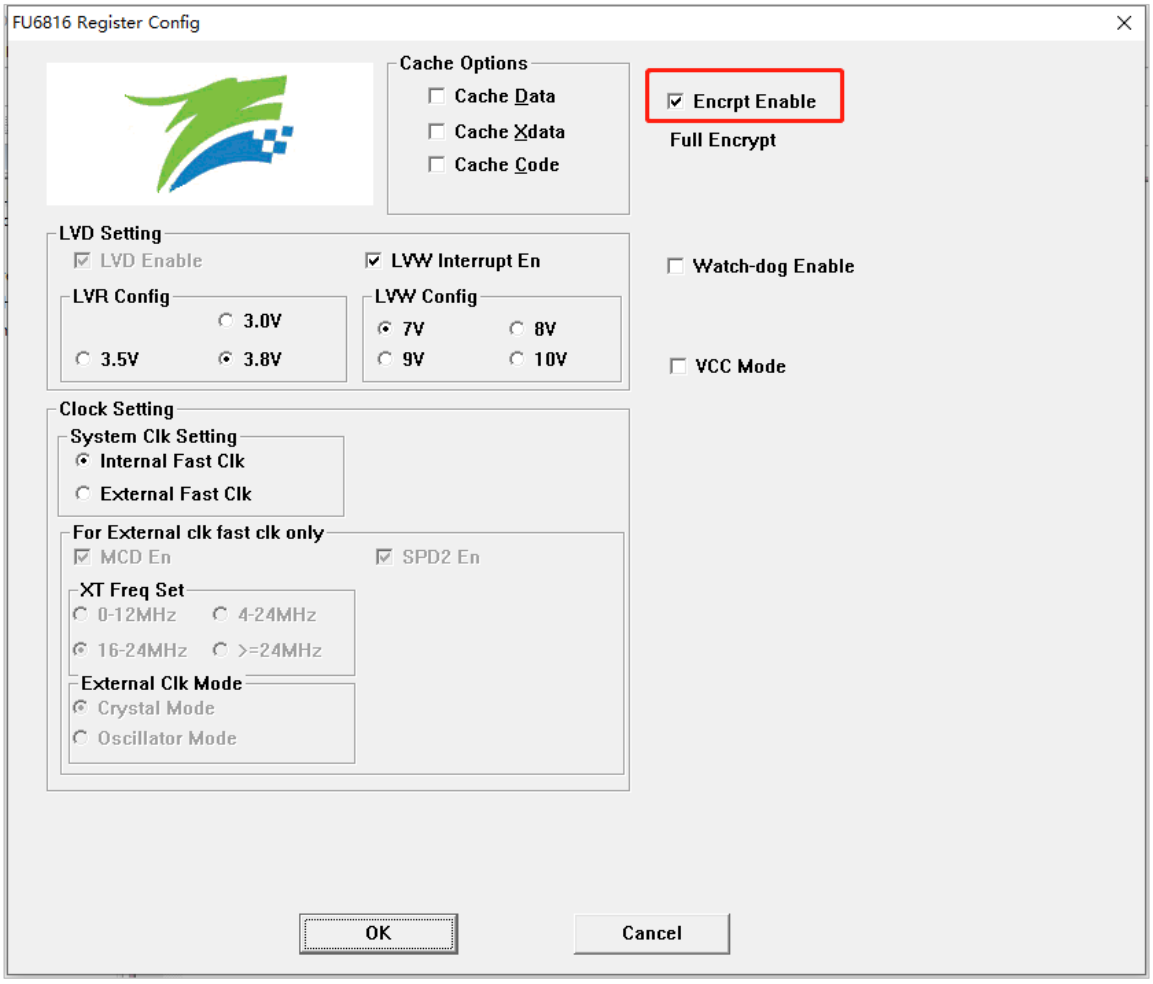


Figure 36-2 Full Code Protection Mode



Operation steps are as follows:

1. Start 8051 IDE, enter Target Options and select Debug tab. As shown in Figure 36-1, click Settings to proceed with the setting;

2. Select the options as shown in Figure 36-2, and click OK. Then compile the project and download it. Get the BIN file and program it to Flash.

37 Magnetoresistive (MR) Sensor

37.1 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.

37.2 Instructions

Figure 37-1 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 37-2 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 37-1 Position Relationship between Sensor Chip and Magnet

