Cortex®-M0+ based 32-Bit General-Purpose Microcontroller

UM Rev. 1.00

Introduction

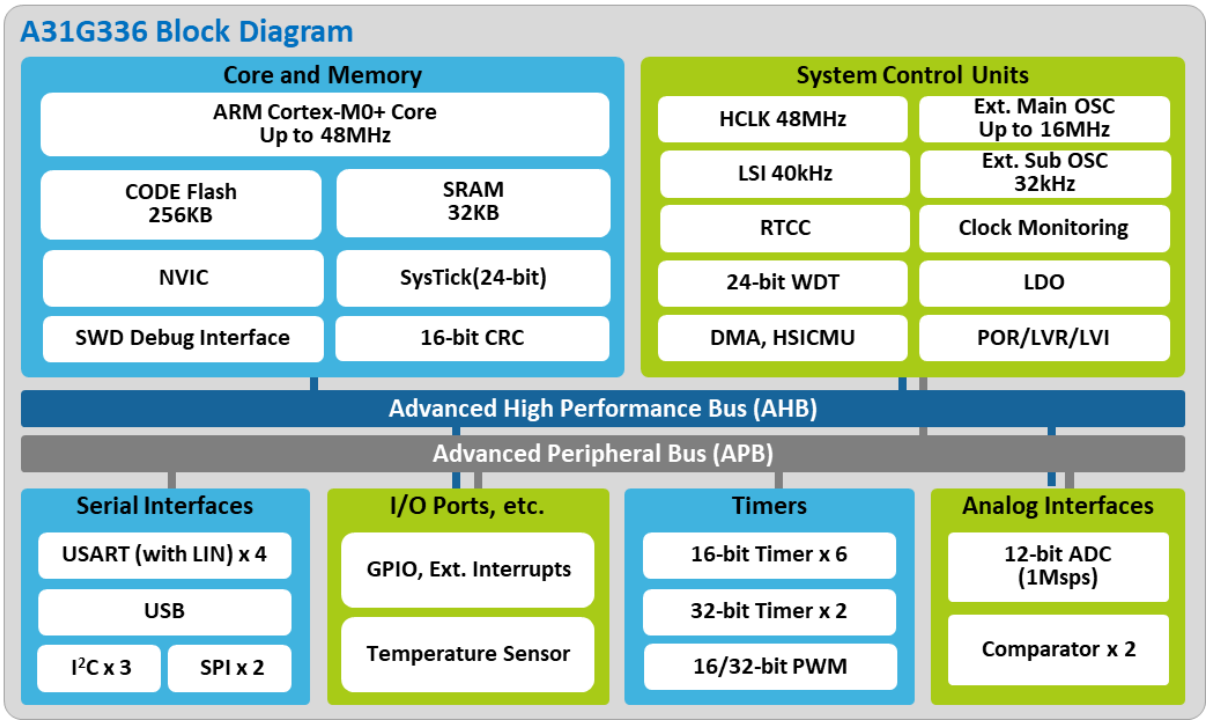
This user's manual contains complete information for application developers who use A31G336 for their specific needs.

The A31G336 is a 32-bit general-purpose microcontroller for various appliances. To meet the requirements for complexity and high performance in consumer electronics, the A31G336 incorporates Arm's high-speed 32-bit Cortex-M0+ core and has up to 256 KB of Flash memory with ECC and 32 KB of SRAM.

As shown in the following figure, the A31G336 has various peripherals such as 16-bit timers, 32-bit timers, 12-bit ADC, CRC generator, USART, I2C, SPI, DMA, Real timer and calendar, Comparator, etc. The A31G336 also has a POR, LVR, LVI, and an internal RC oscillator.

The A31G336 supports sleep and deep sleep modes to reduce power consumption.

Figure 1. A31G336 Block Diagram



Reference Document

- Document '[DDI 0484C]' is provided by ARM and contains information of Cortex-M0+.
- A31G336 Datasheet is provided by ABOV and available at www.abovsemi.com.

Contents

Introduction 1
Reference Document 1
1. Description 17
1.1 Product Category Definition 17
1.2 Availability of Peripherals 18
2. Pinouts and Pin Descriptions 24
2.1 Pinouts 24
2.1.1 A31G336RL (64-LQFP-1010) 24
2.1.2 A31G336RM (64-LQFP-1212) 25
2.1.3 A31G336CL (48-LQFP) 26
2.1.4 A31G336CU (48-QFN) 27
2.1.5 A31G336IU (40-QFN) 28
2.1.6 A31G336KU (32-QFN) 29
2.2 Pin Description 30
3. System and Memory Overview 37
3.1 System Architecture 37
3.1.1 Cortex-M0+ Core 38
3.1.2 Interrupt Controller 38
3.2 Memory Organization 40
3.2.1 Introduction 40
3.2.2 Memory Map and Register Boundary Address 40
3.3 Memory Map 42
3.3.1 Embedded SRAM 42
3.3.2 Flash Memory Overview 42
3.4 Boot Configuration 42
4. System Control Unit (SCU) 43
4.1 SCU Introduction 43
4.2 SCU Main Features 43
4.3 Clocks 44
4.3.1 External Oscillation Clock 44
4.3.2 Internal Oscillation Clock 44
4.3.3 PLL Clock 45
4.3.3.1 PLL Output Frequency Settings 46
4.4 SCU Block Diagram 47
4.5 Clock System 48
4.5.1 HCLK Clock Domain 49
4.5.2 Miscellaneous Clock Domain 50
4.5.3 PCLK Clock Domain 50
4.5.4 Clock Configuration Procedure 51
4.6 Reset 53
4.6.1 Overview 53
4.6.2 Cold Reset 54
4.6.3 Warm Reset 55
4.6.4 LVR Reset 56
4.7 Operation Mode 57
4.7.1 RUN Mode 57
4.7.2 SLEEP Mode 57

4.7.3 DEEP SLEEP Mode 57
4.8 Pins for SCU 59
4.9 SCU Registers 60
4.9.1 SCU_VENDORID: Vendor ID Register 62
4.9.2 SCU_CHIPID: Chip ID Register 62
4.9.3 SCU_REVNR: Revision Number Register 63
4.9.4 SCU_PMREMAP: Program Memory Remap Register 63
4.9.5 SCU_BTPSCR: Boot Pin Status and Control Register 64
4.9.6 SCU_RSTSSR: Reset Source Status Register 65
4.9.7 SCU_NMISRCR: NMI Source Selection Register 67
4.9.8 SCU_SWRSTR: Software Reset Register 68
4.9.9 SCU_SRSTVR: System Reset Validation Register 68
4.9.10 SCU_WUTCR: Wake-up Timer Control Register 69
4.9.11 SCU_WUTDR: Wake-up Timer Data Register 70
4.9.12 SCU_HSITRM: High-Speed Internal RC Trim Register 71
4.9.13 SCU_LSITRM: Low-Speed Internal RC Trim Register 72
4.9.14 SCU_SCCR: System Clock Control Register 73
4.9.15 SCU_CLKSRCR: Clock Source Control Register 74
4.9.16 SCU_SCDIVR1: System Clock Divide Register 1 75
4.9.17 SCU_SCDIVR2: System Clock Divide Register 2 76
4.9.18 SCU_CLKOCR: Clock Output Control Register 77
4.9.19 SCU_CMONCR: Clock Monitoring Control Register 78
4.9.20 SCU_PPCLKEN1: Peripheral Clock Enable Register 1 80
4.9.21 SCU_PPCLKEN2: Peripheral Clock Enable Register 2 81
4.9.22 SCU_PPCLKSR: Peripheral Clock Selection Register 82
4.9.23 SCU_PPRST1: Peripheral Reset Register 1 83
4.9.24 SCU_PPRST2: Peripheral Reset Register 2 85
4.9.25 SCU_HSECR: External Main Oscillator Control Register 87
4.9.26 SCU_LSECR: External Sub Oscillator Control Register 88
4.9.27 SCU_PLLCON: PLL Clock Control Register 89
4.9.28 SCU_VUSB33CR: VUSB33 Control Register 90
4.9.29 SCU_ECCIESR: Flash ECC Interrupt Enable and Status Register 91
4.9.30 SCU_LVICR: Low-Voltage Indicator Control Register 92
4.9.31 SCU_LVRCR: Low-Voltage Reset Control Register 93
4.9.32 SCU Register Map Summary 94
5. Port Control Unit (PCU) and GPIO 97
5.1 Introduction 97
5.1.1 PCU Introduction 97
5.1.2 GPIO Introduction 97
5.2 Main Features 98
5.2.1 PCU Features 98
5.2.2 GPIO Features 98
5.3 GPIO Functional Description 99
5.3.1 GPIO Pins and Internal Signals 101
5.3.2 I/O Port Control Registers 101
5.3.3 I/O Port Data Registers 101
5.3.4 I/O Data Bitwise Handling 101
5.3.5 I/O Alternate Function 102
5.3.6 Interrupt Functionality 104
5.3.7 Output Configuration 105

5.3.8 Input Configuration 106
5.3.9 Alternate Function Configuration 107
5.3.10 Debouncing Functionality 107
5.4 GPIO Registers 108
5.4.1 Pn_MOD: Port n Mode Register 109
5.4.2 Pn_TYP: Port n Output Type Selection Register 110
5.4.3 Pn_AFSR1/2: Port n Alternate Function Selection Register 1/2 111
5.4.4 Pn_PUPD: Port n Pull-Up/Pull-Down Resistor Selection Register 113
5.4.5 Pn_INDR: Port n Input Data Register 114
5.4.6 Pn_OUTDR: Port n Output Data Register 114
5.4.7 Pn_BSR: Port n Output Bit Set Register 115
5.4.8 Pn_BCR: Port n Output Bit Clear Register 115
5.4.9 Pn_OUTDMSK: Port n Output Data Mask Register 116
5.4.10 Pn_DBCR: Port n Debounce Control Register 117
5.4.11 PCU and GPIO Register Map Summary 118
5.5 Interrupt Registers 119
5.5.1 INTC_PnTRIG: Port n Interrupt Trigger Selection Register 120
5.5.2 INTC_PnCR: Port n Interrupt Control Register 120
5.5.3 INTC_PnFLAG: Port n Interrupt Flag Register 121
5.5.4 INTC_EINTnCONF1: External Interrupt Configuration Register 1 121
5.5.5 INTC_EINTnCONF2: External Interrupt Configuration Register 2 122
5.5.6 INTC_MSK: Interrupt Source Mask Register 124
5.5.7 Interrupt Register Map Summary 125
6. Embedded Flash Memory 126
6.1 Introduction 126
6.2 Flash Memory Main Features 127
6.2.1 Procedure For Flash Memory Operation 129
6.2.1.1 Flash Page Erase Procedure (512 bytes) 130
6.2.1.2 Flash Sector Erase Procedure (2048 bytes) 130
6.2.1.3 Flash Write Procedure 131
6.2.1.4 Flash Bulk Erase Procedure 131
6.2.1.5 Flash Bulk Erase Procedure Including Boot Protection Block 132
6.3 Flash Memory Controller Registers 133
6.3.1 FMC_ADR: Flash Memory Address Register 134
6.3.2 FMC_IDR1: Flash Memory Identification Register 1 134
6.3.3 FMC_IDR2: Flash Memory Identification Register 2 135
6.3.4 FMC_CR: Flash Memory Control Register 136
6.3.5 FMC_BCR: Flash Memory Boot Protection Block Bulk Erase Control Register 137
6.3.6 FMC_ERFLAG: Flash Memory Error Flag Register 137
6.3.7 FMC_FETCHCR: Flash Memory Data Fetch Control Register 138
6.3.8 FMC Register Map Summary 139
6.4 Configure Option and Manufacturer Areas 140
6.5 Manufacturer Area 141
6.5.1 CONF_MF1CNFIG: Configuration for Manufacture Information 1 142
6.5.2 CONF_MF2CNFIG: Configuration for Manufacture Information 2 142
6.5.3 CONF_MF3CNFIG: Configuration for Manufacture Information 3 143
6.5.4 CONF_MF4CNFIG: Configuration for Manufacture Information 4 143
6.5.5 CONF_MFCONFIRM: Manufacture Information Confirm Data 144
6.5.6 Manufacturer Area Register Map Summary 145
6.6 Configure Option Pages 146

6.6.1 TS_FREQ_T30: Temperature Sensor Output Frequency 1 147
6.6.2 TS_FREQ_T85: Temperature Sensor Output Frequency 2 147
6.6.3 CONF_RPCNFIG: Configuration for Read Protection 148
6.6.4 CONF_WDTCNFIG: Configuration for Watchdog Timer 149
6.6.5 CONF_LVRCNFIG: Configuration for Low-Voltage Reset 150
6.6.6 CONF_CNFIGWTP1: Configure and User Data Page Erase/Write Protection 1 151
6.6.7 CONF_FMWTP1: Erase/Write Protection 1 for Flash Memory 152
6.6.8 CONF_FMWTP2: Erase/Write Protection 2 for Flash Memory 152
6.6.9 CONF_FMWTP3: Erase/Write Protection 3 for Flash Memory 153
6.6.10 CONF_FMWTP4: Erase/Write Protection 4 for Flash Memory 153
6.6.11 Configure Option Page Register Map Summary 154
7. Direct Memory Access Controller (DMA) 155
7.1 DMA Introduction 155
7.2 DMA Registers 156
7.2.1 DMACHn_CR: DMA Channel n Controller Register 157
7.2.2 DMACHn_IESR: DMA Channel n Interrupt Enable and Status Register 159
7.2.3 DMACHn_PAR: DMA Channel n Peripheral Address Register 160
7.2.4 DMACHn_MAR: DMA Channel n Memory Address Register 160
7.2.5 DMA Register Map Summary 161
8. Watchdog Timer (WDT) 162
8.1 WDT Introduction 162
8.2 WDT Main Features 162
8.3 WDT Block Diagram 163
8.4 WDT Functional Description 164
8.4.1 Timing Diagram 164
8.4.2 Pre-Scale Table 165
8.5 WDT Registers 166
8.5.1 WDT_CR: Watchdog Timer Control Register 167
8.5.2 WDT_SR: Watchdog Timer Status Register 168
8.5.3 WDT_DR: Watchdog Timer Data Register 169
8.5.4 WDT_CNT: Watchdog Timer Counter Register 169
8.5.5 WDT_WINDR: Watchdog Timer Window Data Register 170
8.5.6 WDT_CNTR: Watchdog Timer Counter Reload Register 170
8.5.7 WDT Register Map Summary 171
9. REAL TIME CLOCK AND CALENDAR (RTCC) 172
9.1 RTCC Introduction 172
9.2 RTCC Main Features 172
9.3 RTCC Block Diagram 173
9.4 RTCC Registers 174
9.4.1 RTC_CR: RTCC Control Register 175
9.4.2 RTC_ECR: RTCC Time Error Correction Register 177
9.4.3 RTC_SCNT: RTCC Sub Control Register 178
9.4.4 RTC_SEC: Second Counter Register 178
9.4.5 RTC_MIN: RTCC Minute Counter Register 179
9.4.6 RTC_HOUR: RTCC Hour Counter Register 180
9.4.7 RTC_DAY: RTCC Day Counter Register 182
9.4.8 RTC_WEEK: RTCC Week Counter Register 183
9.4.9 RTC_MONTH: RTCC Month Counter Register 184
9.4.10 RTC_YEAR: RTCC Year Counter Register 185

9.4.11 RTC_ALMIN: RTCC Alarm Minute Counter Register 185
9.4.12 RTC_ALHOUR: RTCC Alarm Hour Counter Register 186
9.4.13 RTC_ALWEEK: RTCC Alarm Week Counter Register 187
9.4.14 RTCC Register Map Summary 188
10. Timer Counter 10/11/12/13/14/15 189
10.1 Timer Counter 10/11/12/13/14/15 Introduction 189
10.2 Timer Counter 10/11/12/13/14/15 Main Features 189
10.3 Timer Counter 10/11/12/13/14/15 Functional Description 190
10.3.1 Block Diagram 190
10.4 Pin Description for Timer Counter 10/11/12/13/14/15 190
10.5 Functional Description 191
10.5.1 Timer Counter 10/11/12/13/14/15 191
10.5.2 16-bit Timer/Counter Mode 192
10.5.3 16-bit Capture Mode 194
10.5.4 16-bit PPG Mode 196
10.6 TIMER10/11/12/13/14/15 Registers 198
10.6.1 TIMERn_CR: Timer /Counter n Control Register 199
10.6.2 TIMERn_ADR: Timer/Counter n A Data Register 201
10.6.3 TIMERn_BDR: Timer/Counter n B Data Register 201
10.6.4 TIMERn_CAPDR: Timer/Counter n Capture Data Register 202
10.6.5 TIMERn_PREDR: Timer/Counter n Prescaler Data Register 202
10.6.6 TIMERn_CNT: Timer/Counter n Counter Register 203
10.6.7 TIMERn Register Map Summary 204
11. Timer Counter 20/21 205
11.1 Timer Counter 20/21 Introduction 205
11.2 Timer Counter 20/21 Main Features 205
11.3 Timer Counter 20/21 Functional Description 206
11.3.1 Block Diagram 206
11.4 Pin Description for Timer Counter 20/21 206
11.5 Functional Description 207
11.5.1 Timer Counter 20/21 207
11.5.2 32-bit Timer/Counter Mode 208
11.5.3 32-bit Capture Mode 210
11.5.4 32-bit PPG Mode 212
11.6 TIMER 20/21 Registers 214
11.6.1 TIMER2n_CR: Timer/Counter 2n Control Register 215
11.6.2 TIMER2n_ADR: Timer/Counter 2n A Data Register 217
11.6.3 TIMER2n_BDR: Timer/Counter 2n B Data Register 217
11.6.4 TIMER2n_CAPDR: Timer/Counter 2n Capture Data Register 218
11.6.5 TIMER2n_PREDR: Timer/Counter 2n Prescaler Data Register 218
11.6.6 TIMER2n_CNT: Timer/Counter 2n Counter Register 219
11.6.7 TIMER20 Register Map Summary 220
11.6.8 TIMER21 Register Map Summary 221
12. USART (UART and SPI) 222
12.1 Introduction 222
12.2 Main Features 222
12.3 USART 10/11/12/13 Block Diagram 223
12.4 Pin Description for USART 10/11/12/13 224
12.5 UART Functional Description 225
12.5.1 USART Clock Generation 225

12.5.2 External Clock (SCKn) 226
12.5.3 Synchronous Mode Operation 227
12.5.4 UART Data Format 228
12.5.5 UART Parity Bit 229
12.5.6 UART Transmitter 230
12.5.6.1 UART Sending TX Data 230
12.5.6.2 UART Transmitter Flag and Interrupt 230
12.5.6.3 UART Parity Generator 231
12.5.6.4 UART Disabling Transmitter 231
12.5.7 UART Receiver 232
12.5.7.1 UART Receiving RX Data 232
12.5.7.2 UART Receiver Flag and Interrupt 232
12.5.7.3 UART Parity Checker 233
12.5.7.4 UART Disabling Receiver 233
12.5.7.5 Asynchronous Data Reception 233
12.5.7.6 Receive Timeout Function 235
12.5.7.7 UART Auto Baud Rate Detection 236
12.5.8 SPI Mode 237
12.5.9 SPI Clock Formats and Timing 238
12.5.10 Local Interconnection Network (LIN) Mode 240
12.6 USART Registers 242
12.6.1 USARTn_CR1: USARTn Control Register 1 243
12.6.2 USARTn_CR2: USARTn Control Register 2 245
12.6.3 USARTn_CR3: USARTn Control Register 3 246
12.6.4 USARTn_ST: USARTn Status Register 248
12.6.5 USARTn_BDR: USARTn Baud Rate Generation Register 249
12.6.6 USARTn_RDR: USARTn Receive Data Register 249
12.6.7 USARTn_TDR: USARTn Transmit Data Register 250
12.6.8 USARTn_RTODR: USARTn Receive Timeout Data Register 250
12.6.9 USARTn_RCDR: USARTn Receive Character Detection Data Register 251
12.6.10 USART Register Map Summary 252
13. Serial Peripheral Interface (SPI) 253
13.1 Introduction 253
13.2 Main Features 253
13.3 SPI 0/1 Functional Description 254
13.3.1 Block Diagram 254
13.4 Pin Description for SPI 0/1 254
13.5 Functional Description 255
13.5.1 SPI Clock Formats and Timing 255
13.6 SPI 0/1 Registers 257
13.6.1 SPIn_CR: SPIn Control Register 258
13.6.2 SPIn_SR: SPIn Status Register 259
13.6.3 SPIn_RDR: SPIn Receive Data Register 259
13.6.4 SPIn_TDR: SPIn Transmit Data Register 260
13.6.5 SPIn_PREDR: SPIn Prescaler Data Register 260
13.6.6 SPIn Register Map Summary 261
14. I2C (Inter-Integrated Circuit) 262
14.1 Introduction 262
14.2 Main Features 262
14.3 I2C 0/1/2 Block Diagram 263

14.4 Pin Description for I2C 0/1/2 263
14.5 I2C 0/1/2 Functional Description 264
14.5.1 I2C Bit Transfer 264
14.5.2 START/Repeated START/STOP 264
14.5.3 Data Transfer 265
14.5.4 Acknowledge 266
14.5.5 Synchronization and Arbitration 267
14.6 I2C Operation 269
14.6.1 Master Transmitter 269
14.6.2 Master Receiver 271
14.6.3 Slave Transmitter 273
14.6.4 Slave Receiver 274
14.7 I2C Registers 275
14.7.1 I2Cn_CR: I2Cn Control Register 276
14.7.2 I2Cn_ST: I2Cn Status Register 277
14.7.3 I2Cn_SAR1: I2Cn Slave Address Register 1 278
14.7.4 I2Cn_SAR2: I2Cn Slave Address Register 2 278
14.7.5 I2Cn_DR: I2Cn Data Register 279
14.7.6 I2Cn_SDHR: I2Cn SDA Hold Time Register 279
14.7.7 I2Cn_SCLR: I2Cn SCL Low Period Register 280
14.7.8 I2Cn_SCHR: I2Cn SCL High Period Register 280
14.7.9 I2C Register Map Summary 281
15. High-Speed 12-bit ADC 282
15.1 Introduction 282
15.2 Main Features 282
15.3 ADC Block Diagram 283
15.4 Pin Description for ADC 284
15.5 ADC Functional Description 285
15.5.1 ADC Enable/Disable Control 285
15.5.2 Channel Selection 285
15.5.3 ADC Conversion Timing 286
15.5.4 ADC Conversion Mode 287
15.5.4.1 Single Conversion Mode 287
15.5.4.2 Sequential Conversion Mode 288
15.5.4.3 Continuous Conversion Mode 289
15.5.5 ADC Oversampling 290
15.5.6 ADC Recommend Circuit 291
15.6 ADC Registers 292
15.6.1 ADC_CR: A/D Converter Control Register 293
15.6.2 ADC_OVSCR: A/D Converter Oversampling Control Register 294
15.6.3 ADC_IESR: A/D Converter Interrupt Enable and Status Register 295
15.6.4 ADC_DR: A/D Converter Data Register 296
15.6.5 ADC_PREDR: A/D Converter Prescaler Data Register 297
15.6.6 ADC_SAMR: A/D Converter Sampling Time Register 297
15.6.7 ADC_CHSELR: A/D Converter Channel Selection Register 298
15.6.8 ADC Register Map Summary 299
16. Cyclic Redundancy Check (CRC) and Checksum 300
16.1 Introduction 300
16.2 Main Features 300
16.3 CRC and Checksum Block Diagram 301

16.4 CRC and Checksum Functional Description 302
16.4.1 CRC Polynomial Structure 302
16.4.2 The CRC Operation Procedure in Auto CRC/Checksum Mode 302
16.4.3 The CRC Operation Procedure in User CRC/Checksum Mode 303
16.5 CRC and Checksum Registers 304
16.5.1 CRC_CR: CRC Control Register 305
16.5.2 CRC_IN: CRC Input Data Register 307
16.5.3 CRC_RLT: CRC Result Data Register 307
16.5.4 CRC_INIT: CRC Initial Data Register 308
16.5.5 CRC_SADR: CRC Start Address Register 308
16.5.6 CRC_EADR: CRC End Address Register 309
16.5.7 CRC Register Map Summary 310
17. Comparator 0/1 311
17.1 Introduction 311
17.2 Main Features 311
17.3 Comparator 0/1 Block Diagram 312
17.4 Pin description for Comparator 0/1 313
17.5 Comparator 0/1 Registers 313
17.5.1 CMPn_CR: Comparator n Control Register 314
17.5.2 CMPn_SR: Comparator n Status Register 316
17.5.3 CMP Register Map Summary 317
18. Temperature Sensor (TS) 318
18.1 Introduction 318
18.2 Main Features 318
18.3 TS Block Diagram 318
18.4 TS Functional Description 319
18.4.1 Ring-Oscillator of Temperature Sensor 319
18.4.2 Frequency Counting 319
18.4.3 Temperature Calculation 320
18.5 TS Registers 321
18.5.1 TS_CR: Temperature Sensor Control Register 322
18.5.2 TS_IDR: Temperature Sensor Interval Data Register 323
18.5.3 TS_ICNTR: Temperature Sensor Interval Counter Register 323
18.5.4 TS_OUTDR: Temperature Sensor Control Register 324
18.5.5 TS Register Map Summary 325
19. High-Speed Internal OSC Clock Management Unit (HSICMU) 326
19.1 Introduction 326
19.2 Main Features 326
19.3 HSICMU Block Diagram 327
19.4 HSICMU Functional Description 328
19.4.1 Synchronization Input 328
19.4.2 Frequency Error Measurement 328
19.4.3 Frequency Error Evaluation and Automatic Trimming 329
19.4.4 HSICMU Initialization and Configuration 330
19.5 HSICMU Registers 331
19.5.1 HSICMU_CR: HSICMU Control Register 332
19.5.2 HSICMU_CFG: HSICMU Configure Register 333
19.5.3 HSICMU_ISR: HSICMU Interrupt and Status Register 334
19.5.4 HSICMU_ICR: HSICMU Interrupt and Status Clear Register 335
19.5.5 HSICMU_TRIM: HSICMU Control TRIM Register 336