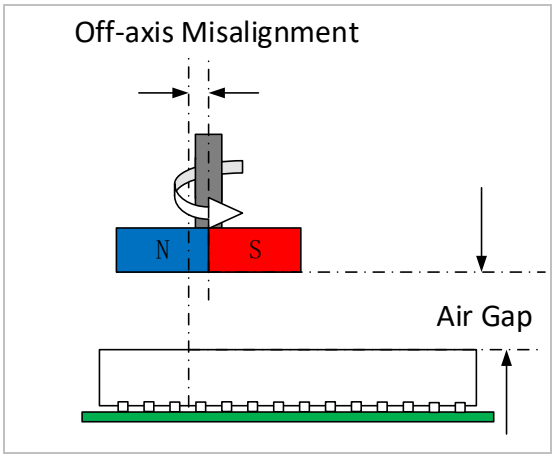
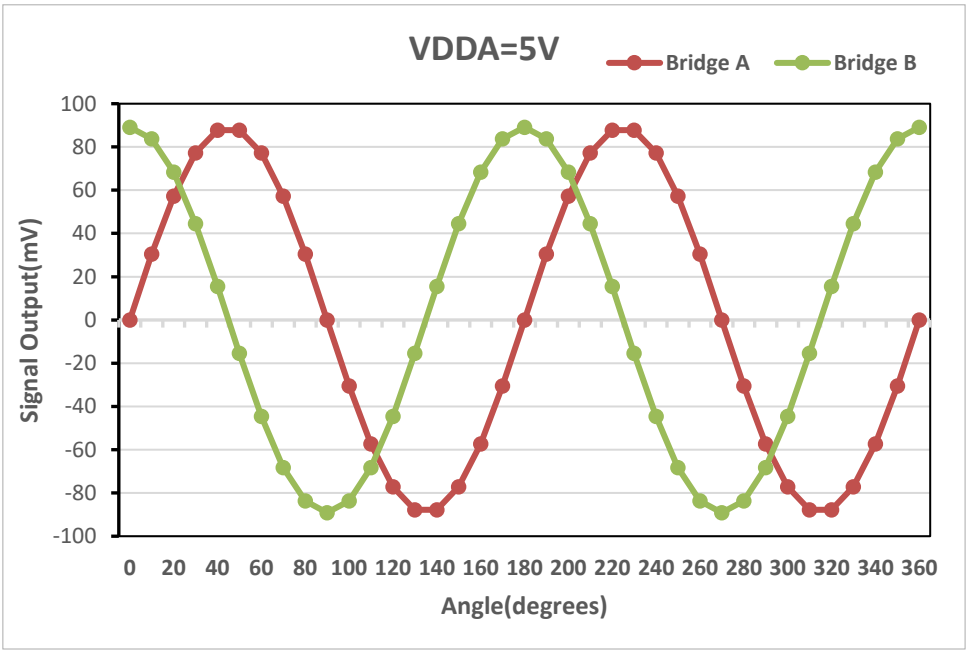


Table 37-1 Electrical and Magnetic Characteristics

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	-	%/℃
TCmag2	SmCo	-	-0.035	-	%/℃

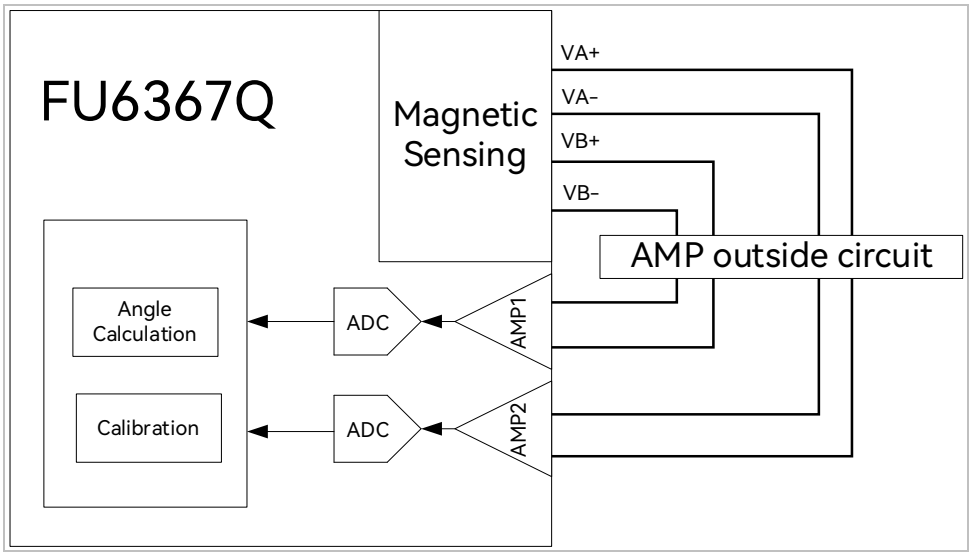
Figure 37-2 Voltage Waveform of Signal Output and Angle



37.3 Functional Block Diagram

A simplified block diagram of the chip application is shown in Figure 37-3. 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 gains. ADC and hardware decoder are used for angular decoding.