19.5.6 HSICMU_ATRIM: HSICMU Control Auto-TRIM Register 336
19.5.7 HSICMU_CAP: HSICMU Capture Value Register 337
19.5.8 HSICMU_THR: HSICMU Frequency Error Rate Threshold Register 337
19.5.9 HSICMU Register Map Summary 338
20. USB Full Speed Device Interface 339
20.1 Introduction 339
20.2 Main Features 339
20.3 USB Implementation 339
20.4 USB Block Diagram 340
20.5 USB Functional Description 341
20.5.1 USB Pins and Signals 341
20.5.2 USB Description of USB Block 342
20.6 Programming Considerations 343
20.6.1 Select Clock Source 343
20.6.2 Endpoint Initialization 343
20.6.3 Endpoint Control 343
20.6.4 USB Interrupt Handling 344
20.6.5 Power Saving 345
20.6.6 Buffer Control 346
20.6.7 Handling Transactions with USB Device Peripheral 347
20.7 USB Registers 349
20.7.1 USB_FADDR: Function Address Register 351
20.7.2 USB_POWER: Power Management Register 352
20.7.3 USB_INTRIN1: Interrupt for EP0 + IN EP1 to EP5 Register 353
20.7.4 USB_INTROUT1: Interrupt for OUT EP1 to EP5 Register 354
20.7.5 USB_INTRUSB: Interrupt for Common USB 355
20.7.6 USB_INTRIN1E: Interrupt Enable for INTIN 356
20.7.7 USB_INTROUT1E: Interrupt Enable for INTOUT 357
20.7.8 USB_INTRUSBE: Interrupt Enable for INTUSB 358
20.7.9 USB_FRAME1: Frame Number bit 0 to 7 359
20.7.10 USB_FRAME2: Frame Number bit 8 to 10 359
20.7.11 USB_INDEX: Index for Selecting Endpoint Register 360
20.7.12 USB_PUPDCR: USB DP/DM Pins Pull-up/Down Resistor Control Register. 361
20.7.13 USB_PHYCR: USB Power Control Register. 362
20.7.14 USB_INMAXP: Maximum Package Size for IN Register 362
20.7.15 USB_CSR0: Control Status Register 0 for EP0 Register 363
20.7.16 USB_INCSR1: Control Status Register 1 for IN Register 364
20.7.17 USB_INCSR2: Control Status Register 2 for IN Register 365
20.7.18 USB_OUTMAXP: Maximum Package Size for OUT Register 366
20.7.19 USB_OUTCSR1: Control Status Register 1 for OUT Register 367
20.7.20 USB_OUTCSR2: Control Status Register 2 for OUT Register 368
20.7.21 USB_COUNT0: Number of Bytes for EP0 Register 368
20.7.22 USB_OUTCOUNT1: Number of Bytes for Lower Byte Register 369
20.7.23 USB_OUTCOUNT2: Number of Bytes for Upper Byte Register 369
20.7.24 USB_FIFOn: FIFO for EP0 to EP5 Register 370
20.7.25 USB_DMAINTR: USB DMA Interrupt Pending Register 370
20.7.26 USB_DMA1CNTL: USB DMA Channel 1 Control Register 371
20.7.27 USB_DMA1ADDR: USB DMA Channel 1 Memory Address Register 372
20.7.28 USB_DMA1COUNT: USB DMA Channel 1 Transfer Byte Count Register 372
20.7.29 USB_DMA2CNTL: USB DMA Channel 2 Control Register 373

20.7.30 USB_DMA2ADDR: USB DMA Channel 2 Memory Address Register 374
20.7.31 USB_DMA2COUNT: USB DMA Channel 2 Transfer Byte Count Register 374
20.7.32 USB Register Map Summary 375
Abbreviations for Registers 379
Glossary 380
Revision History 381

List of Tables

Table 1. A31G336 Memory Density 17
Table 2. A31G336 Features and Peripheral Counts 18
Table 3. Pin Description 30
Table 4. Interrupt Vector Map 38
Table 5. A31G336 Memory Map and Peripheral Register Boundary Address 41
Table 6. Boot Mode Pin List 42
Table 7. Clock Sources 49
Table 8. Functional Table on Current Mode 58
Table 9. Pins and External Signals for SCU 59
Table 10. Base Address of SCU (Chip Configuration) 60
Table 11. SCU (Chip Configuration) Register Map 60
Table 12. Base Address of SCU (Clock Generation) 61
Table 13. SCU Register Map (Clock Generation) 61
Table 14. Base Address of SCU (LVR/LVI) 61
Table 15. SCU Register Map (LVR/LVI) 61
Table 16. SCU CHIPCONFIG Register Map Summary 94
Table 17. SCU Register Map Summary 95
Table 18. SCU LVI/LVR Register Map Summary 96
Table 19. PCU & GPIO Internal Signal 101
Table 20. PCU and GPIO Pins 101
Table 21. GPIO Alternate Function 102
Table 22. Interrupt Sources and Corresponding Pins in GPIO Module 104
Table 23. Base Address of PCU 108
Table 24. PCU and GPIO Register Map 108
Table 25. PCU and GPIO Register Map Summary 118
Table 26. Base Address of Interrupt Register 119
Table 27. Interrupt Register Map 119
Table 28. Corresponding Interrupts of IMSKx 124
Table 29. Interrupt Register Map Summary 125
Table 30. Flash Memory Features 127
Table 31. Base Address of Flash Memory Controller 133
Table 32. FMC Register Map 133
Table 33. FMC Register Map Summary 139
Table 34. Manufacturer Area Map 141
Table 35. Manufacturer Area 0 Configuration Map Summary 145
Table 36. Configure Option Area Map 146
Table 37. Configure Option Page 0 Configuration Map Summary 154
Table 38. Configure Option Page 1 Configuration Map Summary 154
Table 39. Base Address of DMA Channel 0/1/2/3 156
Table 40. DMA Channel n Register Map (n = 0 and 3) 156
Table 41. DMA Register Map Summary 161
Table 42. Prescaled WDT Counter Clock Frequency 165
Table 43. Base Address of WDT 166
Table 44. WDT Register Map 166
Table 45. WDT Register Map Summary 171
Table 46. Base Address of WT 174
Table 47. WT Register Map 174

Table 48. Value of RTC_HOUR/RTC_ALHOUR by HS24 bit 181
Table 49. RTCC Register Map Summary 188
Table 50. Pins and External Signals for Timer Counter n (n = 10, 11, 12, 13, 14 and 15) 190
Table 51. Timer n Operating Modes (n = 10, 11, 12, 13, 14 and 15) 191
Table 52. Base Address of Timer 10/11/12/13/14/15 198
Table 53. TIMER Register Map 198
Table 54. TIMERn Register Map Summary 204
Table 55. Pins and External Signals for Timer Counter 2n (n = 0, 1) 206
Table 56. Timer 2n Operating Modes (n = 0, 1) 207
Table 57. Base Address of Timer 20/21 214
Table 58. TIMER Register Map 214
Table 59. TIMER20 Register Map Summary 220
Table 60. TIMER21 Register Map Summary 221
Table 61. Pins and External Signals for USART 10/11/12/13 224
Table 62. Equations for Calculating USART Baud Rate Register Settings (n = 10, 11, 12 and 13) 226
Table 63. CPOL Functionality (n = 10, 11, 12, and 13) 238
Table 64. Base Address of USART 10/11/12/13 242
Table 65. USARTn Register Map 242
Table 66. USART Register Map Summary 252
Table 67. Pins and External Signals for SPI (n = 0 and 1) 254
Table 68. CPOL Functionality (n = 0 and 1) 255
Table 69. Base Address of SPI 0/1 257
Table 70. SPIn Register Map 257
Table 71. SPIn Register Map Summary 261
Table 72. Pins and External Signals for I2Cn (n = 0, 1 and 2) 263
Table 73. Base Address of I2Cn Interface 275
Table 74. I2C Register Map 275
Table 75. I2C Register Map Summary 281
Table 76. ADC Frequency Set according to AVDD 286
Table 77. ADC Result Data 290
Table 78. Base Address of ADC 292
Table 79. ADC Register Map 292
Table 80. ADC Register Map Summary 299
Table 81. Base Address of CRC and Checksum 304
Table 82. CRC and Checksum Register Map 304
Table 83. CRC Register Map Summary 310
Table 84. Pins and External Signals for Comparator 0/1 313
Table 85. Base Address of Comparator 0/1 313
Table 86. Comparator n Register Map (n = 0 and 1) 313
Table 87. CMP Register Map Summary 317
Table 88. TS Output Frequency Registers in Configure Option Page 0 320
Table 89. Base Address of TS 321
Table 90. TS Register Map 321
Table 91. TS Register Map Summary 325
Table 92. Base Address of HSICMU 331
Table 93. HSICMU Register Map 331
Table 94. HSICMU Register Map Summary 338
Table 95. A31G336 USB Implementation 339
Table 96. Pins Assignment of USB: External Pins 341
Table 97. Base Address of USB 349

Table 98. USB Register Map 349
Table 99. USB Register Map Summary 375
Table 100. Abbreviations for Registers 379

List of Figures

Figure 1. A31G336 Block Diagram 1
Figure 2. 64-LQFP-1010 Pinouts 24
Figure 3. 64-LQFP-1212 Pinouts 25
Figure 4. 48-LQFP Pinouts 26
Figure 5. 48-QFN Pinouts 27
Figure 6. 40-QFN Pinouts 28
Figure 7. 32-QFN Pinouts 29
Figure 8. System Block Diagram 37
Figure 9. Memory Map 40
Figure 10. PLL Block Diagram 45
Figure 11. SCU Block Diagram 47
Figure 12. Clock Source Configuration 48
Figure 13. Miscellaneous Clock Configuration 50
Figure 14. Clock Configuration Procedure 52
Figure 15. Power-Up POR Sequence 54
Figure 16. Reset Configuration 55
Figure 17. LVR Reset Timing Diagram 56
Figure 18. Operation Mode Transition 57
Figure 19. Reset Circuit Diagram 66
Figure 20. Wake-Up Timer Circuit Diagram 70
Figure 21. Clock Monitoring Circuit Diagram 79
Figure 22. VUSB33 Pin Circuit Diagram 90
Figure 23. LVI Block Diagram 92
Figure 24. LVR Block Diagram 93
Figure 25. PCU/GPIO Block Diagram 100
Figure 26. I/O Port Block Diagram (GPIO Pins) 100
Figure 27. Output Port Block Diagram 105
Figure 28. Input Port Block Diagram 106
Figure 29. Example Timing Diagram of Port Debouncing 107
Figure 30. Configuration Map for External Interrupt Group 0/1/2/3 123
Figure 31. Flash Memory Map (256 KB Code Flash) 128
Figure 32. Configure Option Area Structure 140
Figure 33. DMA Controller Block Diagram 155
Figure 34. WDT Block Diagram 163
Figure 35. WDT Interrupt and WDT Reset Timing Diagram 164
Figure 36. RTCC Block Diagram 173
Figure 37. Timer Counter n Timer Block Diagram (n = 10, 11, 12, 13, 14 and 15) 190
Figure 38. 16-Bit Timer/Counter Mode for Timer n (n = 10, 11, 12, 13, 14 and 15) 192
Figure 39. 16-Bit Timer/Counter n Example (n = 10, 11, 12, 13, 14 and 15) 193
Figure 40. 16-Bit Capture Mode for Timer n (n = 10, 11, 12, 13, 14 and 15) 194
Figure 41. 16-Bit Capture Mode for Timer n (n = 10, 11, 12, 13, 14 and 15) 195
Figure 42. Express Timer Overflow in Capture Mode (n = 10, 11, 12, 13, 14 and 15) 195
Figure 43. 16-Bit PPG Repeat and One-Shot Mode for Timer n (n = 10, 11, 12, 13, 14 and 15) 196
Figure 44. 16-Bit PPG Mode Timing Chart for Timer n (n = 10, 11, 12, 13, 14 and 15) 197
Figure 45. Timer Counter 2n Timer Block Diagram (n = 0, 1) 206
Figure 46. 32-Bit Timer/Counter Mode for Timer 2n (n = 0, 1) 208
Figure 47. 32-Bit Timer/Counter 2n Example (n = 0, 1) 209

Figure 48. 32-Bit Capture Mode for Timer 2n (n = 0, 1) 210
Figure 49. 32-Bit Capture Mode for Timer 2n (n = 0, 1) 211
Figure 50. Express Timer Overflow in Capture Mode 211
Figure 51. 32-Bit PPG Repeat and One-Shot Mode for Timer 2n (n = 0, 1) 212
Figure 52. 32-Bit PPG Mode Timing Chart for Timer 2n (N = 0, 1) 213
Figure 53. UART and LIN Block Diagram of USART (n = 10, 11, 12, and 13) 223
Figure 54. SPIn Block Diagram of USART (n = 10, 11, 12, and 13) 224
Figure 55. Clock Generation Block Diagram (USART, n = 10, 11, 12, and 13) 225
Figure 56. Synchronous Mode SCKn Timing (USART, n = 10, 11, 12, and 13) 227
Figure 57. Frame Format (UART) 228
Figure 58. Asynchronous Start Bit Sampling (n = 10, 11, 12, and 13) 233
Figure 59. Asynchronous Sampling of Data and Parity Bit (n = 10, 11, 12, and 13) 234
Figure 60. Stop Bit Sampling and Next Start Bit Sampling (n = 10, 11, 12, and 13) 235
Figure 61. Receive Timeout Function (n = 10, 11, 12, and 13) 235
Figure 62. Auto Baud Rate Detection Timing Diagram (n = 10, 11, 12, and 13) 236
Figure 63. USART SPIn Clock Formats when CPHAn=0 (n = 10, 11, 12, and 13) 238
Figure 64. USART SPIn Clock Formats when CPHAn=1 (n = 10, 11, 12, and 13) 239
Figure 65. LIN Break Field Detection and Transmit Timing Diagram (n = 10, 11, 12, and 13) 241
Figure 66. SPIn Block Diagram (n = 0 and 1) 254
Figure 67. SPIn Clock Formats When CPHAn=0 (n = 0 and 1) 255
Figure 68. SPIn Clock Formats When CPHAn=1 (n = 0 and 1) 256
Figure 69. I2Cn Block Diagram (n = 0, 1 and 2) 263
Figure 70. I2C Bus Bit Transfer (n = 0, 1 and 2) 264
Figure 71. START and STOP Conditions (n = 0, 1 and 2) 264
Figure 72. I2C Bus Data Transfer (n = 0, 1 and 2) 265
Figure 73. I2C Bus Acknowledge (n = 0, 1 and 2) 266
Figure 74. Clock Synchronization during Arbitration Procedure (n = 0, 1 and 2) 267
Figure 75. Arbitration Procedure Between Two Masters (n = 0, 1 and 2) 268
Figure 76. 12-Bit ADC Block Diagram 283
Figure 77. 12-Bit ADC Enable/Disable Timing Chart 285
Figure 78. High-Speed ADC Conversion Timing Chart 286
Figure 79. Example of Single Conversion Mode 287
Figure 80. Example of Sequential Conversion Mode 288
Figure 81. Example of Continuous Conversion Mode 289
Figure 82. Recommend Circuit for ADC Input 291
Figure 83. CRC and Checksum Timer Block Diagram 301
Figure 84. CRC Polynomial Structure 302
Figure 85. Comparator 0/1 Block Diagram 312
Figure 86. TS Block Diagram 318
Figure 87. Temperature Sensor Timing Diagram 319
Figure 88. HSICMU Block Diagram 327
Figure 89. HSICMU Counter Behavior 328
Figure 90. USB Block Diagram 340
Figure 91. USB Interrupt Service Routine 344
Figure 92. USB Buffer Control 346
Figure 93. Setup Transaction Followed by Data IN Transaction 347
Figure 94. Data Out Transfer 348

1. Description

A31G336 is a 32-bit high-performance microcontroller with up to 256 KB of Flash memory with ECC. It is a powerful microcontroller which provides effective solutions to various electrical appliances which require both low power consumption and high performance (Arm Cortex-M0+ core).

1.1 Product Category Definition

Table 1 gives an overview of memory density versus product line. The present document describes the superset of features for each product category.

See Table 2 for the list of features per category.

Table 1. A31G336 Memory Density

Memory Density		Category
Flash	RAM	
256 KB	32 KB	A31G336

1.2 Availability of Peripherals

Table 2 summarizes and lists product specific features available in the A31G336 considering the largest package.

For the availability of peripherals and their numbers across all sale types, refer to the particular device's Datasheet: **Product selection table** or **Ordering information**.

Table 2. A31G336 Features and Peripheral Counts

Peripherals		Description
Core	CPU	• Max. operating frequency: 48 MHz • 32-bit ARM Cortex-M0+ CPU • Register settings in CPU: - General purpose register: 32-bit Thumb-2 instruction set - Main Stack Pointer (MSP) and Process Stack Pointer (PSP): R13 - Link Register (LR): R14 - Program Counter (PC): R15 • Data ordering format: Little-Endian • Von Neumann Architecture • AHB / APB
	Interrupt Controller	• NVIC (Nested-Vectored Interrupt Controller) • Up to 32 peripheral interrupts supported. • Assignable with four different priority levels
Memory	Code Flash	• Capacity: 256 KB • A high-capacity Code Flash memory built in • Max. 48 MHz Flash access speed - 0-wait: up to 20 MHz - 1-wait: up to 40 MHz - 2-wait: up to 48 MHz • Erase unit: Page (512-byte), Sector (2048-byte), Bulk (256 KB, full-chip) • Program unit: Word (4-byte) • Read protection • Self-programming • CRC code generation and verification for the Flash memory • Endurance: 10,000 cycles • Lifetime: 10 years • ECC (1-bit correction and 2-bits detection)
	Boot ROM	• Initiating the microcontroller's boot mode by receiving an external input signal at the BOOT pin, • UART boot mode • In-system programming - A user can store information in the internal Flash memory by using an application board to write data to the memory.

Table 2. A31G336 Features and Peripheral Counts (continued)

Peripherals		Description
	SRAM	• Capacity: 32 KB • Usable as a program's work area • Part of the SRAM can be remapped into an interrupt vector area
System Control Unit (SCU)	Operating frequency	• Up to 48 MHz
	Clock	• High-speed Internal oscillator (HSI) - 32 MHz (±1.5% @ Ta = 0°C to +50°C) - 32 MHz (±3% @ Ta = -40°C to +85°C) • Low-speed Internal oscillator (LSI) - 40 kHz (±15% @ Ta = -40°C to +85°C) • External Main Oscillator (HSE): 2 MHz to 16 MHz • External Sub-Oscillator (LSE): 32.768 kHz • Phase-locked loop (PLL): 48 MHz
	Clock monitoring	• System fail-safe function by clock monitoring - High frequency internal RC oscillator (HSI) - External main oscillator (HSE) - External sub oscillator (LSE) - Main system clock (MCLK)
	Operating mode	• RUN mode • SLEEP mode • DEEP-SLEEP (STOP) mode
	Reset	• nRESET pin reset • Core reset • Software reset • POR (Power-On Reset) • LVR (Low-Voltage Reset) • WDTR (Watch Dog Timer Reset) • MON (Clock monitoring reset)
	VDC	• Low-dropout (LDO) regulator built in for low-voltage operation
	POR	• The POR generator detects an internal 1.4 V and generates a reset signal
	LVI	• 13 low-voltage detection levels • Supports interrupts • Supports wake-up from SLEEP mode and DEEP SLEEP modes
	Wake-up	• Wake-up by an external interrupt (GPIO) pin • Wake-up by a Real Time Clock/Calendar (RTCC) • Wake-up by a Watchdog Timer (WDT) • Wake-up by a Low-Voltage Indicator (LVI) • Wake-up by a I2C • Wake-up by a USART
		• Wake-up by a T20

Table 2. A31G336 Features and Peripheral Counts (continued)

Peripherals		Description
		• Wake-up by a comparator • Wake-up by a USB
	Wake-Up Timer (WUT)	• 16-bit down-count timer • Underflow interrupt • Stabilization time upon wake-up from deep sleep mode
General-Purpose I/O (GPIO)		• Input / Output (I/O) port for general purpose • 64-LQFP-1010 - I/O pins: 53 • 64-LQFP-1212 - I/O pins: 53 • 48-LQFP-0707 - I/O pins: 37 • 48-QFN-0505 - I/O pins: 37 • 40-QFN-0505 - I/O pins: 29 • 32-QFN-0505 - I/O pins: 22 • Each pin can be set for one of the following modes: - Push-pull output - Open drain output - Input • The use of each pin can be set by setting the register • Corresponding pin can be configured as an external interrupt source, either the level-trigger / edge-trigger interrupt or the rising-edge / falling-edge / both-edge interrupt • Pull-up or pull-down resistor, and debouncing can be set for each pin • Each pin bit can be individually set / reset • Wake-up events triggered by external asynchronous inputs
Direct Memory Access (DMA) Controller		• 4-channel Direct Memory Access (DMA) supporting peripherals • 8-bit / 16-bit / 32-bit data transfers • Compatible with 19 different types of peripherals • SPI0, SPI1, USART10, USART11, USART12, USART13, I2C0, I2C1, I2C2, • ADC
General Purpose Timer	16-bit Timer	• General-purpose 16-bit up-count timer • 6 channels - Timer 1n capture port (T1nCAP) input channels - Timer 1n output port (T1nOUT) output channels • Timer operating modes - Timer/counter mode

Table 2. A31G336 Features and Peripheral Counts (continued)

Peripherals		Description
		- Capture mode - PPG one-shot mode - PPG repeat mode • Interrupt events - Timer/counter match interrupt - Timer capture interrupt • Input clock selection - PCLK clock divided by prescaler is selectable - External clock is selectable • Timer signals can be generated through T1nOUT pins • 12-bit prescaler
	32-bit Timer	• General-purpose 32-bit up-count timer • Two channels - Timer 2n capture port (T2nCAP) input channels - Timer 2n output port (T2nOUT) output channels • Timer operating modes - Timer/counter mode - Capture mode - PPG one-shot mode - PPG repeat mode • Interrupt events - Timer/counter match interrupt - Timer capture interrupt • Input clock selection - PCLK clock divided by prescaler is selectable - External clock is selectable (T20, T21) - LSE clock is selectable (T20) • Timer signals can be generated through T2nOUT pins • 12-bit prescaler
System Timer	WDT	• 24-bit down-count timer • Reset generation • Window match interrupt • Underflow interrupt • LSI or PCLK is selectable
	RTCC	• Calendar with 0.5 seconds, seconds, minutes, hours, day, week, month, and year up to 2099. • Time error correction function • Alarm function with interrupt • Wake-up possible from DEEP SLEEP mode
Communication Interface	USART	• Four channels for asynchronous, synchronous, and serial peripheral interfaces • Asynchronous/synchronous modes (UART) • 5- to 9-bit data transfers • Even / Odd / Non-parity generation and checking • 1-bit or 2-bit stop bit generation and checking • 12-bit baud rate generator • Receive time out function from start bit (Asynchronous only)

Table 2. A31G336 Features and Peripheral Counts (continued)

Peripherals		Description
		• Receive character detection • Auto baud rate detection • 1-wire half-duplex communication • Local interconnection network (LIN) • Tx break transmit and Rx break detection • Serial peripheral interface (SPI) • Master / slave operation • Loop-back function • 8-bit data transmit / receive • Both least significant bit (LSB)-first and most significant bit (MSB)-first modes available
	SPI	• Two synchronous serial communication port channels • Master / slave operation • Loop-back mode • Programmable and flexible communication - 8-bit data transmit / receive - SPI clock speed - Both least significant bit (LSB)-first and most significant bit (MSB)-first modes available
	I2C	• Standard I2C communication protocol • Three channels supported • Master and slave modes supported for each channel • 7-bit addressing supported for slave mode • SCL signal's high/low periods and SDA signal's hold time settable
	USB	• USB 2.0 Full-Speed device • Crystal-less with clock management unit (HSICMU) • 1024 bytes packet buffer memory SRAM • Number of endpoints from 1 to 5 in addition to Endpoint 0 • IN and OUT Endpoint • USB DMA 2 channels, 64 bytes buffer size.
High-Speed Internal OSC Clock Management Unit (HSICMU)		• Selectable synchronization signal source - USB SOF (Start of Frame) sent received toggle - LSE • External pin (PD5)
12-bit A/D Converter	ADC	• 16 analog input channels • Software and timer match signal triggers supported • Conversion modes and over sampling supported
Comparator	CMP	• External analog inputs • Hysteresis function • Low and fast speed selectable • Wake-up possible from DEEP SLEEP mode
Cyclic Redundancy Check	CRC	• CRC operating modes: - CRC-CCITT (0x1021) - CRC-16 (0x8005) • Auto and user mode supported

Table 2. A31G336 Features and Peripheral Counts (continued)

Peripherals		Description
Temperature Sensor	TS	• -40°C to +85°C wide range of operating temperature • A down counter at 20-bit intervals to count the frequency of the TS • A 24-bit data register to store the count value of the temperature sensor frequency
Operating Voltage		1.8 V to 5.5 V
Operating temperature		Commercial grade (-40°C to +85°C)
Package		• Six types of package options - 64-LQFP-1010 (0.5 mm pitch) - 64-LQFP-1212 (0.65 mm pitch) - 48-LQFP-0707 (0.5 mm pitch) - 48-QFN-0505 (0.35 mm pitch) - 40-QFN-0505 (0.40 mm pitch) - 32-QFN-0505 (0.50 mm pitch)

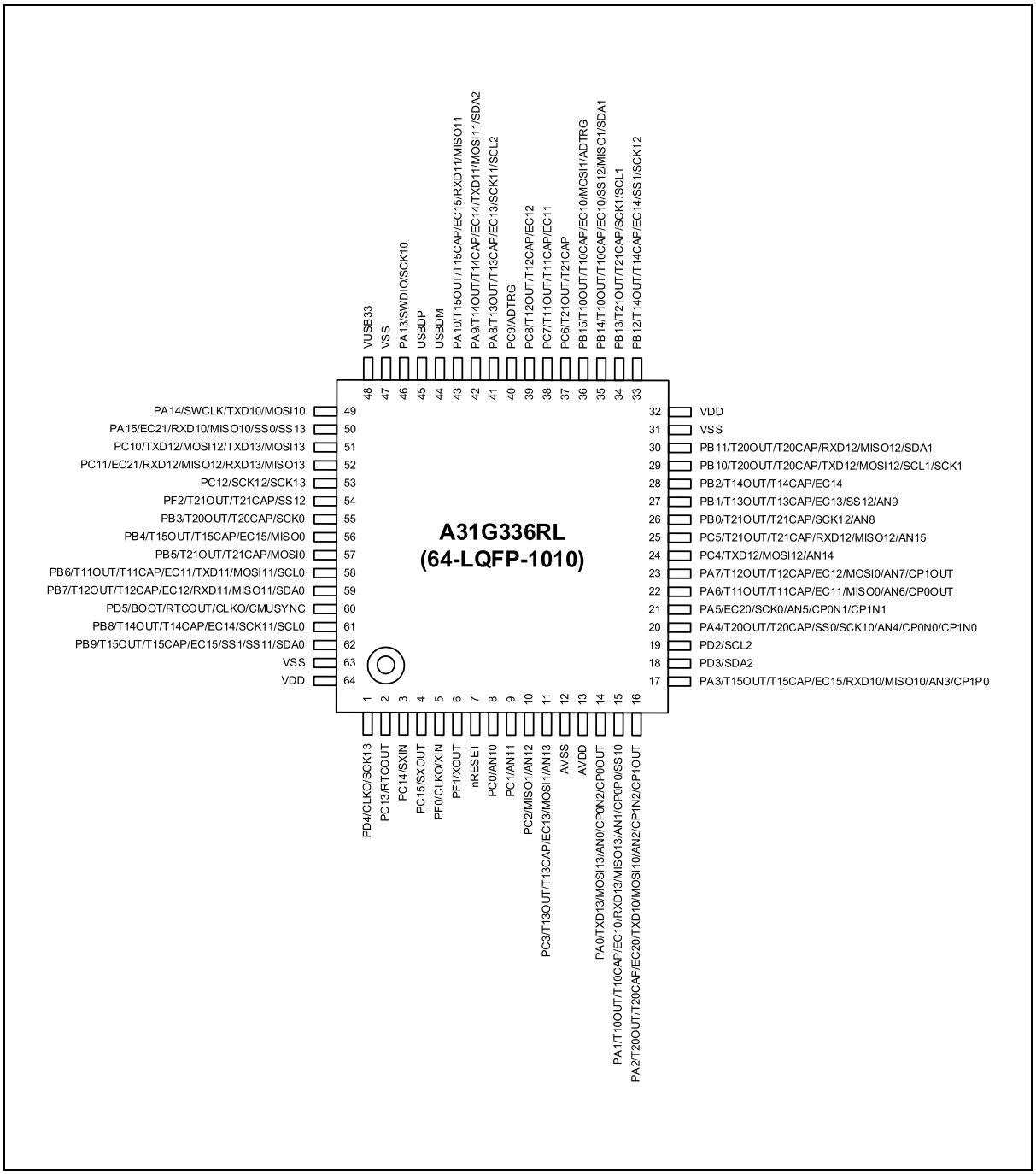
2. Pinouts and Pin Descriptions

In this chapter, pinouts and pin descriptions of the A31G336 are described.