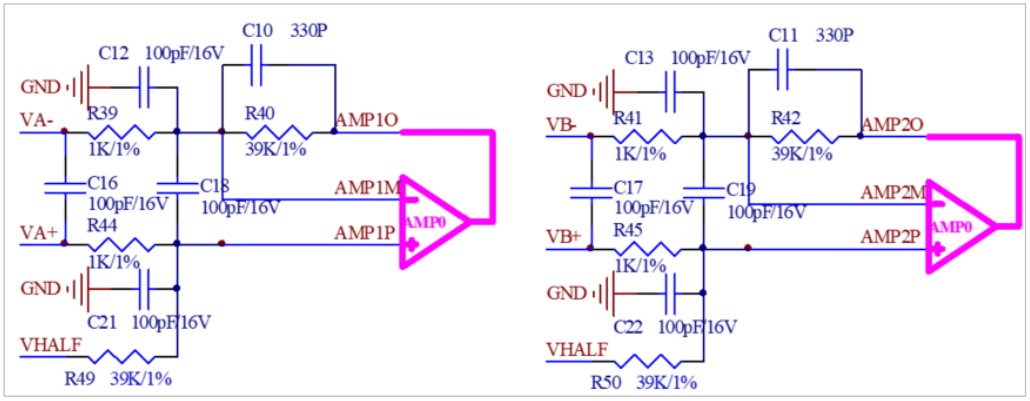
Figure 37-3 Functional Block Diagram of MR Sensor



37.4 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 37-4 Recommended Circuit Diagram



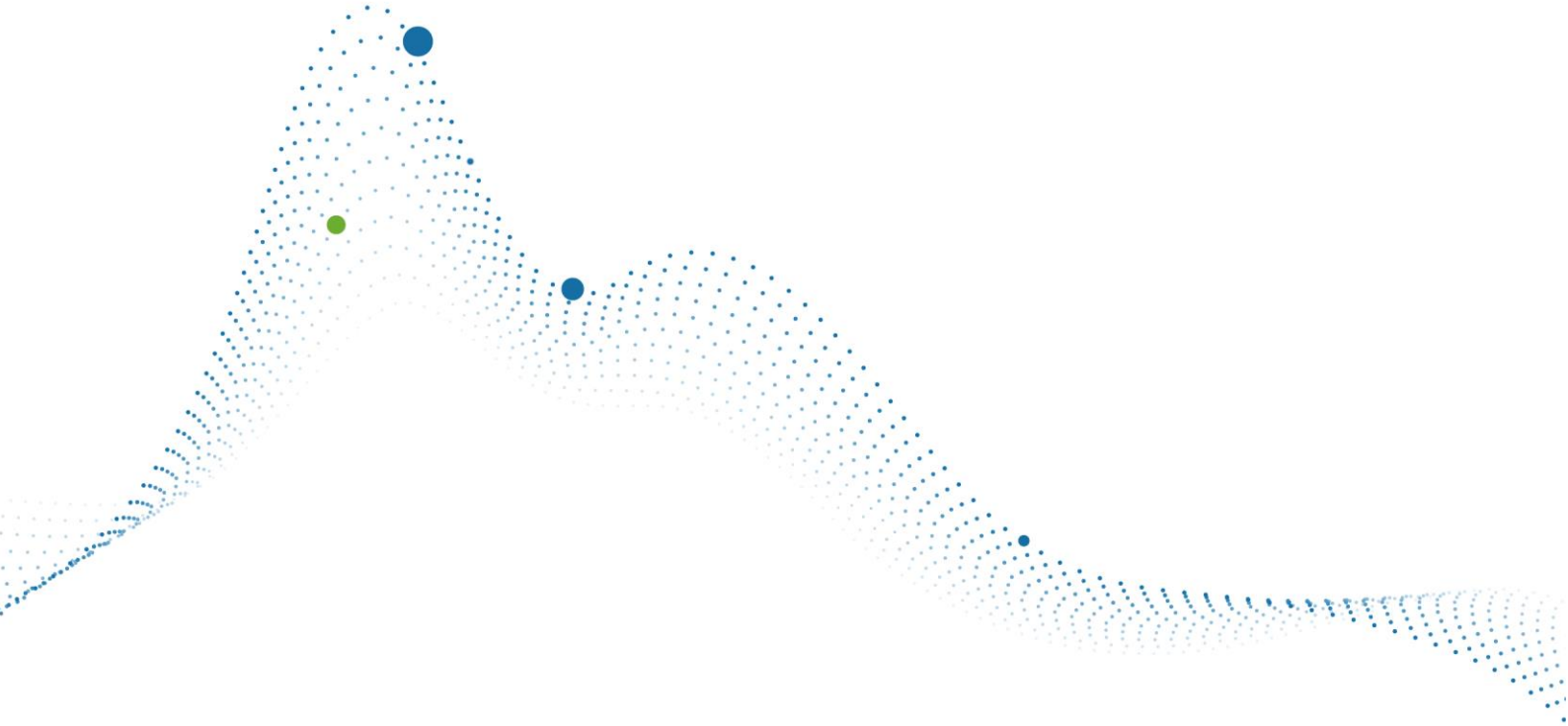
38 Revision History

Rev.	Description	Date	Prepared By
V1.0	First release, translated from Chinese version V1.0	2024/08/13	Eric Deng

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

Tel.: 0755-26867710
Fax: 0755-26867715
P.C.: 518057
Web: www.fortiortech.com



The content of this document

Copyright by Fortior Technology (Shenzhen) Co., Ltd. All Rights Reserved.