2.1 Pinouts

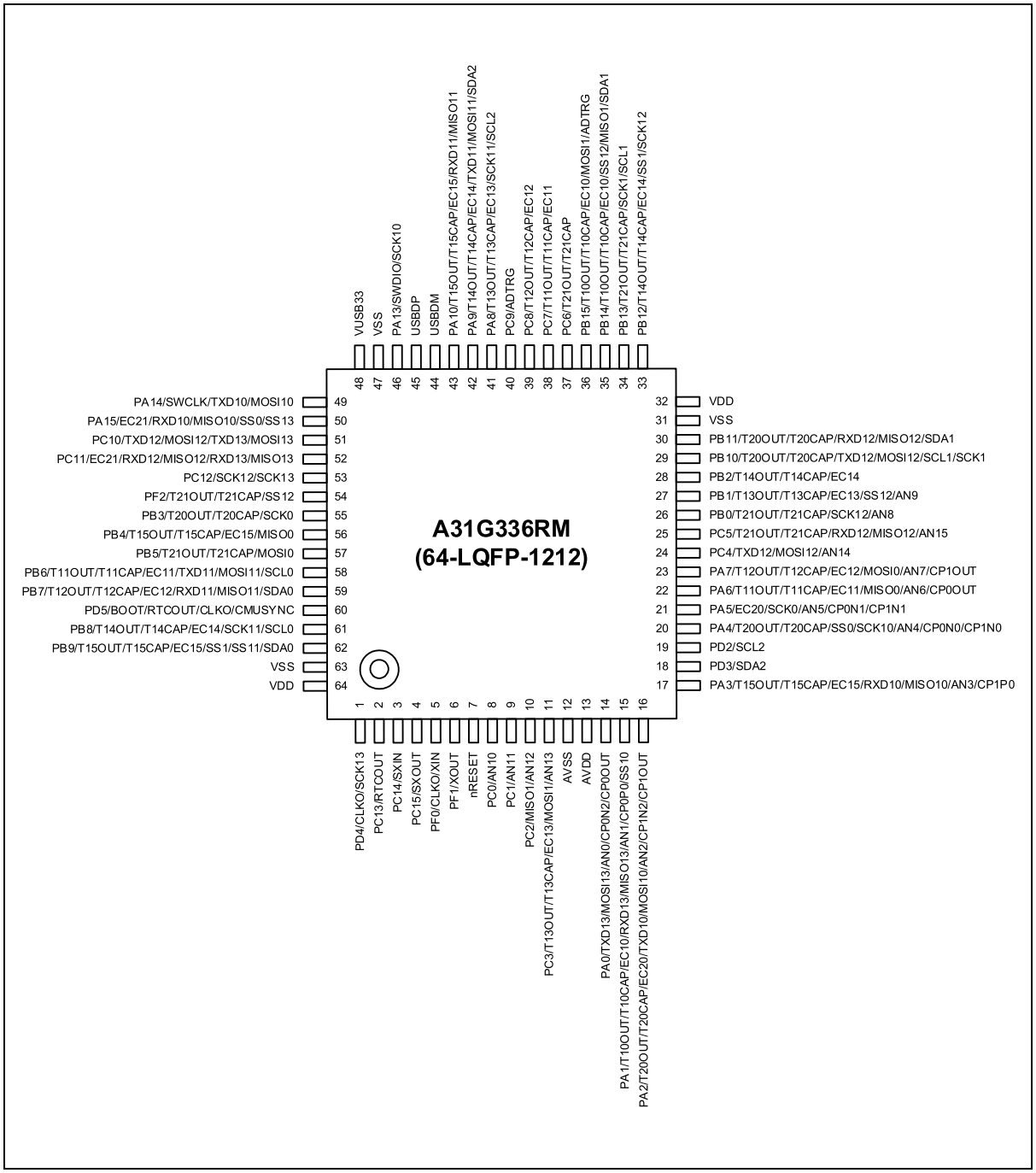
2.1.1 A31G336RL (64-LQFP-1010)

Figure 2. 64-LQFP-1010 Pinouts



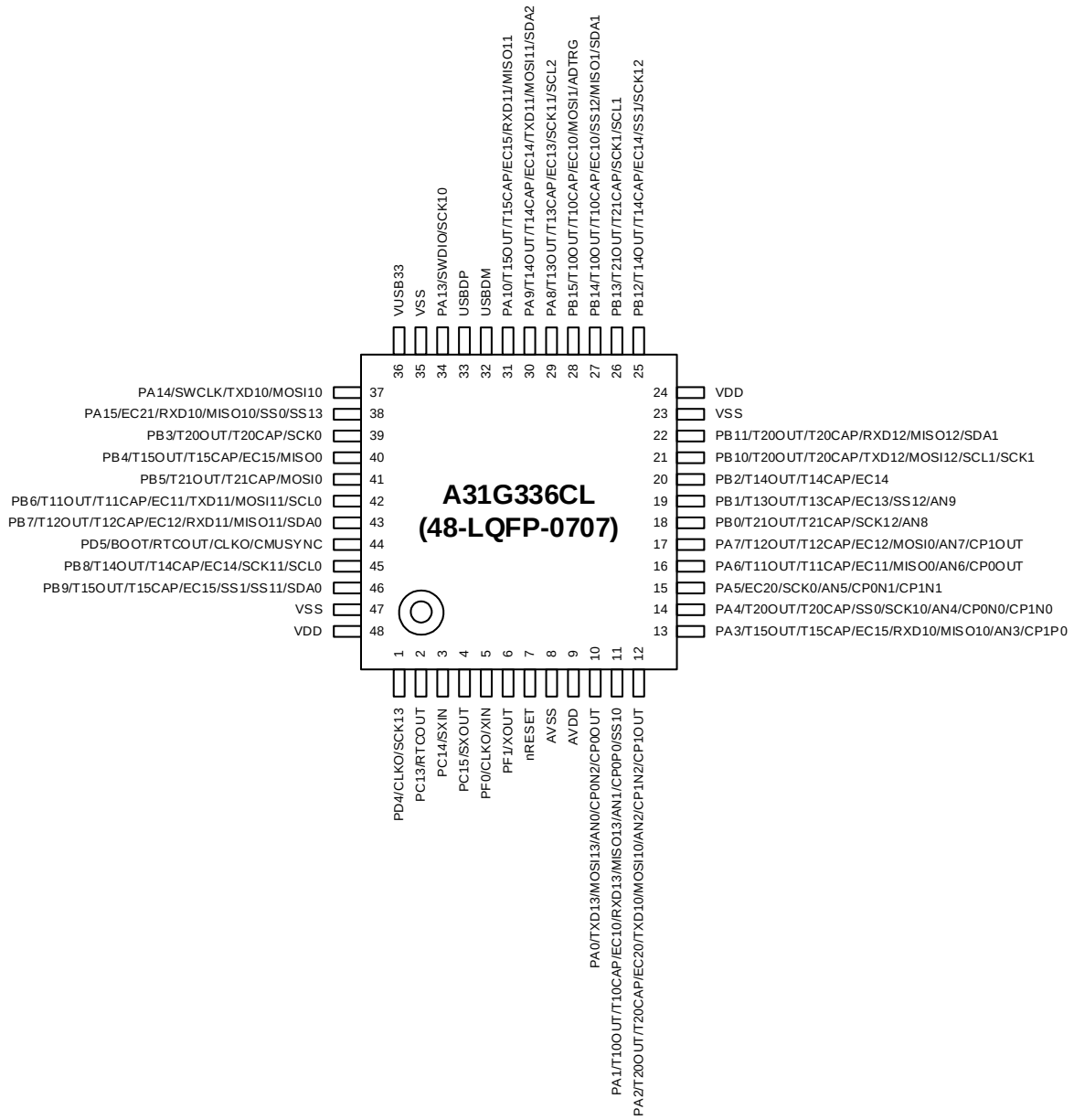
2.1.2 A31G336RM (64-LQFP-1212)

Figure 3. 64-LQFP-1212 Pinouts



2.1.3 A31G336CL (48-LQFP)

Figure 4. 48-LQFP Pinouts

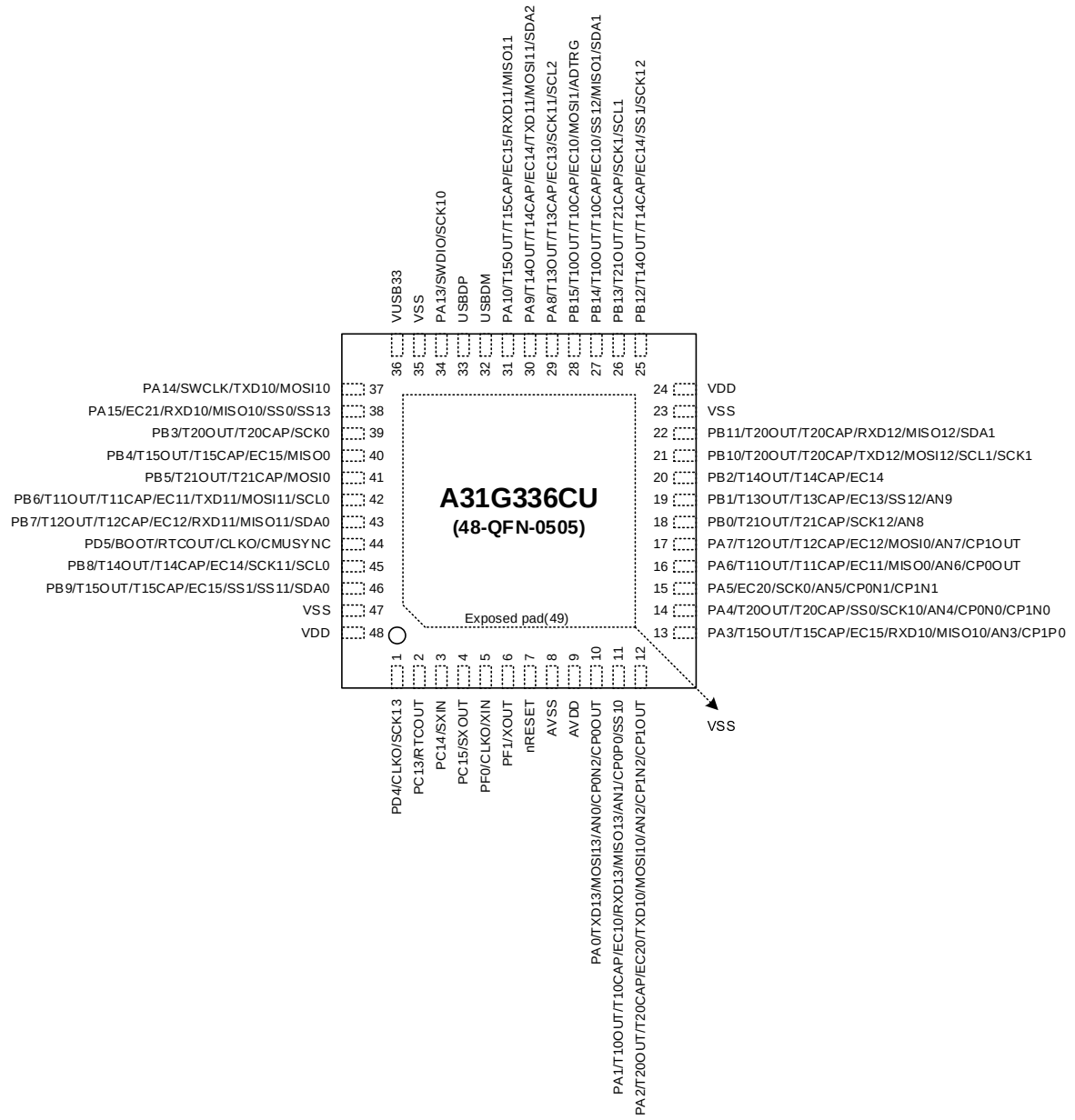


NOTE:

1. Each of the PC[12:0], PD[3:2], and PF[2] pins should be selected as a push-pull output or an input with pull-up or pull-down resistor by software control when the 48-LQFP package is used.

2.1.4 A31G336CU (48-QFN)

Figure 5. 48-QFN Pinouts

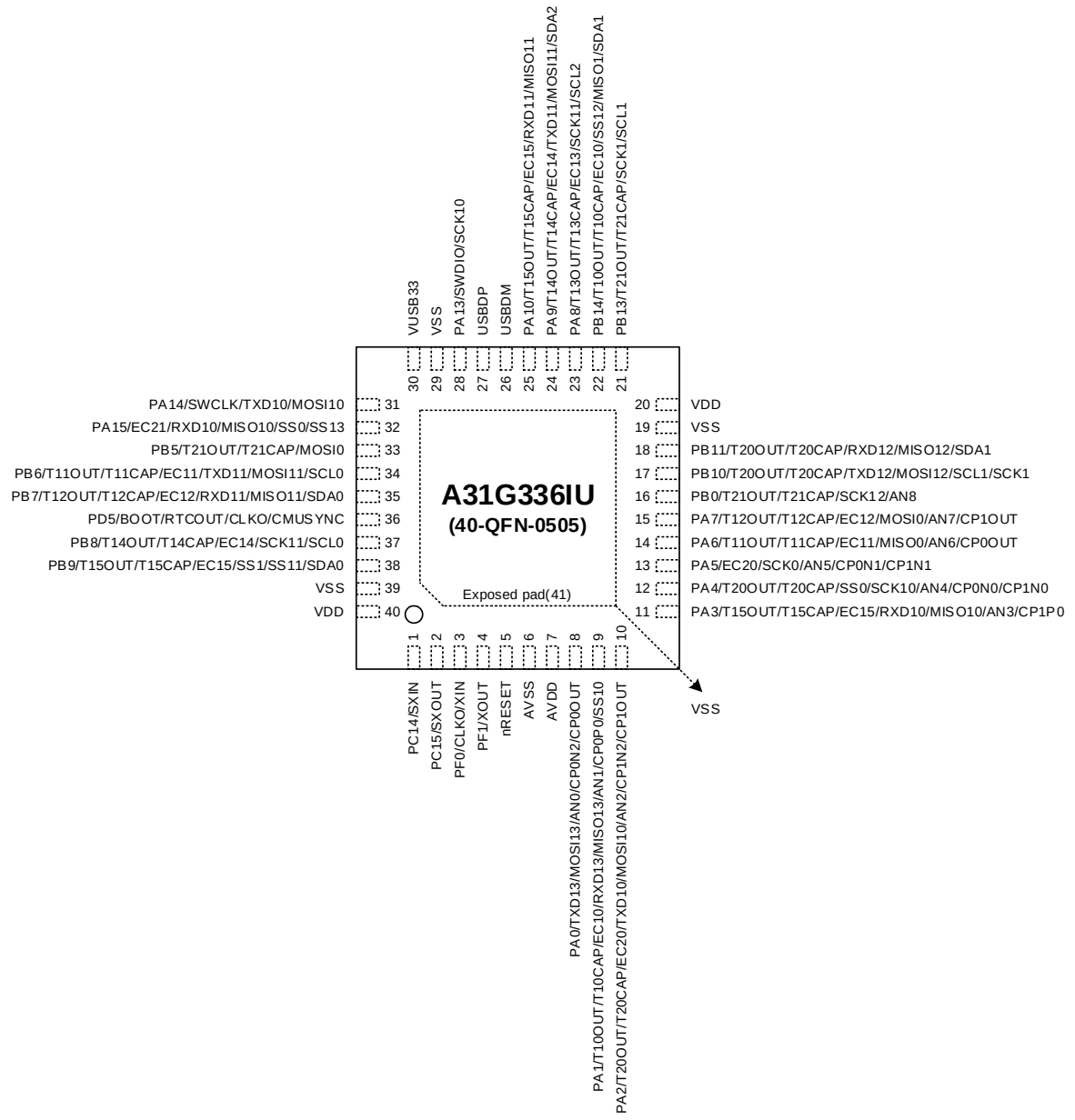


NOTE:

1. Each of the PC[12:0], PD[3:2], and PF[2] pins should be selected as a push-pull output or an input with pull-up or pull-down resistor by software control when the 48-QFN package is used.

2.1.5 A31G336IU (40-QFN)

Figure 6. 40-QFN Pinouts

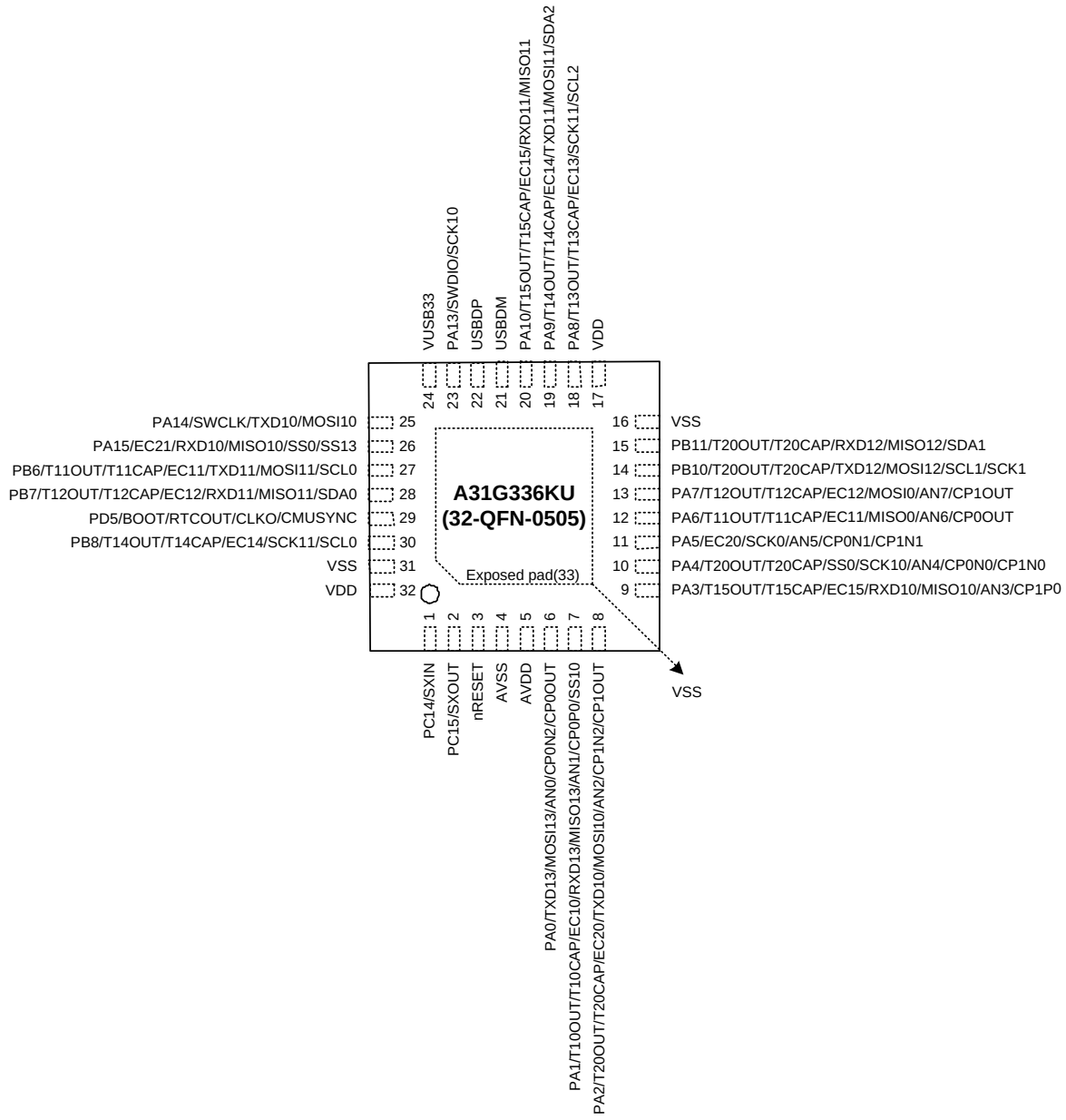


NOTE:

1. The PB[15, 12, 4:1], PC[13:0], PD[4:2], and PF[2] pins should be selected as a push-pull output or an input pull-up or pull-down resistor by software control when the 40-QFN package is used.

2.1.6 A31G336KU (32-QFN)

Figure 7. 32-QFN Pinouts



NOTE:

1. Each of the PB[15:12, 9, 5:0], PC[13:0], PD[4:2], and PF[2:0] pins should be selected as a push-pull output or an input with pull-up or pull-down resistor by software control when the 32-QFN package is used.

2.2 Pin Description

Pin configuration information in contains one pair of power/ground and other dedicated pins. These multi-function pins have up to eight selections of functions including GPIO. Configuration including pin ordering can be changed without notice.

Table 3. Pin Description

Pin No.				Pin Name	Type(1)	Description	Remark
64-Pin	48-Pin	40-Pin	32-Pin				
1	1	-	-	PD4*	IOUDS	Port D bit 4 input / output	
				CLKO	O	System clock output	
				SCK13	IO	SPI clock input / output	
2	2	-	-	PC13*	IOUDS	Port C bit 13 input / output	
				RTCOUT	O	Real time clock output	
3	3	1	1	PC14*	IOUDS	Port C bit 14 input / output	
				SXIN	IA	Sub-oscillator input	
4	4	2	2	PC15*	IOUDS	Port C bit 15 input / output	
				SXOUT	OA	Sub-oscillator output	
5	5	3	-	PF0*	IOUDS	Port F bit 0 input / output	
				XIN	IA	Main oscillator input	
				CLKO	O	System clock output	
6	6	4	-	PF1*	IOUDS	Port F bit 1 input / output	
				XOUT	OA	Main oscillator output	
7	7	5	3	nRESET	IU	External reset input	Always pull-up
8	-	-	-	PC0*	IOUDS	Port C bit 0 input / output	
				AN10	IA	A/D converter analog input channel	
9	-	-	-	PC1*	IOUDS	Port C bit 1 input / output	
				AN11	IA	A/D converter analog input channel	
10	-	-	-	PC2*	IOUDS	Port C bit 2 input / output	
				MISO1	IO	SPI master in slave out	
				AN12	IA	A/D converter analog input channel	
11	-	-	-	PC3*	IOUDS	Port C bit 3 input / output	
				T13OUT	O	Timer 13 pulse output	
				T13CAP	I	Timer 13 capture input	
				EC13	I	Timer 13 event count input	
				MOSI1	IO	SPI master out slave in	
				AN13	IA	A/D converter analog input channel	
12	8	6	4	AVSS	AP	Analog ground	
13	9	7	5	AVDD	AP	AVDD	
14	10	8	6	PA0*	IOUDS	Port A bit 0 input / output	
				TXD13	O	UART data output	
				MOSI13	IO	SPI master out slave in	
				AN0	IA	A/D converter analog input channel	
				CP0N2	IA	Comparator negative input	
				CP0OUT	O	Comparator output	

Table 3. Pin Description (continued)

Pin No.				Pin Name	Type[1]	Description	Remark
64-Pin	48-Pin	40-Pin	32-Pin				
15	11	9	7	PA1*	IOUDS	Port A bit 1 input / output	
				T10OUT	O	Timer 10 pulse output	
				T10CAP	I	Timer 10 capture input	
				EC10	I	Timer 10 event count input	
				RXD13	I	UART data input	
				MISO13	IO	SPI master in slave out	
				AN1	IA	A/D converter analog input channel	
				CP0P0	IA	Comparator positive input	
				SS10	I	SPI slave select input	
16	12	10	8	PA2*	IOUDS	Port A bit 2 input / output	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				EC20	I	Timer 20 event count input	
				TXD10	O	UART data output	
				MOSI10	IO	SPI master out slave in	
				AN2	IA	A/D converter analog input channel	
				CP1N2	IA	Comparator negative input	
				CP1OUT	O	Comparator output	
17	13	11	9	PA3*	IOUDS	Port A bit 3 input / output	
				T15OUT	O	Timer 15 pulse output	
				T15CAP	I	Timer 15 capture input	
				EC15	I	Timer 15 event count input	
				RXD10	I	UART data input	
				MISO10	IO	SPI master in slave out	
				AN3	IA	A/D converter analog input channel	
				CP1P0	IA	Comparator positive input	
18	-	-	-	PD3*	IOUDS	Port D bit 3 input / output	
				SDA2	IO	I2C data input / output	
19	-	-	-	PD2*	IOUDS	Port D bit 2 input / output	
				SCL2	IO	I2C clock input / output	
20	14	12	10	PA4*	IOUDS	Port A bit 4 input / output	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				SS0	I	SPI slave select input	
				SCK10	IO	SPI clock input / output	
				AN4	IA	A/D converter analog input channel	
				CP0N0	IA	Comparator negative input	
				CP1N0	IA	Comparator negative input	
21	15	13	11	PA5*	IOUDS	Port A bit 5 input / output	
				EC20	I	Timer 20 event count input	
				SCK0	IO	SPI clock input / output	
				AN5	IA	A/D converter analog input channel	
				CP0N1	IA	Comparator negative input	
				CP1N1	IA	Comparator negative input	

Table 3. Pin Description (continued)

Pin No.				Pin Name	Type(1)	Description	Remark
64-Pin	48-Pin	40-Pin	32-Pin				
22	16	14	12	PA6*	IOUDS	Port A bit 6 input / output	
				T11OUT	O	Timer 11 pulse output	
				T11CAP	I	Timer 11 capture input	
				EC11	I	Timer 11 event count input	
				MISO0	IO	SPI master in slave out	
				AN6	IA	A/D converter analog input channel	
				CP0OUT	O	Comparator output	
23	17	15	13	PA7*	IOUDS	Port A bit 7 input / output	
				T12OUT	O	Timer 12 pulse output	
				T12CAP	I	Timer 12 capture input	
				EC12	I	Timer 12 event count input	
				MOSI0	IO	SPI master out slave in	
				AN7	IA	A/D converter analog input channel	
				CP1OUT	O	Comparator output	
24	-	-	-	PC4*	IOUDS	Port C bit 4 input / output	
				TXD12	O	UART data output	
				MOSI12	IO	SPI master out slave in	
				AN14	IA	A/D converter analog input channel	
25	-	-	-	PC5*	IOUDS	Port C bit 5 input / output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				RXD12	I	UART data input	
				MISO12	IO	SPI master in slave out	
				AN15	IA	A/D converter analog input channel	
26	18	16	-	PB0*	IOUDS	Port B bit 0 input / output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				SCK12	IO	SPI clock input / output	
				AN8	IA	A/D converter analog input channel	
27	19	-	-	PB1*	IOUDS	Port B bit 1 input / output	
				T13OUT	O	Timer 13 pulse output	
				T13CAP	I	Timer 13 capture input	
				EC13	I	Timer 13 event count input	
				SS12	I	SPI slave select input	
				AN9	IA	A/D converter analog input channel	
28	20	-	-	PB2*	IOUDS	Port B bit 2 input / output	
				T14OUT	O	Timer 14 pulse output	
				T14CAP	I	Timer 14 capture input	
				EC14	I	Timer 14 event count input	

Table 3. Pin Description (continued)

Pin No.				Pin Name	Type(1)	Description	Remark
64-Pin	48-Pin	40-Pin	32-Pin				
29	21	17	14	PB10*	IOUDS	Port B bit 10 input / output	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				TXD12	O	UART data output	
				MOSI12	IO	SPI master out slave in	
				SCL1	IO	I2C clock input / output	
				SCK1	IO	SPI clock input / output	
30	22	18	15	PB11*	IOUDS	Port B bit 11 input / output	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				RXD12	I	UART data input	
				MISO12	IO	SPI master in slave out	
				SDA1	IO	I2C data input / output	
31	23	19	16	VSS	P	Ground	
32	24	20	17	VDD	P	VDD	
33	25	-	-	PB12*	IOUDS	Port B bit 12 input / output	
				T14OUT	O	Timer 14 pulse output	
				T14CAP	I	Timer 14 capture input	
				EC14	I	Timer 14 event count input	
				SS1	I	SPI slave select input	
				SCK12	IO	SPI clock input / output	
34	26	21	-	PB13*	IOUDS	Port B bit 13 input / output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				SCK1	IO	SPI clock input / output	
				SCL1	IO	I2C clock input / output	
35	27	22	-	PB14*	IOUDS	Port B bit 14 input / output	
				T10OUT	O	Timer 10 pulse output	
				T10CAP	I	Timer 10 capture input	
				EC10	I	Timer 10 event count input	
				SS12	I	SPI slave select input	
				MISO1	IO	SPI master in slave out	
				SDA1	IO	I2C data input / output	
36	28	-	-	PB15*	IOUDS	Port B bit 15 input / output	
				T10OUT	O	Timer 10 pulse output	
				T10CAP	I	Timer 10 capture input	
				EC10	I	Timer 10 event count input	
				MOSI1	IO	SPI master out slave in	
				ADTRG	I	ADC trigger input	
37	-	-	-	PC6*	IOUDS	Port C bit 6 input / output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	

Table 3. Pin Description (continued)

Pin No.				Pin Name	Type(1)	Description	Remark
64-Pin	48-Pin	40-Pin	32-Pin				
38	-	-	-	PC7*	IOUDS	Port C bit 7 input / output	
				T11OUT	O	Timer 11 pulse output	
				T11CAP	I	Timer 11 capture input	
				EC11	I	Timer 11 event count input	
39	-	-	-	PC8*	IOUDS	Port C bit 8 input / output	
				T12OUT	O	Timer 12 pulse output	
				T12CAP	I	Timer 12 capture input	
				EC12	I	Timer 12 event count input	
40	-	-	-	PC9*	IOUDS	Port C bit 9 input / output	
				ADTRG	I	ADC trigger input	
41	29	23	18	PA8*	IOUDS	Port A bit 8 input / output	
				T13OUT	O	Timer 13 pulse output	
				T13CAP	I	Timer 13 capture input	
				EC13	I	Timer 13 event count input	
				SCK11	IO	SPI clock input / output	
				SCL2	IO	I2C clock input / output	
42	30	24	19	PA9*	IOUDS	Port A bit 9 input / output	
				T14OUT	O	Timer 14 pulse output	
				T14CAP	I	Timer 14 capture input	
				EC14	I	Timer 14 event count input	
				TXD11	O	UART data output	
				MOSI11	IO	SPI master out slave in	
				SDA2	IO	I2C data input / output	
43	31	25	20	PA10*	IOUDS	Port A bit 10 input / output	
				T15OUT	O	Timer 15 pulse output	
				T15CAP	I	Timer 15 capture input	
				EC15	I	Timer 15 event count input	
				RXD11	I	UART data input	
				MISO11	IO	SPI master in slave out	
44	32	26	21	USBDM	IO	USB data line minus If the USB block is not used, the pin should be connected to a pull-down resistor, either internal or external.	
45	33	27	22	USBDP	IO	USB data line plus If the USB block is not used, the pin should be connected to a pull-down resistor, either internal or external.	
46	34	28	23	PA13	IOUDS	Port A bit 13 input / output	
				SWDIO*	I/O	SWD data input / output	Pull-up when reset
				SCK10	IO	SPI clock input / output	
47	35	29	-	VSS	P	Ground	
48	36	30	24	VUSB33	P	3.3V power input or 3.3V LDO output for USB block. If the USB block is not used, the VUSB33 pin must be connected to VDD. A 0.1uF capacitor should be connected between this pin and ground.	

Table 3. Pin Description (continued)

Pin No.				Pin Name	Type[1]	Description	Remark
64-Pin	48-Pin	40-Pin	32-Pin				
49	37	31	25	PA14	IOUDS	Port A bit 14 input / output	
				SWCLK*	I	SWD clock input	Pull-down when reset
				TXD10	O	UART data output	
				MOSI10	IO	SPI master out slave in	
50	38	32	26	PA15*	IOUDS	Port A bit 15 input / output	
				EC21	I	Timer 21 event count input	
				RXD10	I	UART data input	
				MISO10	IO	SPI master in slave out	
				SS0	I	SPI slave select input	
				SS13	I	SPI slave select input	
51	-	-	-	PC10*	IOUDS	Port C bit 10 input / output	
				TXD12	O	UART data output	
				MOSI12	IO	SPI master out slave in	
				TXD13	O	UART data output	
				MOSI13	IO	SPI master out slave in	
52	-	-	-	PC11*	IOUDS	Port C bit 11 input / output	
				EC21	I	Timer 21 event count input	
				RXD12	I	UART data input	
				MISO12	IO	SPI master in slave out	
				RXD13	I	UART data input	
				MISO13	IO	SPI master in slave out	
53	-	-	-	PC12*	IOUDS	Port C bit 12 input / output	
				SCK12	IO	SPI clock input / output	
				SCK13	IO	SPI clock input / output	
54	-	-	-	PF2*	IOUDS	Port F bit 2 input / output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				SS12	I	SPI slave select input	
55	39	-	-	PB3*	IOUDS	Port B bit 3 input / output	
				T20OUT	O	Timer 20 pulse output	
				T20CAP	I	Timer 20 capture input	
				SCK0	IO	SPI clock input / output	
56	40	-	-	PB4*	IOUDS	Port B bit 4 input / output	
				T15OUT	O	Timer 15 pulse output	
				T15CAP	I	Timer 15 capture input	
				EC15	I	Timer 15 event count input	
				MISO0	IO	SPI master in slave out	
57	41	33	-	PB5*	IOUDS	Port B bit 5 input / output	
				T21OUT	O	Timer 21 pulse output	
				T21CAP	I	Timer 21 capture input	
				MOSI0	IO	SPI master out slave in	

Table 3. Pin Description (continued)

Pin No.				Pin Name	Type(1)	Description	Remark
64-Pin	48-Pin	40-Pin	32-Pin				
58	42	34	27	PB6*	IOUDS	Port B bit 6 input / output	
				T11OUT	O	Timer 11 pulse output	
				T11CAP	I	Timer 11 capture input	
				EC11	I	Timer 11 event count input	
				TXD11	O	UART data output	
				MOSI11	IO	SPI master out slave in	
				SCL0	IO	I2C clock input / output	
59	43	35	28	PB7*	IOUDS	Port B bit 7 input / output	
				T12OUT	O	Timer 12 pulse output	
				T12CAP	I	Timer 12 capture input	
				EC12	I	Timer 12 event count input	
				RXD11	I	UART data input	
				MISO11	IO	SPI master in slave out	
				SDA0	IO	I2C data input / output	
60	44	36	29	PD5	IOUDS	Port D bit 5 input / output	
				BOOT*	IU	Boot mode input	Pull-up when reset
				RTCOUT	O	Real time clock output	
				CLKO	O	System clock output	
				CMUSYNC	I	HSI clock management unit sync input	
61	45	37	30	PB8*	IOUDS	Port B bit 8 input / output	
				T14OUT	O	Timer 14 pulse output	
				T14CAP	I	Timer 14 capture input	
				EC14	I	Timer 14 event count input	
				SCK11	IO	SPI clock input / output	
				SCL0	IO	I2C clock input / output	
62	46	38	-	PB9*	IOUDS	Port B bit 9 input / output	
				T15OUT	O	Timer 15 pulse output	
				T15CAP	I	Timer 15 capture input	
				EC15	I	Timer 15 event count input	
				SS1	I	SPI slave select input	
				SS11	I	SPI slave select input	
				SDA0	IO	I2C data input / output	
63	47	39	31	VSS	P	Ground	
64	48	40	32	VDD	P	VDD	

NOTES:

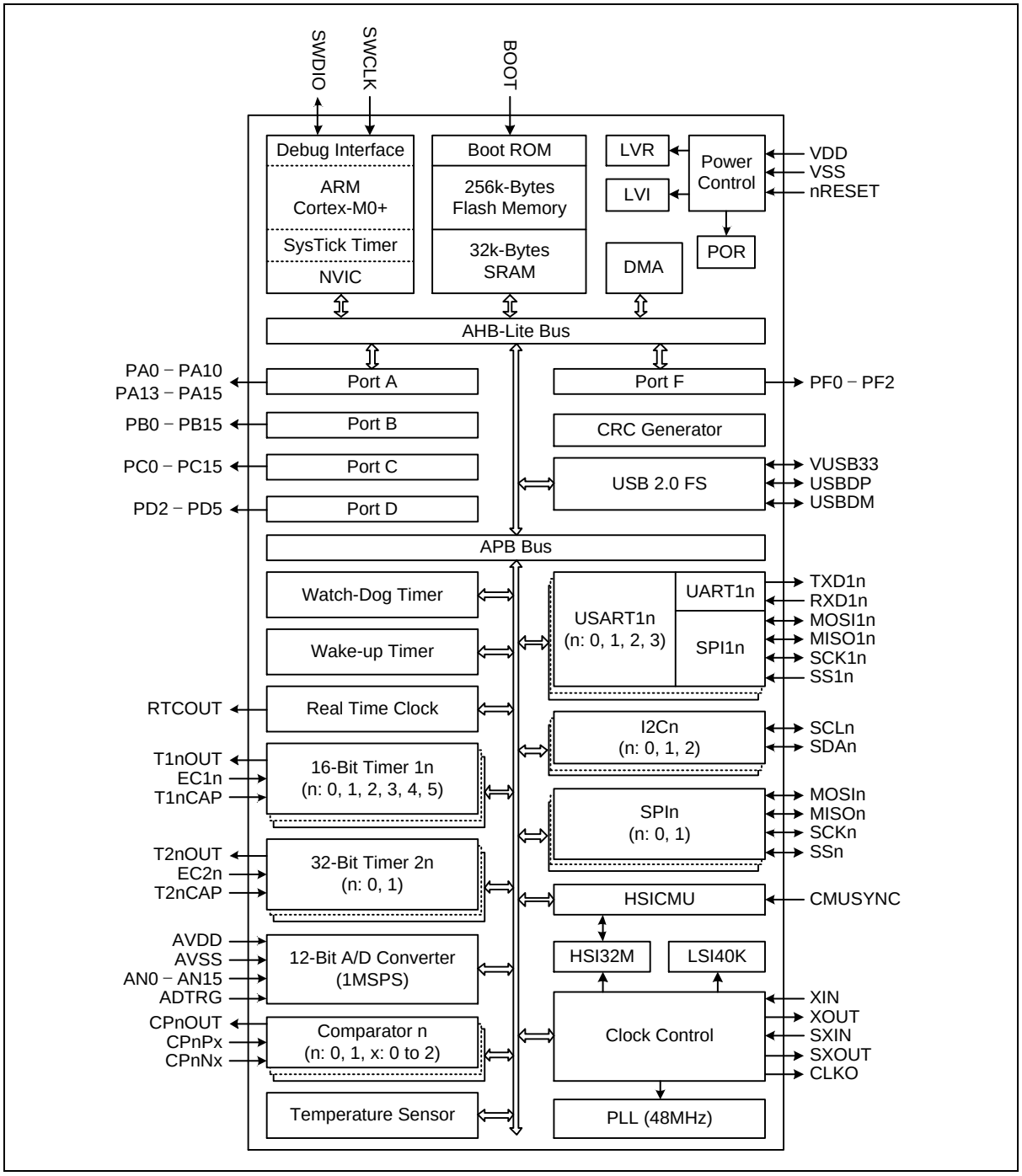
1. I = Input, O = Output, U = Pull-up, D = Pull-down, S = Schmitt-Trigger Input Type, C = CMOS Input Type, A = Analog, P = Power
2. * means ‘selected pin function after reset condition’.
3. Pin orders may be changed with revision notice.
4. nRESET and PA13 (SWDIO) are the default pull-up pins. PA14 (SWCLK) is the default pull-down pin.
5. An unused pin should not be configured as an input floating.
6. After a reset, the PA13 and PA14 pins are configured as SWDIO and SWCLK alternate functions, and the internal pull-down for the serial wire clock (SWCLK) and the internal pull-up for the serial wire data I/O (SWDIO) are enabled.
7. The SWCLK and SWDIO pins should not be switched to other functions while they are being used.

3. System and Memory Overview

In this chapter, the system and memory overview of the A31G336 are described.

3.1 System Architecture

Figure 8. System Block Diagram



3.1.1 Cortex-M0+ Core

The Cortex-M0+ processor is the most energy-efficient Arm processor available. It builds on the very successful Cortex-M0+ processor, retaining full instruction set and tool compatibility, while further reducing energy consumption and increasing performance.

Arm's technical reference manual "DDI 0484C" provides detailed information on Cortex-M0+.

3.1.2 Interrupt Controller

The Cortex-M0+ processor has an embedded interrupt controller named NVIC (Nested Vector Interrupt Controller). The A31G336 has an additional interrupt control block for controlling 32 interrupt sources generated by internal peripherals.

To use interrupts from internal peripherals, both the NVIC and the interrupt control block must be configured properly.

This document only describes the peripheral interrupt controller, therefore for more information on NVIC inside the Cortex-M0+ processor, please refer to the technical reference manual "ARM DDI 0484C" on the Arm technical document site.

Table 4. Interrupt Vector Map

Priority	Vector Address	Interrupt Source
-16	0x0000_0000	Stack Pointer
-15	0x0000_0004	Reset Address
-14	0x0000_0008	NMI Exception
-13	0x0000_000C	Hard Fault Exception
-12	0x0000_0010	Reserved
-11	0x0000_0014	
-10	0x0000_0018	
-9	0x0000_001C	
-8	0x0000_0020	
-7	0x0000_0024	
-6	0x0000_0028	
-5	0x0000_002C	SVCall Exception
-4	0x0000_0030	Reserved
-3	0x0000_0034	
-2	0x0000_0038	PenSV Exception
-1	0x0000_003C	SysTick Exception

Table 4. Interrupt Vector Map (continued)

Priority	Vector Address	Interrupt Source
0	0x0000_0040	LVI, ECCC Interrupt
1	0x0000_0044	WUT Interrupt
2	0x0000_0048	WDT Interrupt
3	0x0000_004C	EINT0 Interrupt
4	0x0000_0050	EINT1 Interrupt
5	0x0000_0054	EINT2 Interrupt
6	0x0000_0058	EINT3 Interrupt
7	0x0000_005C	TIMER10 Interrupt
8	0x0000_0060	TIMER11 Interrupt
9	0x0000_0064	TIMER12 Interrupt
10	0x0000_0068	I2C0 Interrupt
11	0x0000_006C	USART10 Interrupt
12	0x0000_0070	RTCC Interrupt
13	0x0000_0074	Reserved
14	0x0000_0078	I2C1 Interrupt
15	0x0000_007C	TIMER20 Interrupt
16	0x0000_0080	TIMER21 Interrupt
17	0x0000_0084	USART11 Interrupt
18	0x0000_0088	ADC Interrupt
19	0x0000_008C	CMP[1:0] Interrupt
20	0x0000_0090	TS Interrupt
21	0x0000_0094	TIMER13 Interrupt
22	0x0000_0098	TIMER14 Interrupt
23	0x0000_009C	TIMER15 Interrupt
24	0x0000_00A0	DMACH[1:0] Interrupt
25	0x0000_00A4	I2C2 Interrupt
26	0x0000_00A8	USART12 Interrupt
27	0x0000_00AC	USART13 Interrupt
28	0x0000_00B0	DMACH[3:2] Interrupt
29	0x0000_00B4	SPI0 Interrupt
30	0x0000_00B8	SPI1 Interrupt
31	0x0000_00BC	USB Interrupt

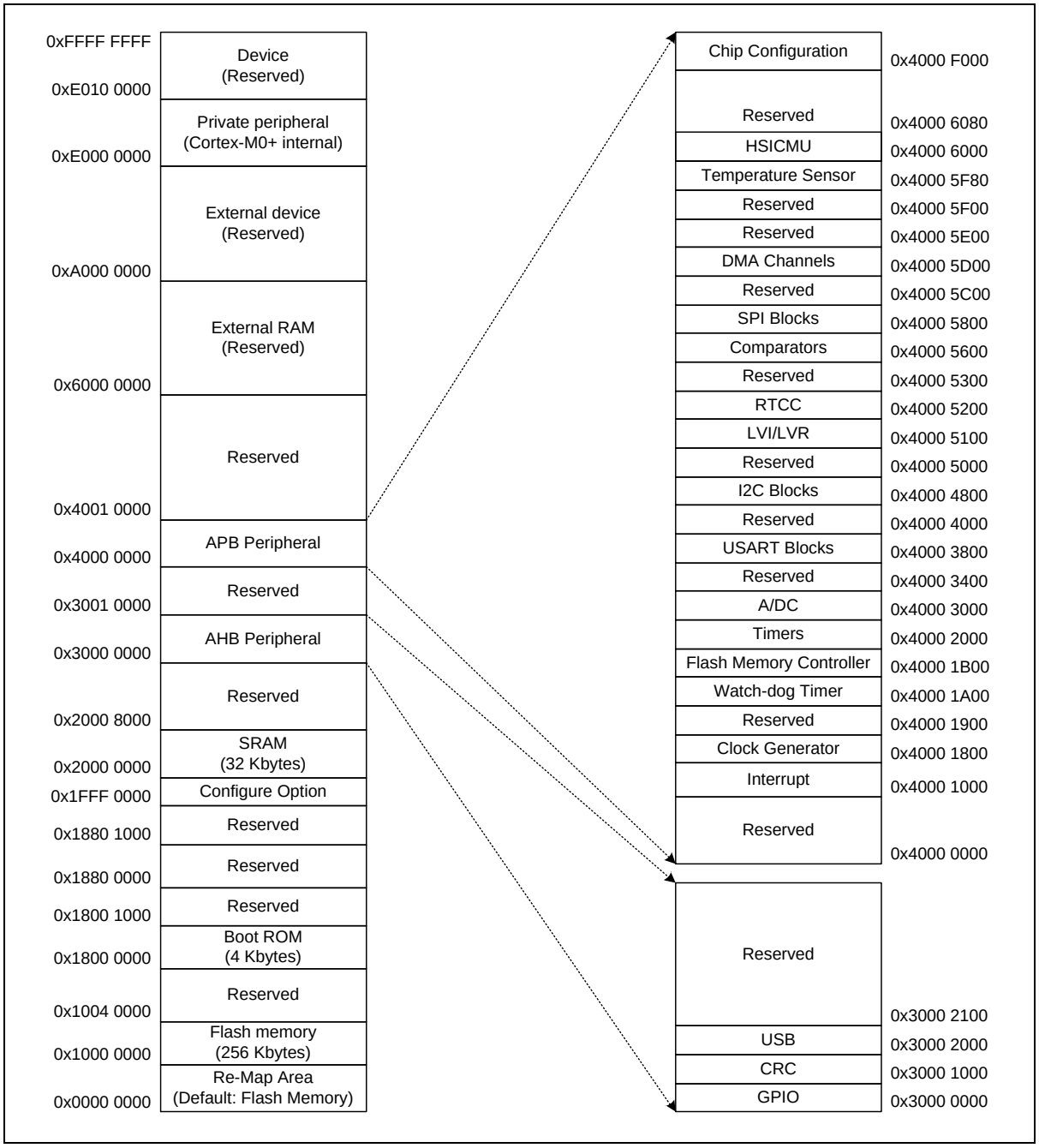
3.2 Memory Organization

3.2.1 Introduction

Program memory, data memory, registers, and I/O ports are organized in the same address space. The bytes are coded in memory in Little Endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte the most significant.

3.2.2 Memory Map and Register Boundary Address

Figure 9. Memory Map



All the memory map areas that are not allocated to on-chip memories and peripherals are considered “Reserved”. For the detailed mapping of available memory and register areas, refer to the following table.

The following table gives the boundary addresses of the peripherals available in the devices.

Table 5. A31G336 Memory Map and Peripheral Register Boundary Address

Boundary Address	Memory Area
0x1000_0000	Flash memory
0x2000_0000	Internal SRAM
0x3000_0000	GPIO
0x3000_1000	CRC
0x3000_2000	USB
0x4000_1000	Interrupt
0x4000_1800	SCUCG
0x4000_1A00	WDT
0x4000_1B00	FMC
0x4000_2000	Timers
0x4000_3000	ADC
0x4000_3800	USARTs
0x4000_4800	I2Cs
0x4000_5100	SCULV
0x4000_5200	RTCC
0x4000_5600	Comparators
0x4000_5800	SPIs
0x4000_5D00	DMAs
0x4000_5F80	Temperature Sensor
0x4000_6000	HSICMU
0x4000_F000	SCUCC

3.3 Memory Map

3.3.1 Embedded SRAM

The A31G336 has a block of 0-wait on-chip SRAM. Size of the SRAM is 32 KB and its base address is 0x2000_0000.

This SRAM memory area is usually used for data memory and stack memory. Sometimes the code is dumped into the SRAM memory for fast operation or Flash erase / programming operation. This device does not support memory remap strategy. So, jump and return are required to perform the code in SRAM memory area.

3.3.2 Flash Memory Overview

The A31G336 provides internal 256 KB code Flash memory with ECC and a controller. This is enough to control the general system. Self-programming is available and ISP and SWD programming is also supported in boot mode or debugging mode.

CPU can access Flash memory with two wait state up to 48 MHz bus frequency.

3.4 Boot Configuration

The A31G336 has a boot mode option to program internal Flash memory. Boot mode can be entered by setting BOOT pin to ‘L’ at reset timing (Normal state is ‘H’).

Boot mode supports the UART boot which uses the UART interface to transmit and receive boot data (TXD10/RXD10).

The pins for the boot mode are listed in Table 6.

Table 6. Boot Mode Pin List

Block	Pin Name	Pin Direction	Description
SYSTEM	nRESET	Input	Reset input signal
	BOOT / PD5	Input	Boot mode setting pin
UART mode of USART10	RXD10 / PA3	Input	Receive the UART boot data.
	TXD10 / PA2	Output	Transmit the UART boot data.

4. System Control Unit (SCU)

4.1 SCU Introduction

The A31G336 has a built-in intelligent power control block which manages system analog blocks and operating modes. System Control Unit (SCU) module controls an internal reset and clock signals to maintain optimized system performance and power dissipation.

4.2 SCU Main Features

The clock features of the A31G336 are as follows:

- Operating frequency: Up to 48 MHz
- High-Speed internal oscillator (HSI): 2 MHz to 32 MHz
- Watch-dog timer internal oscillator (LSI): 40 kHz
- High-Speed External Oscillator (HSE): 2 MHz to 48 MHz
- Low-Speed External Oscillator (LSE): 32.768 kHz
- Phase-locked Loop (PLL): 48 MHz
- A31G336 has clock monitoring as a system fail-safe function.

The reset features of the A31G336 are as follows:

- nRESET pin reset
- CPU reset
- Software reset
- Power-On Reset (POR)
- Low-Voltage Detect Reset (LVR)
- Watchdog Timer Reset (WDT)
- Clock Monitoring Reset (MON)

Operating mode features of the A31G336 are as follows:

- RUN mode
- SLEEP mode
- DEEP-SLEEP mode

4.3 Clocks

4.3.1 External Oscillation Clock

- High-Speed External Oscillator (HSE): 2 MHz to 48 MHz
  - X-TAL Oscillator: 2 MHz to 16 MHz
  - External Clock: 2 MHz to 48 MHz
  - Generates HSE clocks by connecting oscillator to XIN pin and XOUT pin.
- Low-Speed External Oscillator (LSE): 32.768 kHz
  - Generates LSE clocks by connecting oscillator to SXIN pin and SXOUT pin.

4.3.2 Internal Oscillation Clock

- High-Speed internal oscillator (HSI): 2 MHz to 32 MHz
- Low-Speed internal oscillator (LSI): 40 kHz

4.3.3 PLL Clock

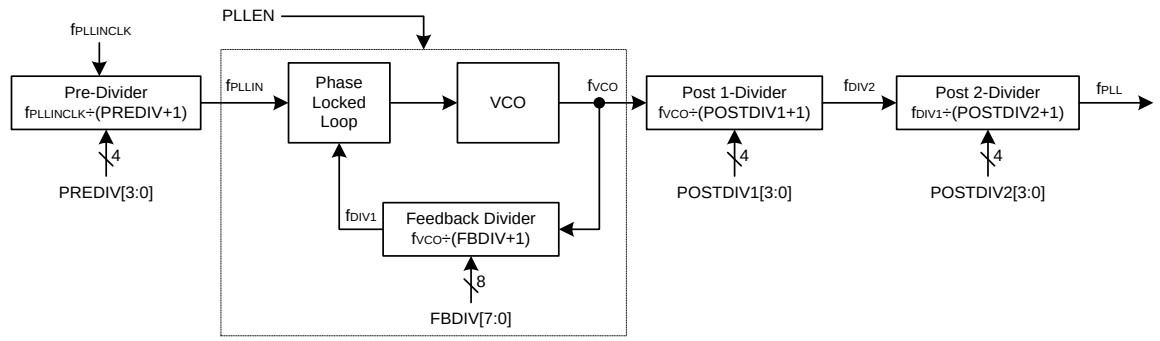
Phase-locked Loop (PLL) can be used to multiply the HSI or HSE output clock frequencies.

The PLL frequency generator generates a high-speed clock (up to 48 MHz). For more information on the PLL frequency generator, refer to Figure 10 and 4.9.27.

Users must complete the PLL configuration (selection of the input clock and multiplication factor) before enabling the PLL.

Once PLL is enabled, these parameters cannot be changed.

Figure 10. PLL Block Diagram



NOTES:

1. PLLEN ='1' to enable PLL block & set PLL Frequency.
2. Wait 200 us for the stabilization of PLL output.
3. Wait until PLLLOCK flag occurred.
4. The output calculation formula is as follows:

$$f_{\mathrm{PLL}} = \frac{f_{\mathrm{PLLINCLK}} \times (\mathrm{FB}+1)}{(\mathrm{R}+1) \times (\mathrm{N}_1+1) \times (\mathrm{N}_2+1)}$$

Register	Bit	Symbol	Description
SCU_PLLCON	PREDIV[3:0]	R	PLLINCLK pre-divider value
	POSTDIV1[3:0]	N_1	Post multiplier value
	POSTDIV2[3:0]	N_2	Post divider value
	FBDIV[7:0]	FB	Feedback divider value

4.3.3.1 PLL Output Frequency Settings

For the A31G336, the PLL's output frequency f_{PLL} can be precisely set in 1 MHz steps. The formula below calculates the input frequency (f_{PLLIN}) fed to the PLL's VCO. The frequency range from 1 MHz to 3 MHz is supported, but it is recommended to set the f_{PLLIN} to 2 MHz if possible.

$$f_{\mathrm{PLLIN}} = \frac{\mathrm{PLLINCLK}}{(\mathrm{R}+1)}, \qquad 1\ \mathrm{MHz} \le f_{\mathrm{PLLIN}} \le 3\ \mathrm{MHz}\ (\mathrm{Recommended}\ f_{\mathrm{PLLIN}} = 2\ \mathrm{MHz})$$

At this time, the range of VCO frequency, f_{VCO} should be set to Max. VCO frequency (Max. f_{VCO}) or less, and the calculation formula is shown below. (Refer to PLL electrical characteristics in Datasheet.):

$$f_{\mathrm{VCO}}\ (\mathrm{MHz}) = f_{\mathrm{PLLIN}} \times (\mathrm{FB}+1), f_{\mathrm{VCO}} \le \mathrm{Max.}\, f_{\mathrm{VCO}}$$

As a result, the final frequency of the PLL block, f_{PLLOUT} can be obtained from the formula below:

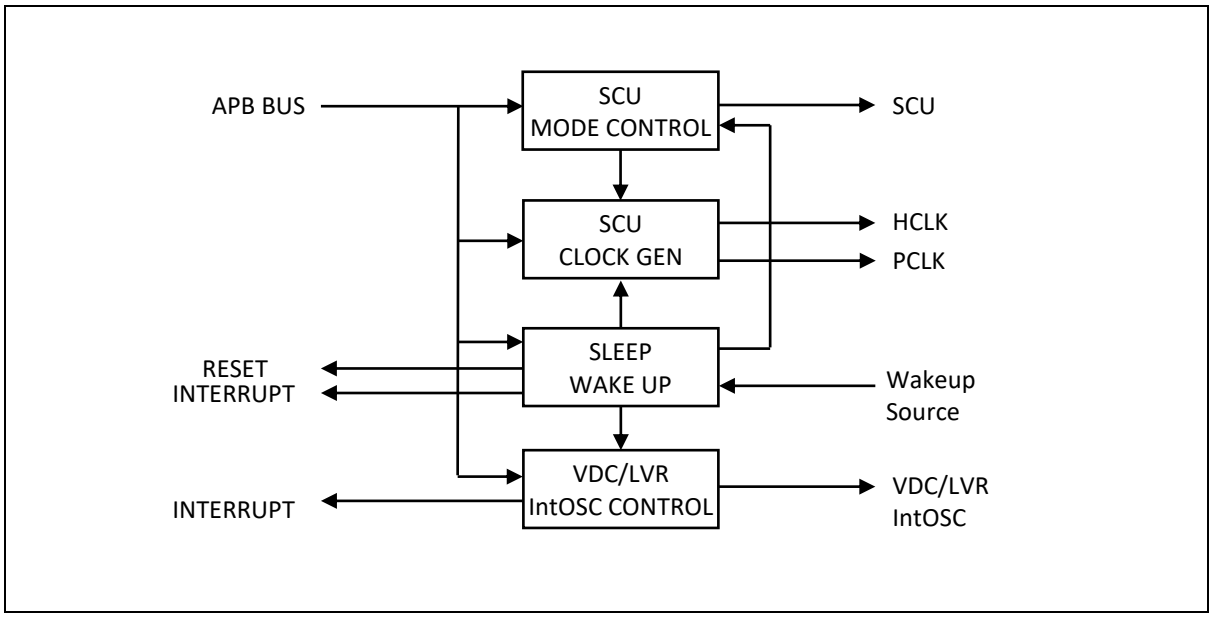
$$f_{\mathrm{PLL}}\ [\mathrm{MHz}] = \frac{f_{\mathrm{PLLINCLK}} \times (\mathrm{FB}+1)}{(\mathrm{R}+1) \times (\mathrm{N}_1+1) \times (\mathrm{N}_2+1)}$$

$$= \frac{f_{\mathrm{PLLIN}} \times (\mathrm{FB}+1)}{(\mathrm{N}_1+1) \times (\mathrm{N}_2+1)}$$

4.4 SCU Block Diagram

Figure 11 shows the SCU block diagram.

Figure 11. SCU Block Diagram

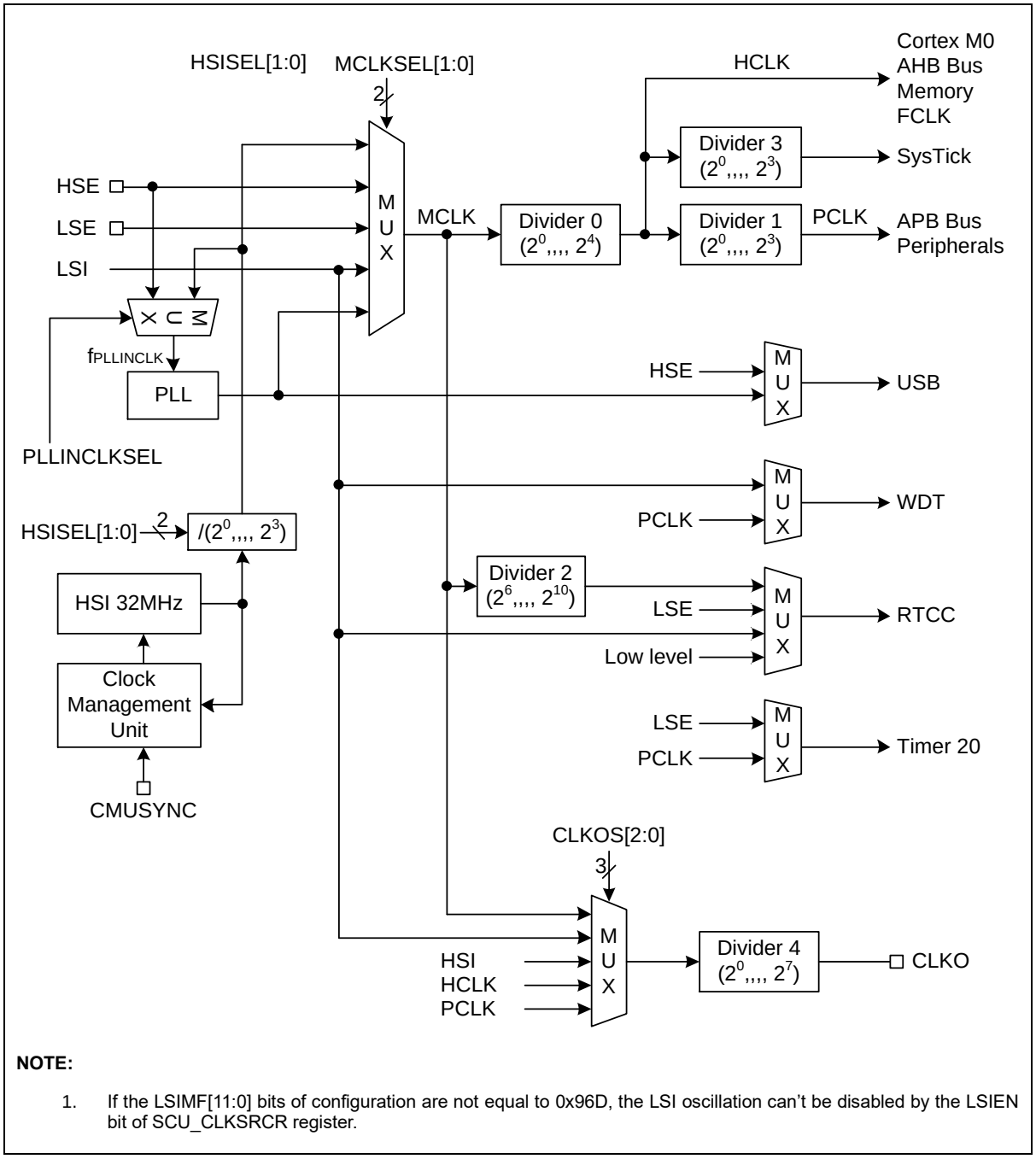


4.5 Clock System

The A31G336 has two main operating clocks. One is the HCLK, which supplies clocks to the CPU and AHB bus system, and the other is the PCLK, which supplies clocks to the peripheral systems.

Users can control the clock system variation using software. Figure 12 shows the clock system of the A31G336 and Table 7 shows the description for clock sources.

Figure 12. Clock Source Configuration



NOTE:

1. If the LSIMF[11:0] bits of configuration are not equal to 0x96D, the LSI oscillation can't be disabled by the LSIEN bit of SCU_CLKSRCR register.

Each MUX that switches the clock source has a glitch-free circuit. So, clocks can be switched without glitch risks. When users change the clock mux control, be sure that both clock sources are alive. If either is not alive, the clock change operation stops, and the system shuts down (and won't recover).

Table 7. Clock Sources

Clock name	Mnemonic	Frequency	Description
Main OSC	HSE	X-TAL (2 MHz to 16 MHz) External Clock (2 MHz to 48 MHz)	External Main Crystal OSC External Main Clock
Sub OSC	LSE	X-TAL (32.768 kHz)	External Sub Crystal OSC
High-Speed Internal RC OSC	HSI	2 MHz to 32 MHz	High-Speed Internal RC OSC
Low-Speed Internal RC OSC	LSI	40 kHz	Low-Speed Internal RC OSC
PLL clock	PLL	4 to 48 MHz	Phase-locked loop frequency generator generates a high-speed clock

4.5.1 HCLK Clock Domain

The HCLK clock feeds the clock to the CPU and AHB bus. Cortex-M0+ CPU requires two clocks, FCLK and HCLK. The FCLK is a free running clock and is always running except during power down mode. The HCLK can be stopped during SLEEP mode.

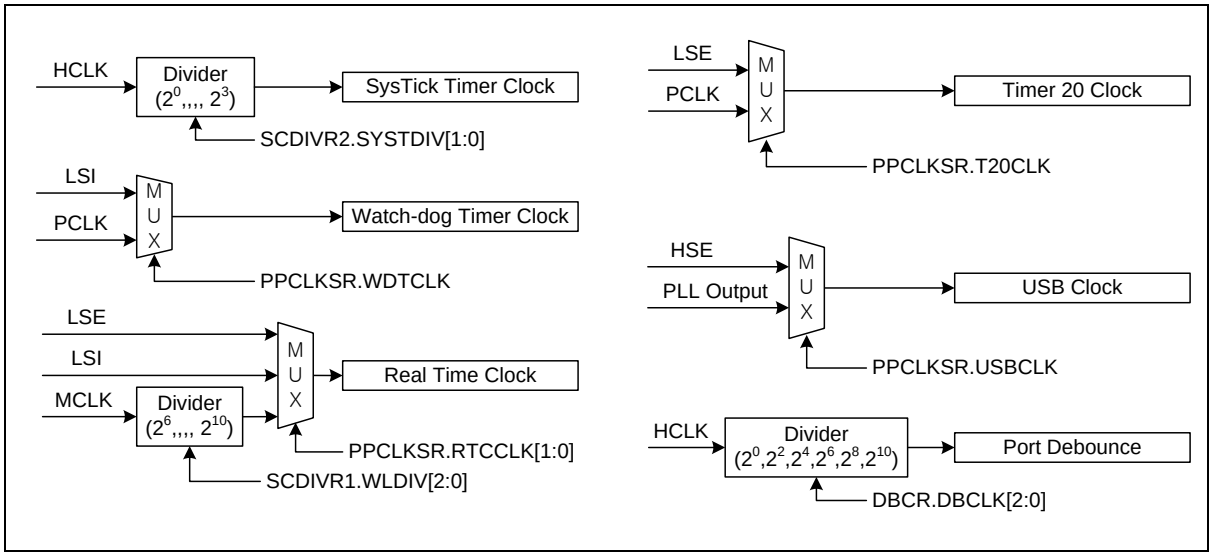
The HCLK clock operates the BUS system and memory systems. The maximum clock speed of the BUS operation is 48 MHz. The HCLK frequency should be limited to 48 MHz or lower.

4.5.2 Miscellaneous Clock Domain

Various clock sources are required for each functional block. The SCU provides clock source selectivity with dedicated pre-scaler for each functional block. The clock selection mux does not support glitch-free function, so the clock is unpredictable during clock selection.

Figure 13 shows the configurations of miscellaneous clocks.

Figure 13. Miscellaneous Clock Configuration



4.5.3 PCLK Clock Domain

The PCLK is the master clock for all peripherals except for the CRC generator, USB block, and ports. It can shut down during power down modes. Each peripheral clock is generated by the SCU_PPCLKEN1 and SCU_PPCLKEN2 register set.

Figure 12 illustrates the PCLK clock distributions. Peripherals are not accessible until the PCLK clock of each block is enabled, and even by reading its registers.

4.5.4 Clock Configuration Procedure

When the device is powered on, a default system clock is generated by the HSI (2 MHz) clock. The HSI is enabled by default during the power up sequence. Other clock sources are enabled by user controls and configuration options with a system clock.

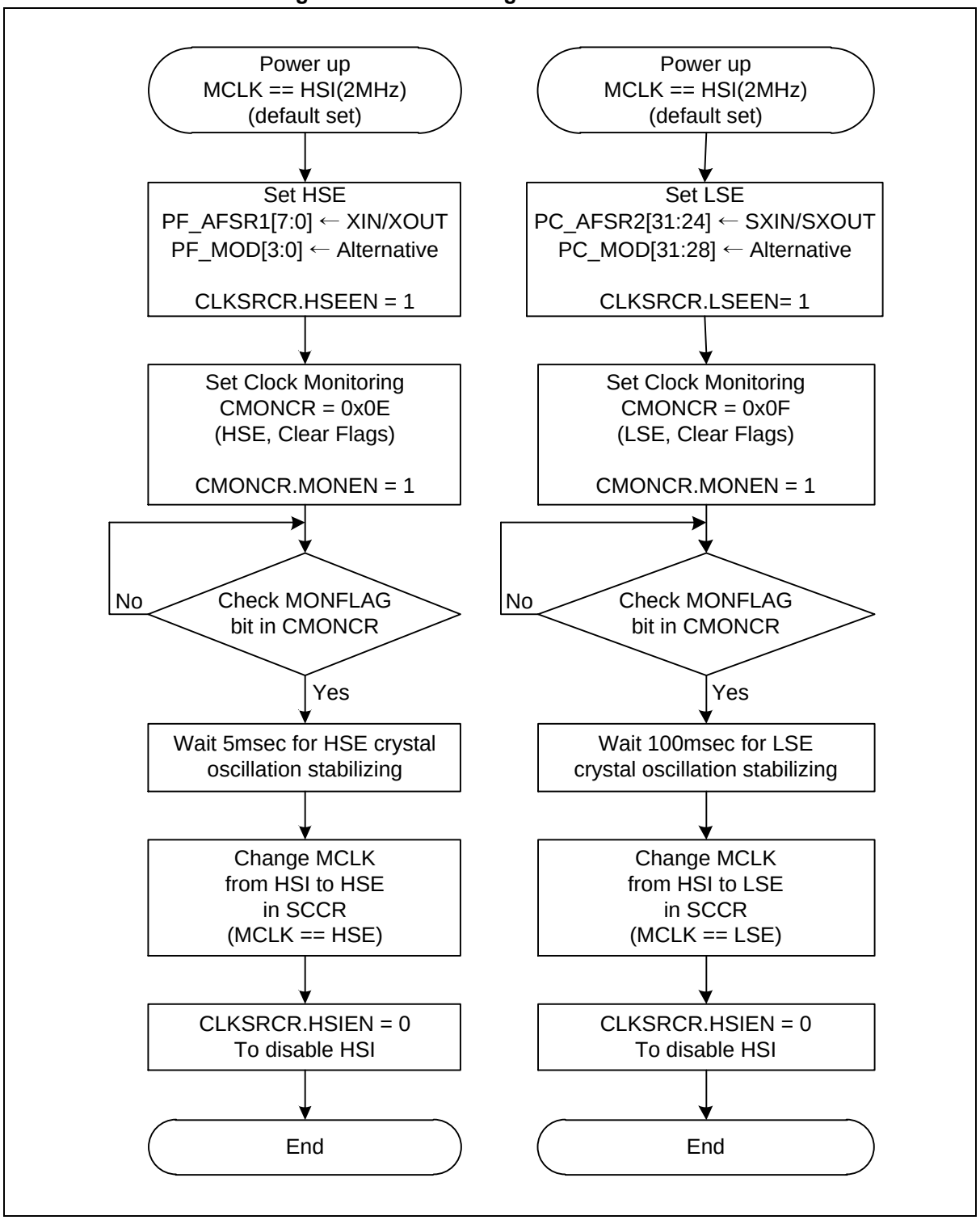
The HSE and LSE clocks are enabled by the HSEEN and LSEEN bits of the SCU_CLKSRCR register, respectively. Before enabling HSE and LSE blocks, the pin mux configuration should be set for XIN/XOUT and SXIN/SXOUT functions.

The PF0/PF1 and PC14/PC15 pins are shared by the HSE's XIN/XOUT function and LSE's SXIN/SXOUT function – The PF_MOD/PC_MOD and PF_AFSR1/PC_AFSR2 registers should be configured properly.

After enabling the HSE and LSE blocks, a user can check the stability of crystal oscillation through the clock monitoring control register, SCU_CMONCR. It takes more than 1ms to ensure stable crystal oscillation before changing the system clock.

Figure 14 shows an example of a flow chart configuring the system clock as the HSE and LSE clocks.

Figure 14. Clock Configuration Procedure



4.6 Reset

4.6.1 Overview

The A31G336 has two system resets. One is the cold reset by POR, which is effective during power up or down sequence, and the other is the warm reset, which is generated by several reset sources. The reset event makes the device turn back to its initial state.

The cold reset has a single reset source (POR), and the warm reset has several reset sources listed below:

- nRESET pin
- WDT reset
- LVR reset
- MON reset
- Software reset
- CPU request reset

4.6.2 Cold Reset

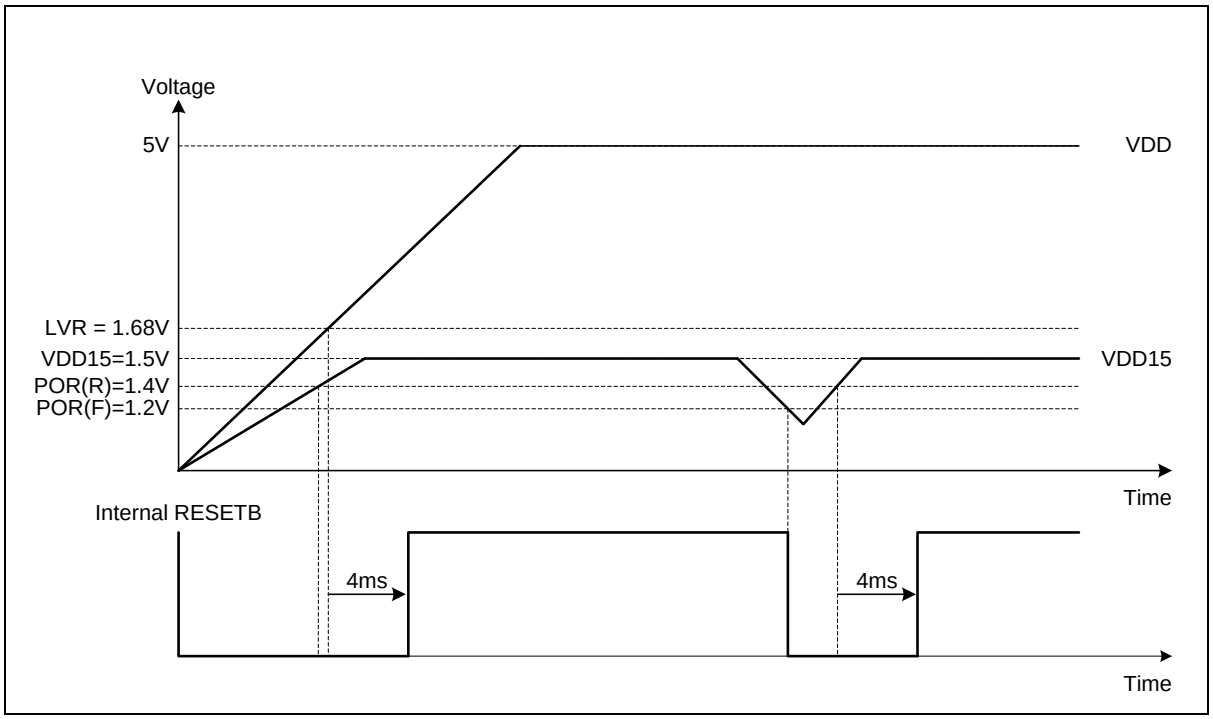
The cold reset is one of the important features of the A31G336 when powered on. This characteristic globally affects the system boot.

The internal VDC is enabled when the VDD power is supplied. The internal VDD level slope follows the external VDD power slope.

The internal POR trigger level is at 1.4 V of the internal VDC voltage. At this time, the boot operation begins. The internal RC clock turns on and counts 4ms for the internal VDC level to stabilize. At this time, the external VDD voltage level should be bigger than the initial LVR level (1.68 V). After 4ms of counting, the CPU reset is released, and operation begins.

Figure 15 shows the waveform of the power up sequence and internal reset.

Figure 15. Power-Up POR Sequence



The register SCU_RSTSSR shows the POR reset status. The last reset comes from the POR.

The PORSTA bit in the SCU_RSTSSR register is set to '1'. After power on, this bit is always set to '1' unless it is cleared by software. If an abnormal internal voltage drop is detected during normal operation, the system is reset, and this bit is set to '1'.

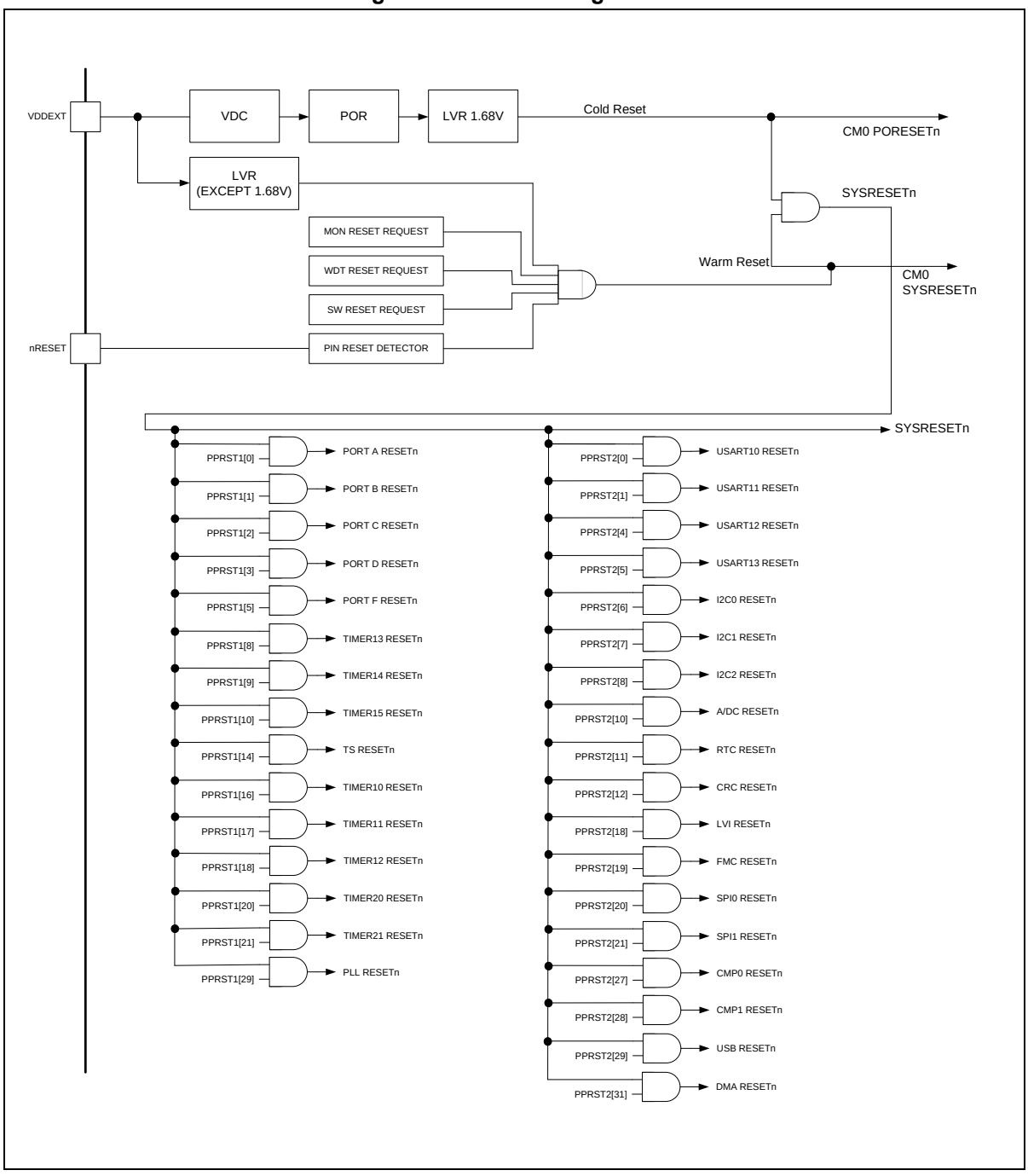
When the cold reset is applied, the entire device returns to its initial state.

4.6.3 Warm Reset

The warm reset event has several reset sources, and some parts of the device return to their initial states when the warm reset takes place.

The warm reset status appears in the register SCU_RSTSSR. A reset for each peripheral block is controlled by the register SCU_PPRST. The reset can be masked independently.

Figure 16. Reset Configuration

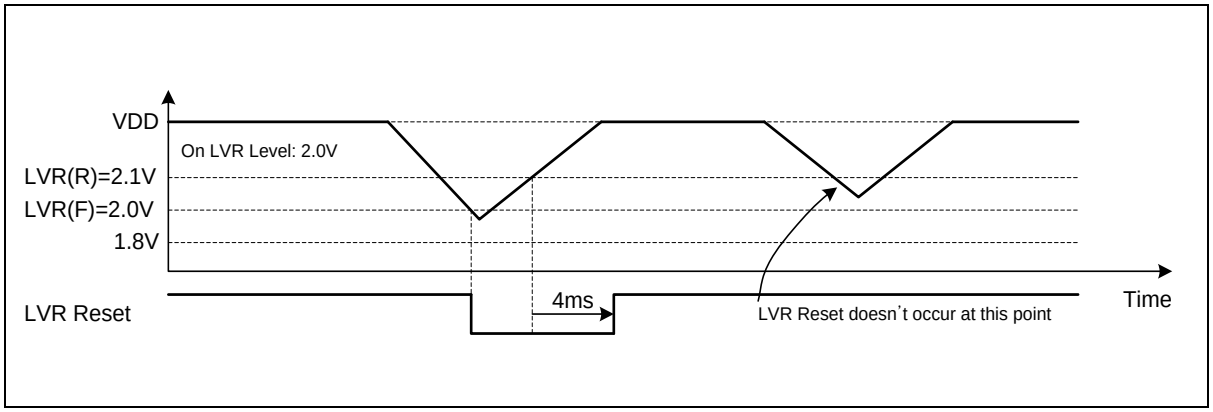


4.6.4 LVR Reset

The LVR voltage level is set by a low-voltage reset configuration register (CONF_LVRCNFIG) in the configure option page 1. The LVR reset status appears in the register SCU_RSTSSR.

The LVR reset is controlled by the register SCU_LVRCR. This register is cleared to “0x00” when the POR reset occurs.

Figure 17. LVR Reset Timing Diagram

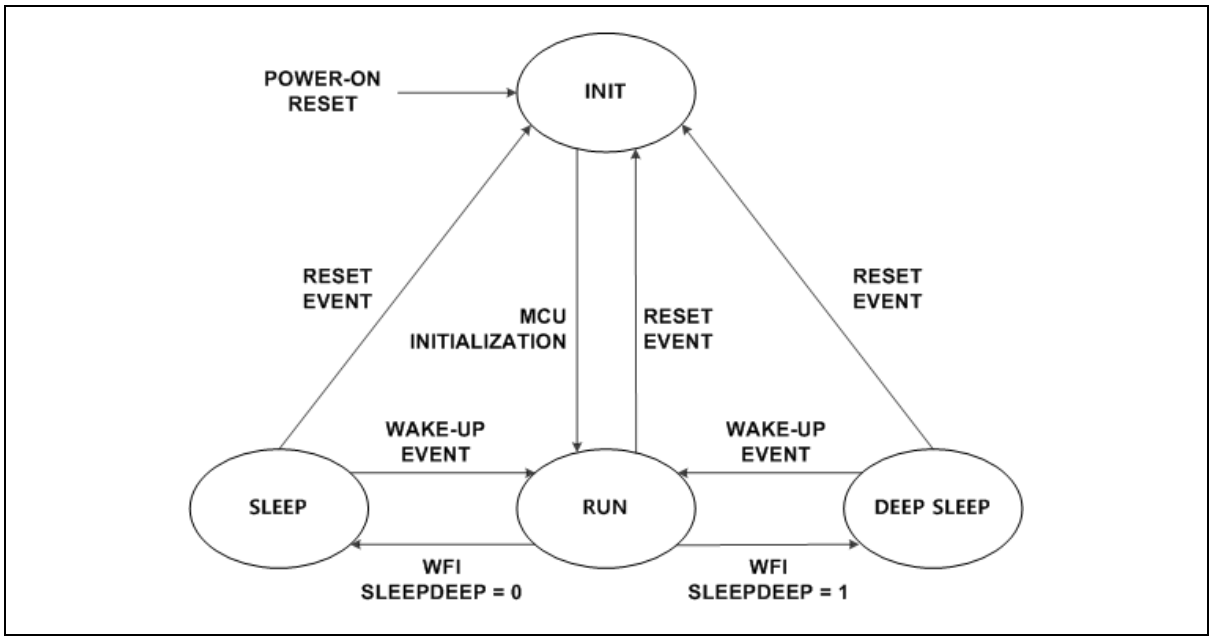


4.7 Operation Mode

INIT mode is the initial state of the device when resetting. At RUN mode, the chip runs at its max CPU performance with a high-speed clock system. At SLEEP and DEEP SLEEP mode, the chip runs at a low power consumption mode. The system saves power by halting the processor core and unused peripherals.

Figure 18 shows a diagram of the operation mode transition.

Figure 18. Operation Mode Transition



4.7.1 RUN Mode

This mode is to operate CPU core and peripheral hardware with a high-speed clock. The device enters the INIT state after resetting and then enters the RUN mode.

4.7.2 SLEEP Mode

In this mode, only the CPU is stopped. Each peripheral function is turned on by the function enable bit and the clock enable bit of the register SCU_PPCLKEN.

4.7.3 DEEP SLEEP Mode

In this mode, not only the CPU but also the selected system clock (MCLK) is stopped. The Watch timer with a sub clock and the Watchdog Timer with LSI still operate in this mode.

Table 8. Functional Table on Current Mode

IP	Main Run (IDD1)	Main Sleep (IDD2)	Sub Run (IDD3)	Sub Sleep (IDD4)	Deep Sleep (IDD5)
CPU	O	X	O	X	X
Flash	O	X	O	X	X
SRAM	O	X	O	X	X
FMC	Optional	X	Optional	X	X
CRC	Optional	X	Optional	X	X
POR	O	O	O	O	O
LVR/LVI	Optional	Optional	Optional	Optional	Optional
GPIO	O	O	O	O	O
SCU	O	O	O	O	O
SPI	Optional	Optional	Optional	Optional	X
I2C	Optional	Optional	Optional	Optional	Optional
USART	Optional	Optional	Optional	Optional	Optional
SysTick	Optional	Optional	Optional	Optional	X
T10 – T15	Optional	Optional	Optional	Optional	X
T20	Optional	Optional	Optional	Optional	Optional
T21	Optional	Optional	Optional	Optional	X
WDT	Optional	Optional	Optional	Optional	Optional
WUT	O	O	O	O	X
A/DC	Optional	X	X	X	X
Comparator	Optional	Optional	Optional	Optional	Optional
TS	Optional	Optional	X	X	X
RTCC	Optional	Optional	Optional	Optional	Optional
USB	Optional	Suspend/Resume	Suspend/Resume	Suspend/Resume	Suspend/Resume
HSI	O	O	X	X	X
LSI	Optional	Optional	Optional	Optional	Optional
HSE	Optional	Optional	X	X	X
LSE	Optional	Optional	Optional	Optional	Optional
PLL	Optional	Optional	X	X	X

NOTES:

1. O: Enable, X: Disable, Optional: A function can be disabled/enabled by software.
2. It can be woken up from sleep and deep sleep modes by an interrupt source of the optional peripherals and USB resume signal.

4.8 Pins for SCU

Table 9. Pins and External Signals for SCU

Pin Name	Type	Description
nRESET	I	External Reset Input
XIN, XOUT	OSC	External Crystal Oscillator for Main Clock
SXIN, SXOUT	OSC	External Crystal Oscillator for Sub Clock
CLKO	O	Clock Output Monitoring Signal

4.9 SCU Registers

The base address and register map of SCU (chip configuration) are described in the following:

Table 10. Base Address of SCU (Chip Configuration)

Name	Base Address
SCU (chip configuration)	0x4000_F000

Table 11. SCU (Chip Configuration) Register Map

Name	Offset	Type	Description	Reset Value	Reference
SCU_VENDORID	0x0000	RO	Vendor Identification Register	0x4142_4F56	4.9.1
SCU_CHIPID	0x0004	RO	Chip Identification Register.	0x4731_3360	4.9.2
SCU_REVNR	0x0008	RO	Revision Number Register	0x0000_00xx	4.9.3
SCU_PMREMAP	0x0014	RW	Program Memory Remap Register	0x0000_0000	4.9.4
SCU_BTPSCR	0x0018	RW	Boot Pin Status and Control Register	0x0000_00xx	4.9.5
SCU_RSTSSR	0x001C	RW	Reset Source Status Register	0x0000_00xx	4.9.6
SCU_NMISRCR	0x0020	RW	NMI Source Selection Register	0x0000_0020	4.9.7
SCU_SWRSTR	0x0024	WO	Software Reset Register	0x0000_0000	4.9.8
SCU_SRSTVR	0x0028	RO	System Reset Validation Register	0x0000_0055	4.9.9
SCU_WUTCR	0x002C	RW	Wake-up Timer Control Register	0x0000_0000	4.9.10
SCU_WUTDR	0x0030	RW	Wake-up Timer Data Register	0x0000_00FA	4.9.11
SCU_HSITRM	0x00A8	RW	HSI RC Trim Register	0x0000_0xxx	4.9.12
SCU_LSITRM	0x00AC	RW	LSI RC Trim Register	0x0000_00xx	4.9.13

NOTE:

1. The CHIPID is written by hardware if the proper configure address is read.

The base address and register map of the SCU (clock generation) are described in the following:

Table 12. Base Address of SCU (Clock Generation)

Name	Base Address
SCU (clock generation)	0x4000_1800

Table 13. SCU Register Map (Clock Generation)

Name	Offset	Type	Description	Reset Value	Ref.
SCU_SCCR	0x0000	RW	System Clock Control Register	0x0000_0000	4.9.14
SCU_CLKSRCR	0x0004	RW	Clock Source Control Register	0x0000_000C	4.9.15
SCU_SCDIVR1	0x0008	RW	System Clock Divide Register 1	0x0000_0000	4.9.16
SCU_SCDIVR2	0x000C	RW	System Clock Divide Register 2	0x0000_0000	4.9.17
SCU_CLKOCR	0x0010	RW	Clock Output Control Register	0x0000_0000	4.9.18
SCU_CMONCR	0x0014	RW	Clock Monitoring Control Register	0x0000_0000	4.9.19
SCU_PPCLKEN1	0x0020	RW	Peripheral Clock Enable Register 1	0x0000_0000	4.9.20
SCU_PPCLKEN2	0x0024	RW	Peripheral Clock Enable Register 2	0x0002_0000	4.9.21
SCU_PPCLKSR	0x0040	RW	Peripheral Clock Selection Register	0x0010_0000	4.9.22
SCU_PPRST1	0x0060	RW	Peripheral Reset Register 1	0x0000_0000	4.9.23
SCU_PPRST2	0x0064	RW	Peripheral Reset Register 2	0x0000_0000	4.9.24
SCU_HSECR	0x0080	RW	High-Speed External Clock Control Register	0x0000_0000	4.9.25
SCU_LSECR	0x0084	RW	Low-Speed External Clock Control Register	0x0000_0010	4.9.26
SCU_PLLCON	0x0090	RW	PLL Control Register	0x0000_0000	4.9.27
SCU_VUSB33CR	0x00A0	RW	VUSB33 Control Register	0x0000_0000	4.9.28
SCU_ECCIESR	0x00B0	RW	Flash ECC Interrupt Enable and Status Register	0x0000_0000	4.9.29

The base address and register map of the SCU (LVR/LVI) are described in the following:

Table 14. Base Address of SCU (LVR/LVI)

Name	Base Address
SCU (LVR/LVI)	0x4000_5100

Table 15. SCU Register Map (LVR/LVI)

Name	Offset	Type	Description	Reset Value	Ref.
SCU_LVICR	0x0000	RW	Low-Voltage Indicator Control Register	0x0000_0000	4.9.30
SCU_ LVRCR	0x0004	RW	Low-Voltage Reset Control Register	0x0000_0000	4.9.31

4.9.1 SCU_VENDORID: Vendor ID Register

The SCU_VENDORID register shows the vendor identification information. This is a 32-bit read-only register.

SCU_VENDORID=0x4000_F000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VENDID[31:0]																															
0x4142_4F56																															
R0																															

31 0	VENDID[31:0]	Vendor Identification bits. 0x4142_4F56

4.9.2 SCU_CHIPID: Chip ID Register

The SCU_CHIPID register shows chip identification information. This is a 32-bit read-only register.

SCU_CHIPID=0x4000_F004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHIPID[31:0]																															
0x4731_3360																															
R0																															

31 0	CHIPID[31:0]	Chip Identification bits.	
		0x4731_3360	256 KB Flash memory for program

4.9.3 SCU_REVNR: Revision Number Register

The SCU_REVNR is a 32-bit read-only register. This register is available at 32-/ 16-/ 8-bit access.

SCU_REVNR=0x4000_F008

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			REVNO[7:0]
0x000000			xx
-			RO

7 0	REVNO[7:0]	Chip Revision Number. These bits are fixed by manufacturer.

4.9.4 SCU_PMREMAP: Program Memory Remap Register

The SCU_PMREMAP is a 32-bit write-only register.

SCU_PMREMAP=0x4000_F014

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
WTIDKY[15:0]		nPMREM[7:0]	PMREM[7:0]
0x0000		0x00	0x00
WO		WO	RW

31 16	WTIDKY[15:0]	Write Identification Key When writing, write 0xE2F1 to these bits, or else writing is ignored.
15 8	nPMREM[7:0]	Write Complement Key When writing, write the complement value of PMREM[7:0], or else writing is ignored.
7 0	PMREM[7:0]	Program Memory Remap 0x69 Boot ROM is re-mapped to address 0x00000000. The address 0x10001000 of Flash memory is re-mapped to the address 0x00001000. Others Flash memory is re-mapped to address 0x00000000.

NOTE:

1. The remapped program memory can be accessed from the original address.

4.9.5 SCU_BTPSCR: Boot Pin Status and Control Register

The SCU_BTPSCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_BTPSCR=0x4000_F018

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4 3 2 1	0
Reserved			BFIND[1:0]	Reserved	BTPSTA
-			x	-	x
.			RW	.	RO

6 5	BFIND[1:0]	BOOT Pin Function Indicator bits. The BFIND[1:0] bits are cleared to “00” by POR, the BFIND[1] bit is cleared to ‘0’ by nRESET, and the bits are not cleared by other system reset. One of the two of the following must be set in the BFIND[1:0] bits to check whether ISP is needed or not.
		10 Check the BOOT pin when a system reset occurs by nRESET including POR.
		11 Check the BOOT pin when a system reset occurs only by POR.
0	BTPSTA	BOOT Pin Status bit.
		0 The BOOT pin is low level.
		1 The BOOT pin is high level.
		NOTE: 1. This bit is always ‘1’ if the BOOT pin is not selected for alternate function.

NOTE:

1. When a system reset occurs, the PD5 pin is configured as an alternate function for BOOT, the pull-up resistor is enabled, and the debounce filter is enabled.

4.9.6 SCU_RSTSSR: Reset Source Status Register

The SCU_RSTSSR register shows reset source information when reset event occurred. '1' implies a reset event exists, while '0' means a reset event does not exist for a corresponding reset source.

When a reset source is detected, '1' is written into the corresponding bit position and reset status will be cleared.

The SCU_RSTSSR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_RSTSSR=0x4000_F01C

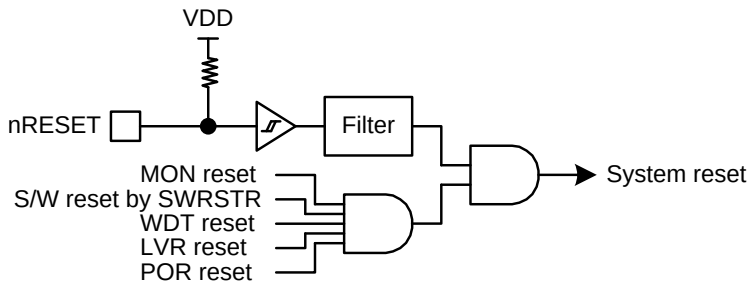
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5	4	3	2	1	0
Reserved				MONSTA	SWSTA	EXTSTA	WDTSTA	LVRSTA	PORSTA
-				x	x	x	x	x	x
.				RW	RW	RW	RW	RW	RW

5	MONSTA	Clock Monitoring Reset Status
	0	Not detected
	1	Clock monitoring reset is detected. The bit is cleared to '0' when '1' is written.
4	SWSTA	Software Reset Status.
	0	Not detected
	1	Software reset is detected. The bit is cleared to '0' when '1' is written.
3	EXTSTA	External Pin Reset Status.
	0	Not detected
	1	External pin reset is detected. The bit is cleared to '0' when '1' is written.
2	WDTSTA	Watchdog Timer Reset Status.
	0	Not detected
	1	Watchdog Timer reset is detected. The bit is cleared to '0' when '1' is written.
1	LVRSTA	LVR Reset Status.
	0	Not detected
	1	LVR reset is detected. The bit is cleared to '0' when '1' is written.
0	PORSTA	POR Reset Status.
	0	Not detected
	1	POR reset is detected. The bit is cleared to '0' when '1' is written.

NOTES:

1. The PORSTA bit is set to '1' and the other bits are cleared to '0' when power-on reset occurs.
2. The corresponding reset status bit may be set to '1' if any reset signal is asserted during power-on reset occurs. For example, The EXTSTA bit may be set if the external reset is asserted during POR.

Figure 19. Reset Circuit Diagram



NOTE :

1. A core software reset will reset only CPU.

4.9.7 SCU_NMISRCR: NMI Source Selection Register

The SCU_NMISRCR is the non-maskable interrupt configuration register, which can be set by software. The SCU_NMISRCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_NMISRCR=0x4000_F020

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4 3 2 1 0
Reserved			NMICON	MONINT	ECCDINT	NMISRC[4:0]
-			0	0	1	0
.			RW	RW	RW	RW

7	NMICON	Non-Maskable Interrupt (NMI) Control.	
		0	Disable NMI
		1	Enable NMI
6	MONINT	Clock Monitoring Interrupt Selection.	
		0	Non-select clock monitoring interrupt for NMI source
		1	Select clock monitoring interrupt for NMI source
5	ECCDINT	ECC Detection Interrupt Selection	
		0	Non-select ECC detection interrupt for NMI source
		1	Select ECC detection interrupt for NMI source
4 0	NMISRC[4:0]	Non-Maskable Interrupt Source Selection.	
		Select one of the interrupt sources 0 to 31 for NMI source.	

NOTE:

1. The interrupt source which is selected for NMI should be disabled in NVIC to avoid both generation of the normal and NMI interrupts.

4.9.8 SCU_SWRSTR: Software Reset Register

The SCU_SWRSTR register is 32-bit size.

SCU_SWRSTR=0x4000_F024

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
WTIDKY[15:0]		Reserved	SWRST[7:0]
0x0000		-	0x00
WO		-	WO

31 16	WTIDKY[15:0]	Write Identification Key When writing, write 0x9EB3 to these bits, or else writing is ignored.
7 0	SWRST[7:0]	Software Reset (System Reset) 0x2D A software reset will be generated for all peripheral and core. Others No effect

4.9.9 SCU_SRSTVR: System Reset Validation Register

The SCU_SRSTVR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_SRSTVR=0x4000_F028

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			VALID[7:0]
-			0x55
-			RO

7 0	VALID[7:0]	System Reset Validation 0x55 System reset is O.K. Others A weak system reset. A system reset must be generated by software.

4.9.10 SCU_WUTCR: Wake-up Timer Control Register

Wake-up timer always works on operating mode. This timer gives a stable time for clock generation during Power on and DEEP SLEEP mode release. The main purpose of this timer is periodical tick timer or a wake-up source.

The SCU_WUTCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_WUTCR=0x4000_F02C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4 3 2	1	0
Reserved			WUTIEN	Reserved	CNTRLD	WUTIFLAG
-			0	-	0	0
-			RW	-	RW	RW

7	WUTIEN	Wake-up Timer Interrupt Enable bit
	0	Disable wake-up timer interrupt
	1	Enable wake-up timer interrupt
1	CNTRLD	Counter Reload bit
	0	No effect
	1	Reload data to counter (Automatically cleared to ‘0’ after operation)
0	WUTIFLAG	Wake-up Timer Interrupt Flag bit
	0	No request occurred
	1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.

NOTE:

1. This bit may not be set to "1b" if the PCLK frequency is slower than the HCLK frequency. So, in order for WUT interrupt to occur normally, the SCU_SCDIVR2.PDIV[1:0] bits must be set to "00b" so that the PCLK frequency is the same as the HCLK frequency.

4.9.11 SCU_WUTDR: Wake-up Timer Data Register

The SCU_WUTDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_WUTDR=0x4000_F030

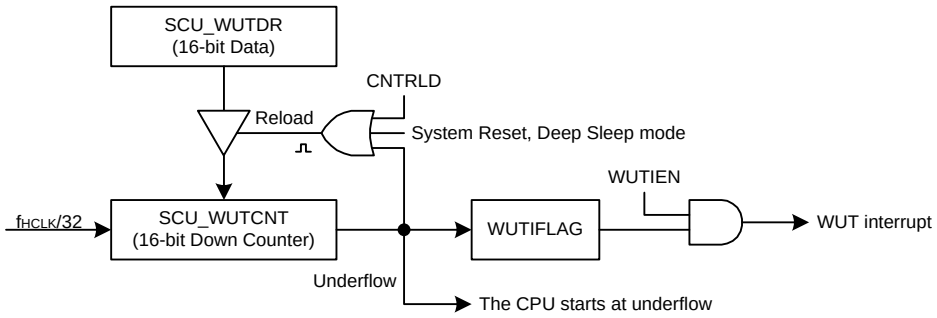
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		WUTDATA[15:0]	
-		0x00FA	
-		RW	

15 0	WUTDATA[15:0]	Wake-up Timer Data. The range is 0x0001 to 0xFFFF.

NOTES:

1. When HSI is system clock, its value should be set to be at least more than 50us.
2. When LSI is system clock, its value should be set to be at least more than 500us

Figure 20. Wake-Up Timer Circuit Diagram



NOTE:

1. The WUTIFLAG bit may not be set to "1b" if the PCLK frequency is slower than the HCLK frequency. So, for WUT interrupt to occur normally, the SCU_SCDIVR2.PDIV[1:0] bits must be set to "00" so that the PCLK frequency is the same as the HCLK frequency.

4.9.12 SCU_HSITRM: High-Speed Internal RC Trim Register

The SCU_HSITRM register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_HSITRM=0x4000_F0A8

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
WTIDKY[15:0]		Reserved	CTRMH[3:0]	FTRMH[6:0]
0x0000		-	x	x
WO		-	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key. When writing, write 0xA6B5 to these bits, or else writing is ignored.
10 7	CTRMH[3:0]	Factory HSI Coarse Trim bits. These bits are fixed by manufacturer and read from “Configure Option Page 0” when a system reset occurs. These bits provide a user-programmable trimming value on operation. The range is -8 to +7, the CTRMH[3] is sign bit, and the frequency is changed by 1.38 MHz step-by-step.
6 0	FTRMH[6:0]	Factory HSI Fine Trim bits. These bits are fixed by manufacturer and read from “Configure Option Page 0” when a system reset occurs. These bits provide a user-programmable trimming value on operation. The range is -64 to +63, the FTRMH[6] is sign bit, and the frequency is changed by 7.4 kHz step-by-step.

4.9.13 SCU_LSITRM: Low-Speed Internal RC Trim Register

The SCU_LSITRM register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_LSITRM=0x4000_F0AC

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2 1 0
WTIDKY[15:0]		nTRMW[7:0]	CTRMW[3:0]	Reserved	FTRMW[2:0]
0x0000		x	x	-	x
WO		WO	RW	-	RW

31 16	WTIDKY[15:0]	Write Identification Key. When writing, write 0x4C3D to these bits, or else writing is ignored.
15 8	nTRMW[7:0]	Write Complement Key On writes, write the complement value of TRMW[7:0], otherwise the write is ignored.
7 4	CTRMW[3:0]	Factory LSI Coarse Trim bits. These bits are fixed by manufacturer and read from “Configure Option Page 0” when a system reset occurs. These bits provide a user-programmable trimming value on operation. The range is -8 to +7, the CTRMW[3] is sign bit, and the frequency is changed by 3.3 kHz step-by-step.
2 0	FTRMW[2:0]	Factory LSI Fine Trim bits. These bits are fixed by manufacturer and read from “Configure Option Page 0” when a system reset occurs. These bits provide a user-programmable trimming value on operation. The range is -4 to +3, the FTRMW[2] is sign bit, and the frequency is changed by 1.2 kHz step-by-step.

4.9.14 SCU_SCCR: System Clock Control Register

The A31G336 has multiple clock sources to generate internal operating clocks. SCU_SCCR register controls such a clock source. This register is 32-bit size.

SCU_SCCR=0x4000_1800

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4 3	2 1 0
WTIDKY[15:0]		Reserved	PLLINCLKSEL	Reserved	MCLKSEL[2:0]
0x0000		-	0	-	0
WO		-	RW	-	RW

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0x570A to these bits, or else writing is ignored.	
7	PLLINCLKSEL	PLL Input Clock Source Selection Register	
		0	The HSI clock selected by CLKSRCR.HSISEL[1:0] is used as PLL input clock
		1	A HSE clock is used as PLL input clock
2 0	MCLKSEL[2:0]	Main Clock Selection bits, MCLK	
		000	High-speed Internal RC oscillator (32 MHz), HSI
		001	External main oscillator (2 – 48 MHz), HSE
		010	External sub oscillator (32.768 kHz), LSE
		011	Low-speed Internal RC oscillator (40 kHz), LSI
		100	PLL output clock

NOTES:

1. The MCLKSEL bits won't be changed on selecting the clock which is disabled by SCU_CLKSRCR register.
2. If the MCLKSEL bits are "010" or "011", the HDIV[2:0] bits of SCU_SCDIVR1 register should be "100" for non-divided system clock.

4.9.15 SCU_CLKSRCR: Clock Source Control Register

The A31G336 has multiple clock sources to generate internal operating clocks. SCU_CLKSRCR register controls each clock source. This register is 32-bit size.

SCU_CLKSRCR=0x4000_1804

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14	13 12	11 10 9	8	7 6 5	4	3	2	1	0
WTIDKY[15:0]		Reserved	HSISEL[1:0]	Reserved	XMFRNG	Reserved	PLLEN	LSIEN	HSIEN	HSEEN	LSEEN
0x0000		-	0	-	0	-	0	1	1	0	0
WO		-	RW	-	RW	-	RW	RW	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key
		When writing, write 0xA507 to these bits, or else writing is ignored.
13 12	HSISEL[1:0]	HSI Frequency Selection bits
		00 32 MHz HSI
		01 16 MHz HSI
		10 8 MHz HSI
		11 4 MHz HSI
8	XMFRNG	Main Oscillator Type and Frequency Range Selection bit
		0 X-tal for HSE, 2 to 16 MHz
		1 External clock for HSE, 2 MHz to 48 MHz
4	PLLEN	PLL Enable bit
		0 Disable PLL
		1 Enable PLL
		NOTES: 1. This bit should be set to '1' after setting the PLLCON register. 2. Before entering deep sleep mode, this bit should be '0' after the system clock is changed from the PLL clock to another clock source.
3	LSIEN	LSI Enable bit, Low-speed internal RC oscillator
		0 Disable LSI
		1 Enable LSI
2	HSIEN	HSI Enable bit, High-speed internal RC oscillator
		0 Disable HSI
		1 Enable HSI
1	HSEEN	HSE Enable bit, External main oscillator
		0 Disable HSE
		1 Enable HSE
0	LSEEN	LSE Enable bit, External sub oscillator. The bit is cleared to '0' by the reset of POR but retained by the other reset.
		0 Disable LSE
		1 Enable LSE

4.9.16 SCU_SCDIVR1: System Clock Divide Register 1

The SCU_SCDIVR1 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_SCDIVR1=0x4000_1808

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2 1 0
Reserved			WLDIV[2:0]	Reserved	HDIV[2:0]
-			0	-	0
-			RW	-	RW

6 4	WLDIV[2:0]	Clock Divide bits for RTCC, Divider 2	
		000	MCLK÷64
		001	MCLK÷128
		010	MCLK÷256
		011	MCLK÷512
		100	MCLK÷1024
		others	Reserved
2 0	HDIV[2:0]	Clock Divide bits for HCLK, Divider 0	
		000	MCLK÷16
		001	MCLK÷8
		010	MCLK÷4
		011	MCLK÷2
		100	MCLK÷1
		others	Reserved (MCLK÷1)

NOTES:

1. If the selected MCLK is LSE or LSI, the HDIV[2:0] bits should be set to “100”.
2. The frequency range of HCLK should be 2.0 to 32[MHz] by software while the HSI is the system clock.
3. After changing the value of HDIV[2:0] bits for system clock speed, a delay of 10 μs is required before changing the value again immediately.

4.9.17 SCU_SCDIVR2: System Clock Divide Register 2

The SCU_SCDIVR2 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_SCDIVR2=0x4000_180C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4	3 2	1 0
Reserved				SYSTDIV[1:0]	Reserved	PDIV[1:0]
-				0	-	0
-				RW	-	RW

5 4	SYSTDIV[1:0]	Clock Divide bits for SysTick Timer, Divider 3	
		00	HCLK÷1
		01	HCLK÷2
		10	HCLK÷4
		11	HCLK÷8
1 0	PDIV[1:0]	Clock Divide bits for PCLK, Divider 1	
		00	HCLK÷1
		01	HCLK÷2
		10	HCLK÷4
		11	HCLK÷8

NOTE:

1. If the selected MCLK is LSE or LSI, the PDIV[1:0] should be set to “00”.

4.9.18 SCU_CLKOCR: Clock Output Control Register

The A31G336 can drive the clock from a selected clock (CLKOS) with a dedicated post divider.

The SCU_CLKOCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_CLKOCR=0x4000_1810

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5 4 3	2 1 0
Reserved			CLKOEN	POLSEL	CLKODIV[2:0]	CLKOS[2:0]
-			0	0	0	0
-			RW	RW	RW	RW

7	CLKOEN	Clock Output Enable bit	
		0	Disable clock output
		1	Enable clock output
6	POLSEL	Clock Output Polarity Selection bit when disable	
		0	Low level during disable
		1	High level during disable
5 3	CLKODIV[2:0]	Output Clock Divide bits, Divider 4	
		000	“Selected clock”÷1
		001	“Selected clock”÷2
		010	“Selected clock”÷4
		011	“Selected clock”÷8
		100	“Selected clock”÷16
		101	“Selected clock”÷32
		110	“Selected clock”÷64
		111	“Selected clock”÷128
2 0	CLKOS[2:0]	Clock Output Selection bits	
		000	MCLK
		001	LSI
		010	HSI
		011	HCLK
		100	PCLK
		others	Reserved (None)

4.9.19 SCU_CMONCR: Clock Monitoring Control Register

Internal clock can be monitored by using internal LSI for security purposes.

The SCU_CMONCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_CMONCR=0x4000_1814

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5	4	3	2	1 0
Reserved			MONEN	MACTS[1:0]	Reserved	MONFLAG	NMINTFG	MONCS[1:0]
-			0	0	-	0	0	0
-			RW	RW	-	RW	RW	RW

7	MONEN	Clock Monitoring Enable bit
	0	Disable clock monitoring
	1	Enable clock monitoring

NOTE:

1. When this bit is reset to ‘0’, the block clears the 4/7-bit counter, inFLAG, and flags.

6 5	MACTS[1:0]	Clock Monitoring Action Selection bits
	00	No action by clock monitoring, but flags will be set/cleared on condition
	01	Reset generation by clock monitoring
	10	The system clock will be changed to the LSI regardless of MCLKSEL[1:0] bits of system clock control register (SCU_SCCR) only when the MCLK is selected for monitoring.
	11	Not used
3	MONFLAG	Clock Monitoring Result Flag bit
	0	The clock under monitoring is not ready.
	1	The clock under monitoring is ready. This bit is cleared to ‘0’ when ‘1’ is written.
2	NMINTFG	Clock Monitoring Interrupt Flag bit (only when the MCLK is selected for monitoring)
	0	No request occurred
	1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.

NOTE:

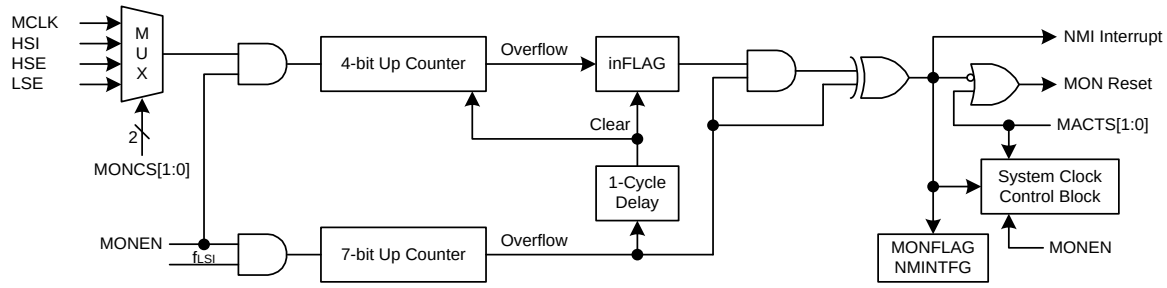
1. When the bit is set, the system clock must be switched to LSI by software.

1 0	MONCS[1:0]	Monitored Clock Selection bits
	00	MCLK
	01	HSI
	10	HSE
	11	LSE

NOTES:

1. The block should be enabled after disable to clear the internal status for new clock monitoring.
2. This block must be disabled by software before entering DEEP SLEEP mode.
3. When the “clock monitoring” function is disabled by software, the following sequence is required.
   - First, the MACTS bits should be cleared to “00”.
   - Second, the MONEN bit must be cleared to ‘0’.

Figure 21. Clock Monitoring Circuit Diagram



NOTE:

1. LSI should be enabled to use this block since it is the watch clock for monitoring.

4.9.20 SCU_PPCLKEN1: Peripheral Clock Enable Register 1

To use a certain peripheral unit, its clock should be activated by writing '1' to the corresponding bit in SCU_PPCLKEN1/SCU_PPCLKEN2 register. Until enabling the clock, the peripheral does not operate properly. To stop the clock of the peripheral unit, write '0' to the corresponding bit in the SCU_PPCLKEN1/PPCLKEN2 register.

The SCU_PPCLKEN1 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPCLKEN1=0x4000_1820

31 30 29 28 27 26 25 24	23 22	21	20	19	18	17	16	15	14	13 12 11	10	9	8	7 6	5	4	3	2	1	0
Reserved		T21CLKE	T20CLKE	Reserved	T12CLKE	T11CLKE	T10CLKE	Reserved	TSCLKE	Reserved	T15CLKE	T14CLKE	T13CLKE	Reserved	PFCLKE	Reserved	PDCLKE	PCCLKE	PBCLKE	PACLKE
-		0	0	-	0	0	0	-	0	-	0	0	0	-	0	-	0	0	0	0
-		RW	RW	-	RW	RW	RW	-	RW	-	RW	RW	RW	-	RW	-	RW	RW	RW	RW

21	T21CLKE	TIMER21 clock enable
20	T20CLKE	TIMER20 clock enable
18	T12CLKE	TIMER12 clock enable
17	T11CLKE	TIMER11 clock enable
16	T10CLKE	TIMER10 clock enable
14	TSCLKE	Temperature Sensor clock enable
10	T15CLKE	TIMER15 clock enable
9	T14CLKE	TIMER14 clock enable
8	T13CLKE	TIMER13 clock enable
5	PFCLKE	Port F clock enable
3	PDCLKE	Port D clock enable
2	PCCLKE	Port C clock enable
1	PBCLKE	Port B clock enable
0	PACLKE	Port A clock enable

NOTE:

1. The peripheral registers may not be read/written by software when the peripheral clock is disabled.

4.9.21 SCU_PPCLKEN2: Peripheral Clock Enable Register 2

The SCU_PPCLKEN2 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPCLKEN2=0x4000_1824

31	30	29	28	27	26 25 24	23 22	21	20	19	18	17	16	15 14 13	12	11	10	9	8	7	6	5	4	3 2	1	0
DMACLKE	Reserved	USBCLKE	CMP1CLKE	CMP0CLKE	Reserved		SPI1CLKE	SPI0CLKE	FMCLKE	LVICLKE	WDTCLKE		Reserved	CRCLKE	RTCCLKE	ADCLKE	Reserved	I2C2CLKE	I2C1CLKE	I2C0CLKE	UST13CLKE	UST12CLKE	Reserved	UST11CLKE	UST10CLKE
0	-	0	0	0	-		0	0	0	0	1		-	0	0	0	-	0	0	0	0	0	-	0	0
RW	-	RW	RW	RW	-		RW	RW	RW	RW	RW		-	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW	RW

31	DMACLKE	DMA clock enable
29	USBCLKE	USB clock enable
28	CMP1CLKE	Comparator 1 clock enable
27	CMP0CLKE	Comparator 0 clock enable
21	SPI1CLKE	SPI 1 clock enable
20	SPI0CLKE	SPI 0 clock enable
19	FMCLKE	Flash Memory Control clock enable. Ignored during Flash operation.
18	LVICLKE	LVI (Low-Voltage Indicator) clock enable
17	WDTCLKE	WDT (Watchdog Timer) clock enable. The LSI won't be disabled if the clock is enabled by watchdog timer configuration register (CONF_WDTCNFIG) in "Configure Option Page 1"
12	CRCLKE	CRC (Cyclic Redundancy Check) clock enable
11	RTCCLKE	Real Time Clock/Calendar clock enable. The bit is cleared to '0' by the reset of POR but retained by the other reset.
10	ADCLKE	ADC (Analog to Digital Converter) clock enable
8	I2C2CLKE	I2C2 (Inter-integrated Circuit) clock enable
7	I2C1CLKE	I2C1 (Inter-integrated Circuit) clock enable
6	I2C0CLKE	I2C0 (Inter-integrated Circuit) clock enable
5	UST13CLKE	USART13 clock enable
4	UST12CLKE	USART12 clock enable
1	UST11CLKE	USART11 clock enable
0	UST10CLKE	USART10 clock enable

NOTE:

1. The peripheral registers may not be read/written by software when the peripheral clock is disabled.

4.9.22 SCU_PPCLKSR: Peripheral Clock Selection Register

The SCU_PPCLKSR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPCLKSR=0x4000_1840

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	T20CLK, Reserved, USBCLK, Reserved, RTCCLK[1:0]		Reserved, WDTCLK
-	1 - 0 - 0		- 0
.	RW . RW . RW		. RW

20	T20CLK	Timer 20 Clock Selection.
	0	LSE clock
	1	PCLK clock
14	USBCLK	USB Clock Selection
	0	HSE
	1	PLL Output
9 8	RTCCLK[1:0]	Real Time Clock/Calendar Clock Selection. The bit is cleared to '0' by the reset of POR but retained by the other reset.
	00	Low level (RTC stuck)
	01	LSE clock
	10	LSI clock
	11	A clock of the MCLK which is divided by two.
0	WDTCLK	Watch-dog Timer Clock Selection.
	0	LSI clock
	1	PCLK clock

4.9.23 SCU_PPRST1: Peripheral Reset Register 1

SCU_PPRST1/PPRST2 register can make a peripheral reset. If a specific bit in this register is set to '1', the peripheral corresponds with this bit occurs a reset event and the registers of the peripheral are initialized with reset values.

The SCU_PPRST1 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPRST1=0x4000_1860

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		PLLRST	Reserved							T21RST	T20RST	Reserved	T12RST	T11RST	T10RST	Reserved	TSRST	Reserved			T15RST	T14RST	T13RST	Reserved		PFRST	Reserved	PDRST	PCRST	PBRST	PARST
-		0	-							0	0	-	0	0	0	-	0	-			0	0	0	-		0	0	0	0	0	0
-		RW	-							RW	RW	-	RW	RW	RW	-	RW	-			RW	RW	RW	-		RW	RW	RW	RW	RW	RW

29	PLLRST	PLL Reset bit
		0 No effect
		1 Reset PLL, Cleared by software
21	T21RST	Timer 21 Reset bit
		0 No effect
		1 Reset Timer 21, Cleared by software
20	T20RST	Timer 20 Reset bit
		0 No effect
		1 Reset Timer 20, Cleared by software
18	T12RST	Timer 12 Reset bit
		0 No effect
		1 Reset Timer 12, Cleared by software
17	T11RST	Timer 11 Reset bit
		0 No effect
		1 Reset Timer 11, Cleared by software
16	T10RST	Timer 10 Reset bit
		0 No effect
		1 Reset Timer 10, Cleared by software
14	TSRST	Temperature Sensor Reset bit
		0 No effect
		1 Reset temperature sensor, Cleared by software
10	T15RST	Timer 15 Reset bit
		0 No effect
		1 Reset Timer 15, Cleared by software
9	T14RST	Timer 14 Reset bit
		0 No effect
		1 Reset Timer 14, Cleared by software
8	T13RST	Timer 13 Reset bit
		0 No effect
		1 Reset Timer 13, Cleared by software
5	PFRST	Port F Reset bit
		0 No effect
		1 Reset Port F, Cleared by software

3	PDRST	Port D Reset bit	
		0	No effect
		1	Reset Port D, Cleared by software
2	PCRST	Port C Reset bit	
		0	No effect
		1	Reset Port C, Cleared by software
1	PBRST	Port B Reset bit	
		0	No effect
		1	Reset Port B, Cleared by software
0	PARST	Port A Reset bit	
		0	No effect
		1	Reset Port A, Cleared by software

4.9.24 SCU_PPRST2: Peripheral Reset Register 2

The SCU_PPRST2 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_PPRST2=0x4000_1864

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMARST	Reserved	USBRST	CMP1RST	CMP0RST	Reserved					SPI1RST	SPI0RST	FMCRST	LVIRST	Reserved					CRRST	RTCRST	ADRST	Reserved	I2C2RST	I2C1RST	I2C0RST	UST13RST	UST12RST	Reserved		UST11RST	UST10RST
0	-	0	0	0	-					0	0	0	0	-					0	0	0	-	0	0	0	0	0	-		0	0
RW	-	RW	RW	RW	-					RW	RW	RW	RW	-					RW	RW	RW	-	RW	RW	RW	RW	RW	-		RW	RW

31	DMARST	DMA Reset bit
		0 No effect
		1 Reset DMA, cleared by software
29	USBRST	USB Reset bit
		0 No effect
		1 Reset USB, cleared by software
28	CMP1RST	Comparator 1 Reset bit
		0 No effect
		1 Reset Comparator 1, cleared by software
27	CMP0RST	Comparator 0 Reset bit
		0 No effect
		1 Reset Comparator 0, cleared by software
21	SPI1RST	SPI1 Reset bit
		0 No effect
		1 Reset SPI1, cleared by software
20	SPI0RST	SPI0 Reset bit
		0 No effect
		1 Reset SPI0, cleared by software
19	FMCRST	FMC (Flash Memory Control) Reset bit. Ignored during Flash operation.
		0 No effect
		1 Reset Flash memory control, cleared by software.
18	LVIRST	LVI (Low-Voltage Indicator) Reset bit
		0 No effect
		1 Reset LVI, cleared by software
12	CRRST	CRC (Cyclic Redundancy check) Reset bit
		0 No effect
		1 Reset CRC, cleared by software
11	RTCRST	RTCC (Real Time Clock and Calendar) Reset bit
		0 No effect
		1 Reset RTCC, cleared by software
10	ADRST	ADC Reset bit
		0 No effect
		1 Reset ADC, cleared by software
8	I2C2RST	I2C2 Reset bit
		0 No effect
		1 Reset I2C2, cleared by software
7	I2C1RST	I2C1 Reset bit
		0 No effect

		1	Reset I2C1, cleared by software
6	I2C0RST	I2C0 Reset bit	
		0	No effect
		1	Reset I2C0, cleared by software
5	UST13RST	USART13 Reset bit	
		0	No effect
		1	Reset USART13 , Cleared by software
4	UST12RST	USART12 Reset bit	
		0	No effect
		1	Reset USART12, cleared by software
1	UST11RST	USART11 Reset bit	
		0	No effect
		1	Reset USART11, cleared by software
0	UST10RST	USART10 Reset bit	
		0	No effect
		1	Reset USART10, cleared by software

4.9.25 SCU_HSECR: External Main Oscillator Control Register

The SCU_HSECR register is 32-bit size. This register is 32-bit size.

SCU_HSECR=0x4000_1880

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4	3 2	1	0
WTIDKY[15:0]		Reserved		HSEISEL[1:0]	HSEFLTSEL[1:0]	Reserved	HSEFLTDIS
0x0000		-		0	0	-	0
WO		-		RW	RW	-	RW

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0x9B37 to these bits, or else writing is ignored	
5 4	HSEISEL [1:0]	External Main Oscillator Current Selection	
		00	High current for HSE
		01	Mid-high current for HSE
		10	Mid-low current for HSE
		11	Low current for HSE
3 2	HSEFLTSEL[1:0]	External Main Oscillator Filter Selection	
		00	12 MHz < HSE ≤ 16 MHz
		01	8 MHz < HSE ≤ 12 MHz
		10	4 MHz < HSE ≤ 8 MHz
		11	2 MHz ≤ HSE ≤ 4 MHz
0	HSEFLTDIS	External Main Oscillator Filter Enable	
		0	Enable noise filter for HSE
		1	Disable noise filter for HSE

4.9.26 SCU_LSECR: External Sub Oscillator Control Register

The SCU_LSECR register is 32-bit size. This register is 32-bit size.

SCU_LSECR=0x4000_1884

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4	3 2 1	0
WTIDKY[15:0]		Reserved		HSEISEL[1:0]	Reserved	LSESCHEN
0x0000		-		1	-	0
WO		-		RW	-	RW

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0x40C3 to these bits, or else writing is ignored	
5 4	LSEISEL [1:0]	External Sub Oscillator Current Selection	
		00	High current for LSE
		01	Mid-high current for LSE
		10	Mid-low current for LSE
		11	Low current for LSE
0	LSESCHEN	External Sub Oscillator Schmitt Trigger Enable	
		0	Disable Schmitt trigger for LSE
		1	Enable Schmitt trigger for LSE

4.9.27 SCU_PLLCON: PLL Clock Control Register

The SCU_PLLCON register is 32-bit size. This register is 32-bit size.

SCU_PLLCON=0x4000_1890

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLLOCK	Reserved											PREDIV[3:0]				FBDIV[7:0]								POSTDIV1[3:0]				POSTDIV2[3:0]			
0	-											0				0								0				0			
RO	-											RW				RW								RW				RW			

31	PLLLOCK	PLL Lock Status
		0 PLL is not locked
		1 PLL is locked
		NOTE: 1. After this bit is set, a 200 µs delay is required to change the system clock to the PLL output clock.
19 16	PREDIV[3:0]	PLL Input Clock Pre-divider bits (R). The range is 0 to 15. $f_{PLLIN} = f_{PLLINCLK} \div (R + 1)$
15 8	FBDIV[7:0]	PLL Feedback divider bits (FB). The range is 0 to 255. $f_{VCO} = f_{PLLIN} \times (FB + 1)$
7 4	POSTDIV1[3:0]	PLL Post-1 divider bits (N1). The range is 0 to 15. $f_{DIV2} = f_{VCO} \div (N1 + 1)$
3 0	POSTDIV2[3:0]	PLL Post-2 divider bits (N2). The range is 0 to 15. $f_{PLL} = f_{DIV2} \div (N2 + 1) = f_{PLLIN} \times (FB + 1) \div \{(N2 + 1) \times (N1 + 1)\}$

4.9.28 SCU_VUSB33CR: VUSB33 Control Register

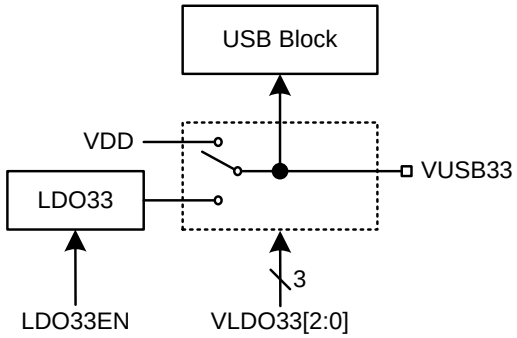
The SCU_VUSB33CR register is 32-bit size. This register is 32-bit size.

SCU_VUSB33CR=0x4000_18A0

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5	4	3	2 1 0
WTIDKY[15:0]		Reserved		LDO33EN	Reserved	VLDO33[2:0]
0x0000		-		0	-	0
WO		-		RW	-	RW

31 16	WTIDKY[15:0]	Write Identification Key	
		When writing, write 0xA369 to these bits, or else writing is ignored	
4	LDO33EN	LDO33 Enable bit. The LDO33 is auto disabled if the VLDO33[2:0] bits are not 0x1.	
		0	Disable LDO33
		1	Enable LDO33
2 0	VLDO33[2:0]	VUSB33 Pin Voltage Control	
		001	The VUSB33 pin is the output voltage of internal LDO.
		010	The VUSB33 pin is the same voltage as VDD.
		Other	An external 3.3 V should be connected to the VUSB33 pin for USB operation.

Figure 22. VUSB33 Pin Circuit Diagram



NOTES:

1. If the USB block is not used, the VUSB33 pin must be connected either internally or externally to VDD.
2. A 0.1 µF capacitor should be connected between the VUSB33 pin and ground.

4.9.29 SCU_ECCIESR: Flash ECC Interrupt Enable and Status Register

The SCU_ECCIESR register is 32-bit size. This register is 32-bit size.

SCU_ECCIESR=0x4000_18B0

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved | ECCDIFLAG | ECCCIFLAG | Reserved | ECCIEN	Reserved	ECCADDR[17:0]	
- 0 0 - 0	-	0	
- RW RW - RW	-	RO	

29	ECCDIFLAG	ECC Detection Interrupt flag. This bit is set when two ECC errors have been detected. When this bit is set, NMI interrupt will be generated if the ECCDINT bit of SCU_ NMISRCR register is set for NMI source.
	0	No request occurred
	1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’
28	ECCCIFLAG	ECC Correction Interrupt flag. This bit is set when one ECC error has been detected and corrected
	0	No request occurred
	1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
24	ECCIEN	ECC Correction Interrupt Enable
	0	Disable ECC correction interrupt
	1	Enable ECC correction interrupt
17 0	ECCADDR[17:0]	Flash ECC Error or Correction Address
		These bits hold the lower 18 bits of the corresponding Flash address when ECC detection or ECC correction is detected. If a new ECC error occurs while ECCDIFLAG or ECCIFLAG is set, this bitfield is not updated

4.9.30 SCU_LVICR: Low-Voltage Indicator Control Register

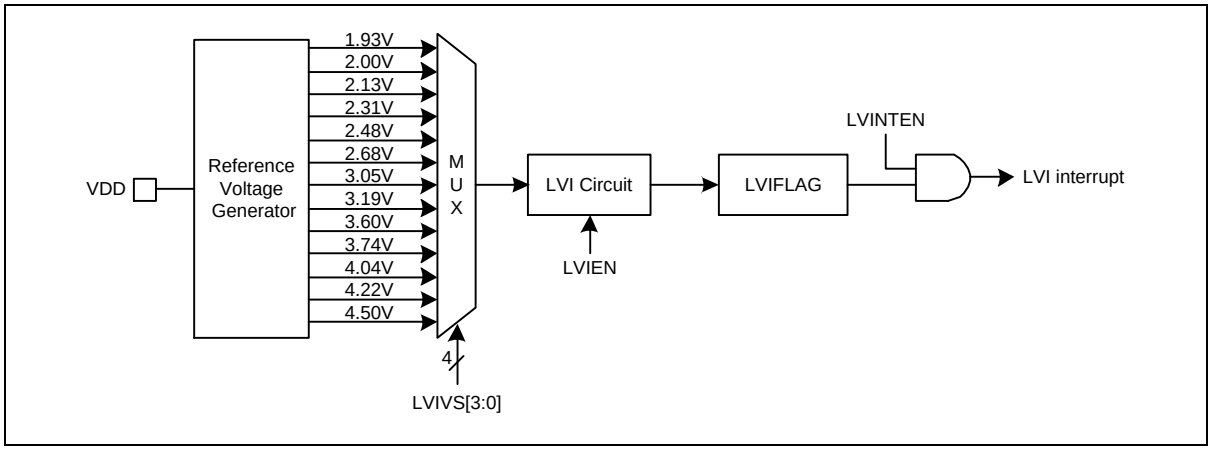
The SCU_LVICR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_LVICR=0x4000_5100

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3 2 1 0
Reserved			LVIEN	Reserved	LVINTEN	LVIFLAG	LVIVS[3:0]
-			0	-	0	0	0
-			RW	-	RW	RW	RW

7	LVIEN	LVI Enable.	
		0	Disable low-voltage indicator.
		1	Enable low-voltage indicator.
5	LVINTEN	LVI Interrupt Enable.	
		0	Disable low-voltage indicator interrupt.
		1	Enable low-voltage indicator interrupt.
4	LVIFLAG	LVI Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
3 0	LVIVS[3:0]	LVI Voltage Selection.	
		0011	1.93 V
		0100	2.00 V
		0101	2.13 V
		0110	2.31 V
		0111	2.48 V
		1000	2.68 V
		1001	3.05 V
		1010	3.19 V
		1011	3.60 V
		1100	3.74 V
		1101	4.04 V
		1110	4.22 V
		1111	4.50 V
		Other values	Not available

Figure 23. LVI Block Diagram



4.9.31 SCU_LVRCR: Low-Voltage Reset Control Register

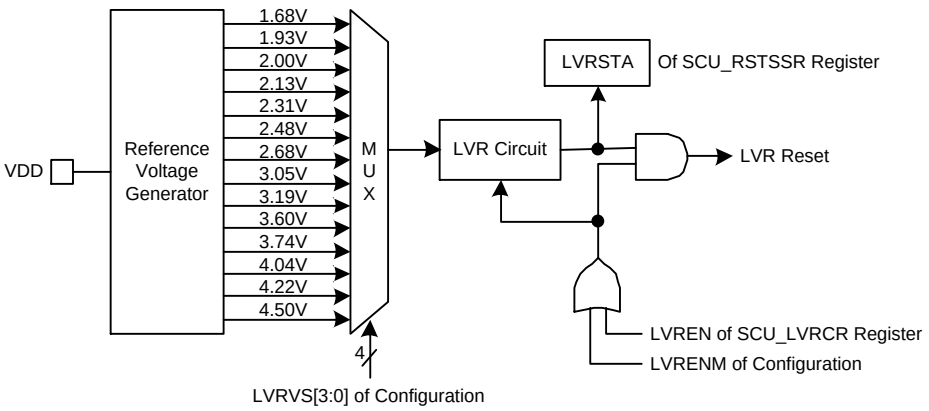
The SCU_LVRCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

SCU_LVRCR=0x4000_5104

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			LVREN[7:0]
-			0x00
-			RW

7 0	LVIEN[7:0]	LVR Enable. These bits are cleared to 0x00 by POR only and retained by other reset signals.	
		0x55	Disable low-voltage reset.
		Others	Enable low-voltage reset.

Figure 24. LVR Block Diagram



NOTE:

1. Voltage level of LVR is set by the low-voltage reset configuration register (CONF_LVRCNFIG) in the Configure Option Page 1.

4.9.32 SCU Register Map Summary

Table 16. SCU CHIPCONFIG Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SCU_ VENDORID	VENDID[31:0]																															
	Reset value	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	1	0	1	1	0
0x04	SCU_CHIPID	CHIPID[31:0]																															
	Reset value	0	1	0	0	0	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0
0x08	SCU_ REVNR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REVNO[7:0]							
	Reset value																									x	x	x	x	x	x	x	x
0x14	SCU_PMREMAP	WTIDKY[15:0]																nPMREM[7:0]								PMREM[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SCU_BTPSCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BFIND[1:0]		Res	Res	Res	Res	BTPSTA
	Reset value																										x	x					x
0x1C	SCU_RSTSSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MONSTA	SWSTA	EXTSTA	WDTSTA	LVRSTA	PORSTA
	Reset value																											x	x	x	x	x	x
0x20	SCU_NMISRCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NMICON	MONINT	ECCDINT	NMISRC[4:0]				
	Reset value																									0	0	1	0	0	0	0	0
0x24	SCU_SWRSTR	WTIDKY[15:0]																Res	Res	Res	Res	Res	Res	Res	Res	SWRST[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x28	SCU_SRSTVR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	VALID[7:0]							
	Reset value																									0	1	0	1	0	1	0	1
0x2C	SCU_WUTCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUTIEN	Res	Res	Res	Res	Res	CNTRLD	WUTIFLAG
	Reset value																									0						0	0
0x30	SCU_WUTDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0
0xA8	SCU_HSITRM	WTIDKY[15:0]																Res	Res	Res	Res	Res	CTRMH[3:0]			FTRMH[6:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						x	x	x	x	x	x	x	x	x	x	x
0xAC	SCU_LSITRM	WTIDKY[15:0]																nTRMW[7:0]								CTRMW[3:0]				Res	FTRMW[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x

Table 17. SCU Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SCU_SCCR	WTIDKY[15:0]																Res	Res	Res	Res	Res	Res	Res	Res	PLLINCLKSEL	Res	Res	Res	Res	MCLKSEL[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0					0	0	0
0x04	SCU_CLKSRCR	WTIDKY[15:0]																Res	Res	HSISEL[1:0]		Res	Res	Res	XMFRNG	Res	Res	Res	PLLEN	LSIEN	HSIEN	HSEEN	LSEEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0				0				0	1	1	0	0
0x08	SCU_SCDIVR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WLDIV[2:0]			Res	HDIV[2:0]		
	Reset value																										0	0	0		0	0	0
0x0C	SCU_SCDIVR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SYSTDIV[1:0]		Res	Res	PDIV[1:0]	
	Reset value																											0	0			0	0
0x10	SCU_CLKOCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLKOEN	POLSEL	CLKODIV[2:0]			CLKOS[2:0]		
	Reset value																									0	0	0	0	0	0	0	0
0x14	SCU_CMONCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MONEN	MACTS[1:0]		Res	MONFLAG	NMINTFG	MONCS[1:0]	
	Reset value																									0	0	0		0	0	0	0
0x20	SCU_PPCLKEN1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	T21CLKE	T20CLKE	Res	T12CLKE	T11CLKE	T10CLKE	Res	TSCLKE.	Res	Res	Res	T15CLKE	T14CLKE	T13CLKE	Res	Res	PFCLKE	Res	PDCLKE	PCCLKE	PBCLKE	PACLKE
	Reset value											0	0		0	0	0		0				0	0	0			0	0	0	0	0	0
0x24	SCU_PPCLKEN2	DMACLKE	Res	USBCLKE	CMP1CLKE	CMP0CLKE	Res	Res	Res	Res	Res	SPI1CLKE	SPI0CLKE	FMCLKE	LVICLKE	WDTCLKE	Res	Res	Res	Res	CRCLKE	RTCCLKE	ADCLKE	Res	I2C2CLKE	I2C1CLKE	I2C0CLKE	UST13CLKE	UST12CLKE	Res	Res	UST11CLKE	UST10CLKE	
	Reset value	0		0	0	0						0	0	0	0	1					0	0	0		0	0	0	0	0			0	0

Table 17. SCU Register Map Summary (continued)

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	SCU_PPCLKSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T20CLK	Res.	Res.	Res.	Res.	Res.	USBCLK	Res.	Res.	Res.	Res.	RTCCLK[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDTCLK
	Reset value												1						0					0	0								0
0x60	SCU_PPRST1	Res.	Res.	PLLRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T21RST	T20RST	Res.	T12RST	T11RST	T10RST	Res.	TSRST	Res.	Res.	Res.	T15RST	T14RST	T13RST	Res.	Res.	PFRST	Res.	PDRST	PCRST	PBRST	PARST
	Reset value			0								0	0		0	0	0		0				0	0	0			0	0	0	0	0	0
0x64	SCU_PPRST2	DMARST	Res.	USBRST	CMP1RST	CMP0RST	Res.	Res.	Res.	Res.	Res.	SPI1RST	SPI0RST	FMCRST	LVIRST	Res.	Res.	Res.	Res.	Res.	CRRST	RTCRST	ADRST	Res.	I2C2RST	I2C1RST	I2C0RST	UST13RST	UST12RST	Res.	Res.	UST11RST	UST10RST
	Reset value	0		0	0	0						0	0	0	0						0	0	0		0	0	0	0	0			0	0
0x80	HSECR	WTIDKY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.		Res.	Res.	HSEISEL[1:0]		HSEFLTSEL[1:0]		Res.	HSEFLTDIS
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											0	0	0	0		0
0x84	SCU_LSECR	WTIDKY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.		Res.	Res.	LSEISEL[1:0]		Res.	Res.	Res.	LSESCHEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											0	0				0
0x90	SCU_PLLCON	PLLLOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV[3:0]				FBDIV[7:0]								POSTDIV1[3:0]				POSTDIV2[3:0]			
	Reset value	0												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA0	SCU_VUSB33CR	WTIDKY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.		Res.	Res.	Res.	LDO33EN	Res.	VLDO33[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												0		0	0	0
0xB0	SCU_ECCIESR	Res.	Res.	ECCDIFLAG	ECCCIFLAG	Res.	Res.	Res.	ECCIEN	Res.	Res.	Res.	Res.	Res.	Res.	ECCADDR[17:0]																	
	Reset value			0	0				0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 18. SCU LVI/LVR Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SCU_LVICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVIEN	Res.	LVINTEN	LVIFLAG	LVIVS[3:0]			
	Reset value																									0		0	0	0	0	0	0
0x04	SCU_LVRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVREN[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

5. Port Control Unit (PCU) and GPIO

5.1 Introduction

5.1.1 PCU Introduction

The A31G336 has a Port Control Unit (PCU) module that controls the external input and output (I/O) ports. By setting the PCU registers, users can configure the pins' uses, input / output direction, Pull-up / Pull-down resistor, and debouncing for their applications as needed.

5.1.2 GPIO Introduction

Pins except the VDD, GND, and certain specific-purpose pins can be used as General-Purpose Input/Output (GPIO) pins.

The GPIO module controls the general I/O ports. Output pins can be set to generate high- or low-level signals by configuring the corresponding bits of the GPIO control register; while logic input pins can be monitored for their input status in the control registers.

5.2 Main Features

5.2.1 PCU Features

The Port Control Unit (PCU) configures, and controls external I/Os as listed below:

- MOD registers define the use of each pin.
  - Input / Output / Alternate function
- Internal Pull-up / Pull-down resistor and Push-pull / Open-drain mode can be configured for each pin.
- Interrupts listed below can be set for each pin:
  - Input level interrupt
  - Input rising-edge interrupt
  - Input falling-edge interrupt
  - Input both-edge interrupt
- Up to four GPIO interrupts are supported, including GPIOn (n = A, B, C, D, and F).
- Each pin can be set for debouncing.

5.2.2 GPIO Features

The GPIO module controls the GPIO ports as listed below:

- Selects the output signal level.
- Enables or disables the Pull-up/Pull-down resistor for pins.

5.3 GPIO Functional Description

Subject to the specific hardware characteristics of each I/O port listed in 5.3.5 I/O port characteristics of this manual, each port bit of the general-purpose I/O (GPIO) ports can be individually configured by software in multiple modes:

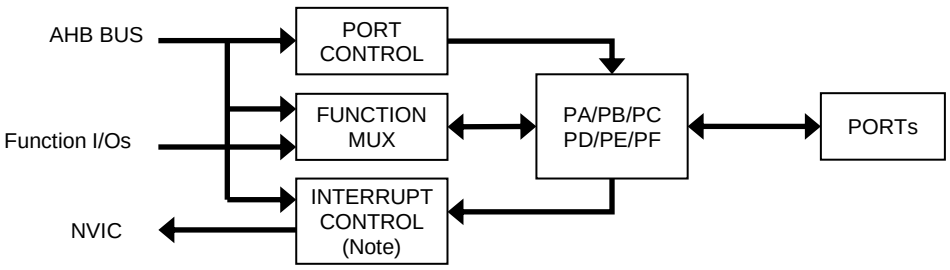
- Input floating
- Input Pull-up
- Input Pull-down
- Analog
- Output Open-drain with Pull-up or Pull-down capability
- Output Push-pull with Pull-up or Pull-down capability
- Alternate function

During and just after reset, the alternate functions are not active and most of the I/O ports are configured in input mode. The debug pins are in Alternate Function pull-up/down after reset:

- PA14: SWCLK with pull-down
- PA13: SWDIO with pull-up

The following figures (Figure 25, Figure 26) show block diagrams of GPIO and external interrupt I/O pins, respectively.

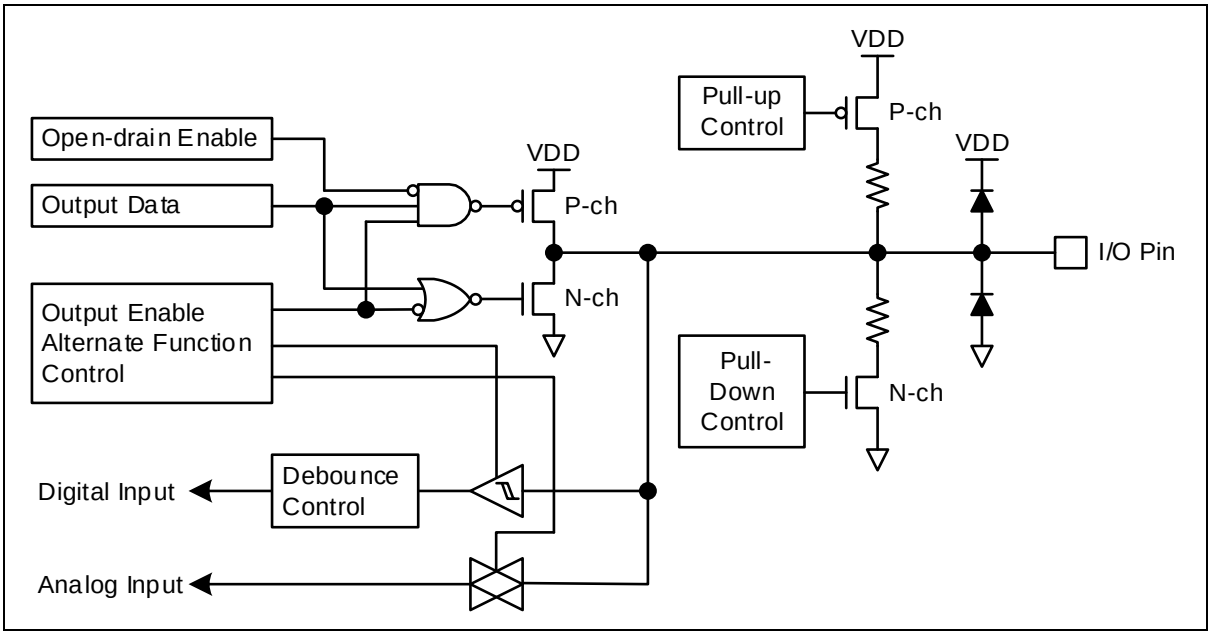
Figure 25. PCU/GPIO Block Diagram



NOTE:

1. Refer to 3.1.2 Interrupt Controller.

Figure 26. I/O Port Block Diagram (GPIO Pins)



5.3.1 GPIO Pins and Internal Signals

Table 19 describes PCU and GPIO internal signal.

Table 19. PCU & GPIO Internal Signal

Signal name	Signal Type	Description
AHB	Clock	AHB clock

Table 20 describes six pins assigned for PCU and GPIO module.

Table 20. PCU and GPIO Pins

Pin name	Type	Description
PA	IO	PA0 to PA10, PA13 to PA15
PB	IO	PB0 to PB15
PC	IO	PC0 to PC15
PD	IO	PD2 to PD5
PF	IO	PF0 to PF2

5.3.2 I/O Port Control Registers

Each GPIO controls up to sixteen I/Os by using three 32-bit memory mapping control registers such as the Pn_MOD, Pn_TYP, and Pn_PUPD registers as shown below:

- The Pn_MOD register selects the I/O mode from the Input, Output, and Alternate modes.
- The Pn_TYP register selects the Push-pull / Open-drain output of each port.
- The Pn_PUPD register controls the Pull-up / Pull-down resistors of all ports I/Os.

5.3.3 I/O Port Data Registers

Each GPIO has two 16-bit memory mapping data registers that operate as shown below:

- The Pn_OUTDR register can be read and written when it has output data.
- The Pn_INDR register stores input data entered via the I/O pin. It can be only read.

5.3.4 I/O Data Bitwise Handling

The bit set register (Pn_BSR) and the bit clear register (Pn_BCR) can be set or clear by the application that controls corresponding bit of the Pn_OUTDR register. These registers support bit banding.

5.3.5 I/O Alternate Function

Each GPIO has two 32-bit memory mapping alternate function registers that operate as shown below:

- The Pn_AFSR1 register selects an alternate function for each port I/O pin (Pin 0 to 7).
- The Pn_AFSR2 register selects an alternate function for each port I/O pin (Pin 8 to 15).

The direction of the alternate function is automatically determined according to the purpose, whether it is input or output.

Table 21 describes the GPIO alternate functions.

Table 21. GPIO Alternate Function

Port	Pin	Alternate Function							
		AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA	0	–	–	TXD13	MOSI13	–	AN0	CP0N2	CP0OUT
	1	T10OUT	T10CAP	RXD13	MISO13	EC10	AN1	CP0P0	SS10
	2	T20OUT	T20CAP	TXD10	MOSI10	EC20	AN2	CP1N2	CP1OUT
	3	T15OUT	T15CAP	RXD10	MISO10	EC15	AN3	CP1P0	–
	4	T20OUT	T20CAP	SS0	SCK10	–	AN4	CP0N0	CP1N0
	5	–	–	–	SCK0	EC20	AN5	CP0N1	CP1N1
	6	T11OUT	T11CAP	–	MISO0	EC11	AN6	–	CP0OUT
	7	T12OUT	T12CAP	–	MOSI0	EC12	AN7	–	CP1OUT
	8	T13OUT	T13CAP	–	SCK11	EC13	SCL2	–	–
	9	T14OUT	T14CAP	TXD11	MOSI11	EC14	SDA2	–	–
	10	T15OUT	T15CAP	RXD11	MISO11	EC15	–	–	–
	13	SWDIO	–	–	SCK10	–	–	–	–
	14	SWCLK	–	TXD10	MOSI10	–	–	–	–
	15	–	–	RXD10	MISO10	EC21	–	SS0	SS13
PB	0	T21OUT	T21CAP	–	SCK12	–	AN8	–	–
	1	T13OUT	T13CAP	–	SS12	EC13	AN9	–	–
	2	T14OUT	T14CAP	–	–	EC14	–	–	–
	3	T20OUT	T20CAP	–	SCK0	–	–	–	–
	4	T15OUT	T15CAP	–	MISO0	EC15	–	–	–
	5	T21OUT	T21CAP	–	MOSI0	–	–	–	–
	6	T11OUT	T11CAP	TXD11	MOSI11	EC11	SCL0	–	–
	7	T12OUT	T12CAP	RXD11	MISO11	EC12	SDA0	–	–
	8	T14OUT	T14CAP	–	SCK11	EC14	SCL0	–	–
	9	T15OUT	T15CAP	SS1	SS11	EC15	SDA0	–	–
	10	T20OUT	T20CAP	TXD12	MOSI12	–	SCL1	SCK1	–
	11	T20OUT	T20CAP	RXD12	MISO12	–	SDA1	–	–
	12	T14OUT	T14CAP	SS1	SCK12	EC14	–	–	–
	13	T21OUT	T21CAP	–	SCK1	–	SCL1	–	–
	14	T10OUT	T10CAP	SS12	MISO1	EC10	SDA1	–	–
	15	T10OUT	T10CAP	–	MOSI1	EC10	ADTRG	–	–

Table 21. GPIO Alternate Function (continued)

Port	Pin	Alternate Function							
		AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PC	0	–	–	–	–	–	AN10	–	–
	1	–	–	–	–	–	AN11	–	–
	2	–	–	–	MISO1	–	AN12	–	–
	3	T13OUT	T13CAP	–	MOSI1	EC13	AN13	–	–
	4	–	–	TXD12	MOSI12	–	AN14	–	–
	5	T21OUT	T21CAP	RXD12	MISO12	–	AN15	–	–
	6	T21OUT	T21CAP	–	–	–	–	–	–
	7	T11OUT	T11CAP	–	–	EC11	–	–	–
	8	T12OUT	T12CAP	–	–	EC12	–	–	–
	9	–	–	–	–	–	ADTRG	–	–
	10	–	–	TXD12	MOSI12	–	TXD13	MOSI13	–
	11	–	–	RXD12	MISO12	EC21	RXD13	MISO13	–
	12	–	–	–	SCK12	–	–	SCK13	–
	13	–	RTCOUT	–	–	–	–	–	–
	14	SXIN	–	–	–	–	–	–	–
	15	SXOUT	–	–	–	–	–	–	–
PD	2	–	–	–	–	–	SCL2	–	–
	3	–	–	–	–	–	SDA2	–	–
	4	–	–	–	SCK13	CLKO	–	–	–
	5	BOOT	RTCOUT	–	–	CLKO	CMUSYNC	–	–
PF	0	XIN	CLKO	–	–	–	–	–	–
	1	XOUT	–	–	–	–	–	–	–
	2	T21OUT	T21CAP	SS12	–	–	–	–	–

NOTES:

1. An unused pin shouldn't be configured as an input floating.
2. After reset, the PA13 and PA14 pins are configured as SWDIO and SWCLK alternate functions, and the internal pull-down on SWCLK and the internal pull-up on SWDIO are activated.
3. The SWCLK and SWDIO pins shouldn't be changed as other alternate functions by software during the pins are connected with debugger host.

5.3.6 Interrupt Functionality

The GPIO module has four interrupt sources, and each port can be an interrupt source. To set each GPIO pins as interrupt sources, users must configure the interrupt trigger selection register (INTC_PnTRIG) and the interrupt control register (INTC_PnCR).

A level-triggered or edge-triggered interrupt can be set for each pin. Once an interrupt occurs, the pin's corresponding bit of the INTC_PnFLAG register is flagged. This flag can be cleared by writing a '1' to the bit.

Table 22. Interrupt Sources and Corresponding Pins in GPIO Module

Interrupt Name	Configurable Pins
GPIOA	Port A[15:13,10:0]
GPIOB	Port B[15:0]
GPIOC	Port C[15:0]
GPIOD	Port D[5:2]
GPIOF	Port F[2:0]

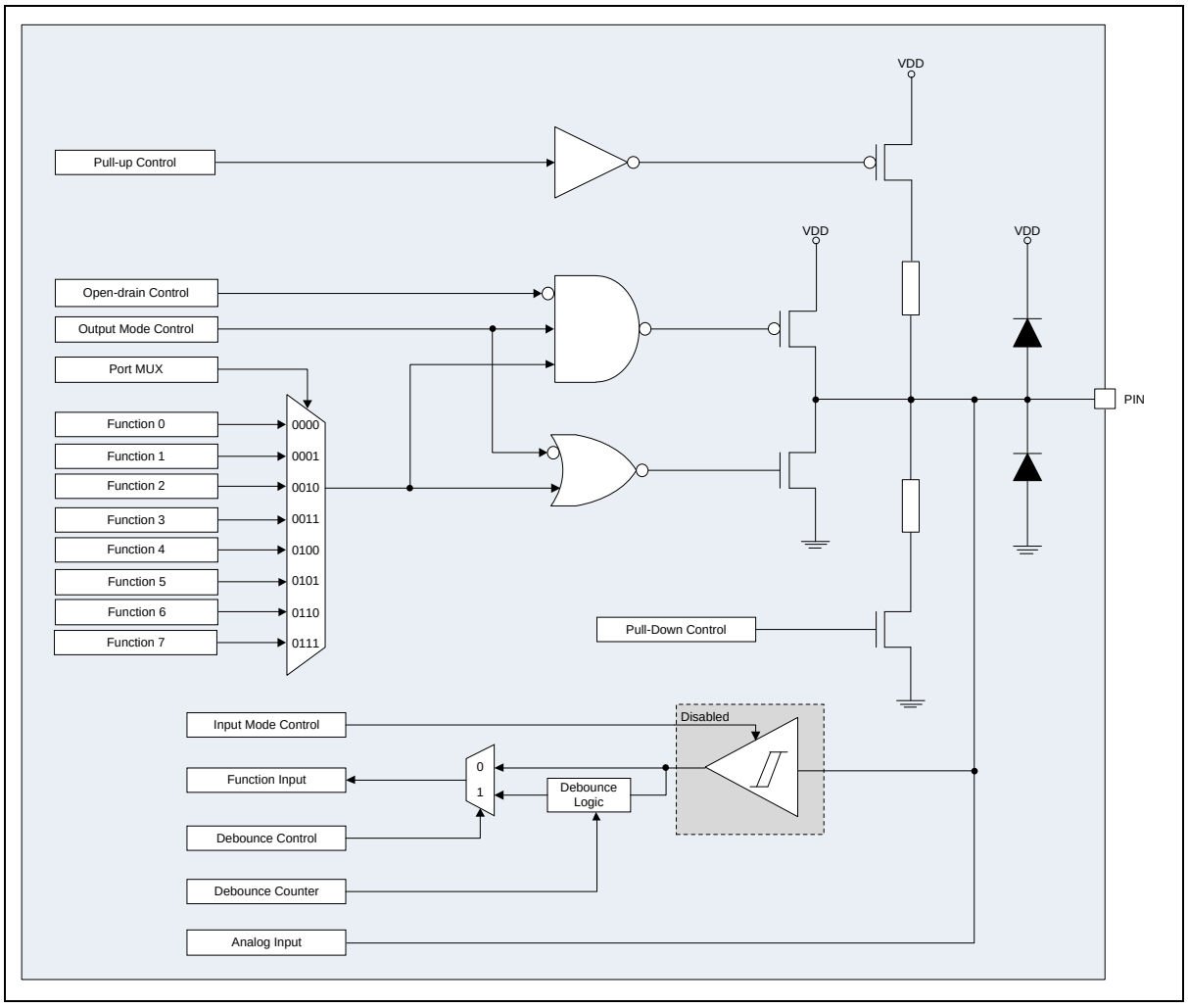
5.3.7 Output Configuration

When the I/O port is programed as output, it features the following:

- The Push-pull driver is turned on or off depending on the Pn_TYP register or function output value.
- Open-drain mode is available.
- The use of the Pull-up and Pull-down resistors is determined by the Pn_PUPD register.

Figure 27 shows a block diagram of an output port.

Figure 27. Output Port Block Diagram



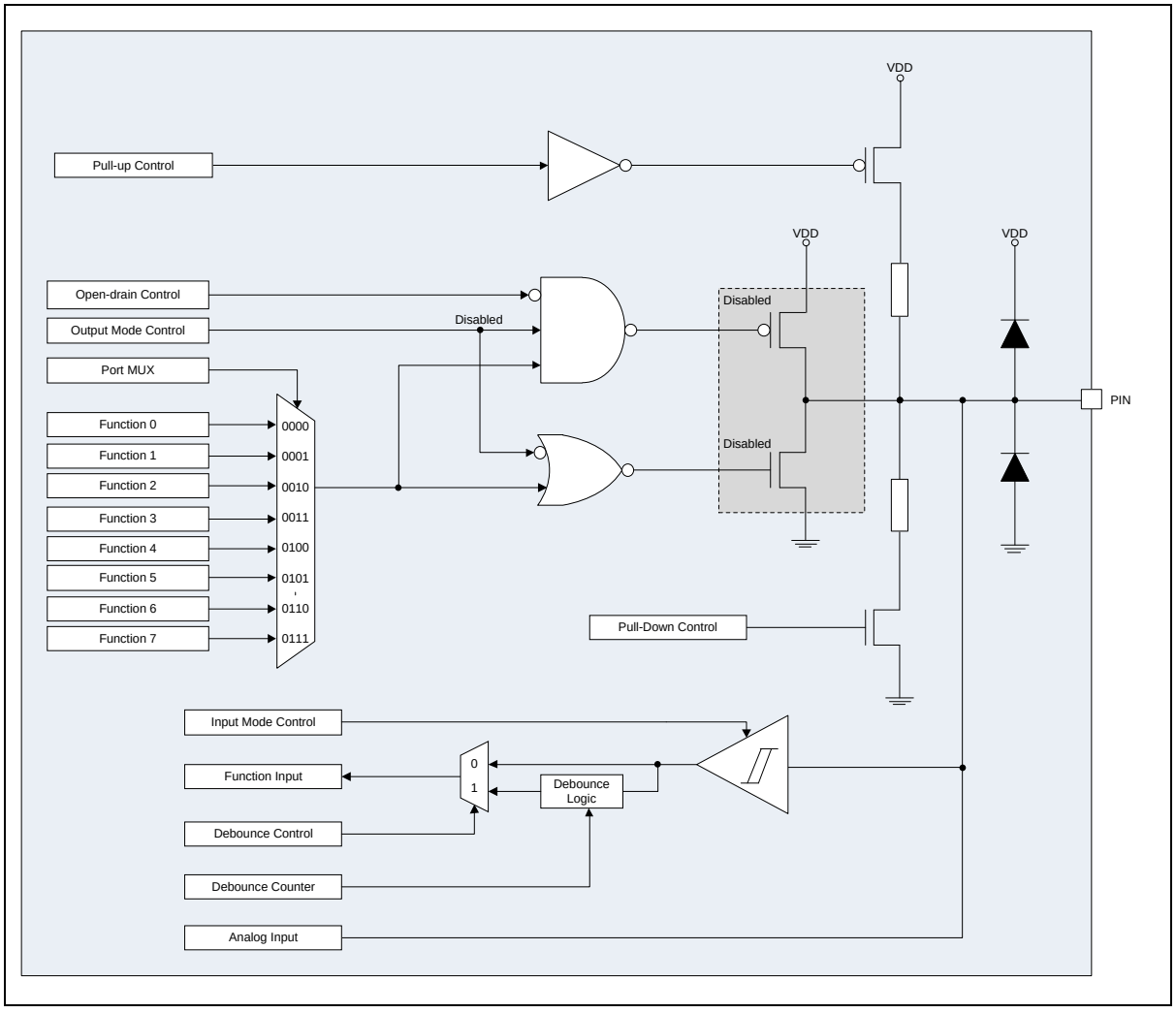
5.3.8 Input Configuration

When the I/O port is programed as input, it features the following:

- All Push-pull drivers are turned off, and Input Schmitt Trigger opens its input.
- Input data is stored in the Pn_INDR register or input signal enters toward functions.

Figure 28 shows a block diagram of an input port.

Figure 28. Input Port Block Diagram



5.3.9 Alternate Function Configuration

For the A31G336, all pins except those with specific purposes can be used as GPIO pins, and the GPIO function can be set by the Pn_MOD register on each port.

Users can select different functions for each port's pins based on the settings of the alternate function selection register. The input data registers capture data entering each I/O pin, as either un-debounced or debounced, at every GPIO clock cycle.

5.3.10 Debouncing Functionality

For the A31G336, each port is capable of debouncing input signals. Debouncing is to filter out noise that can interfere with the port's data input.

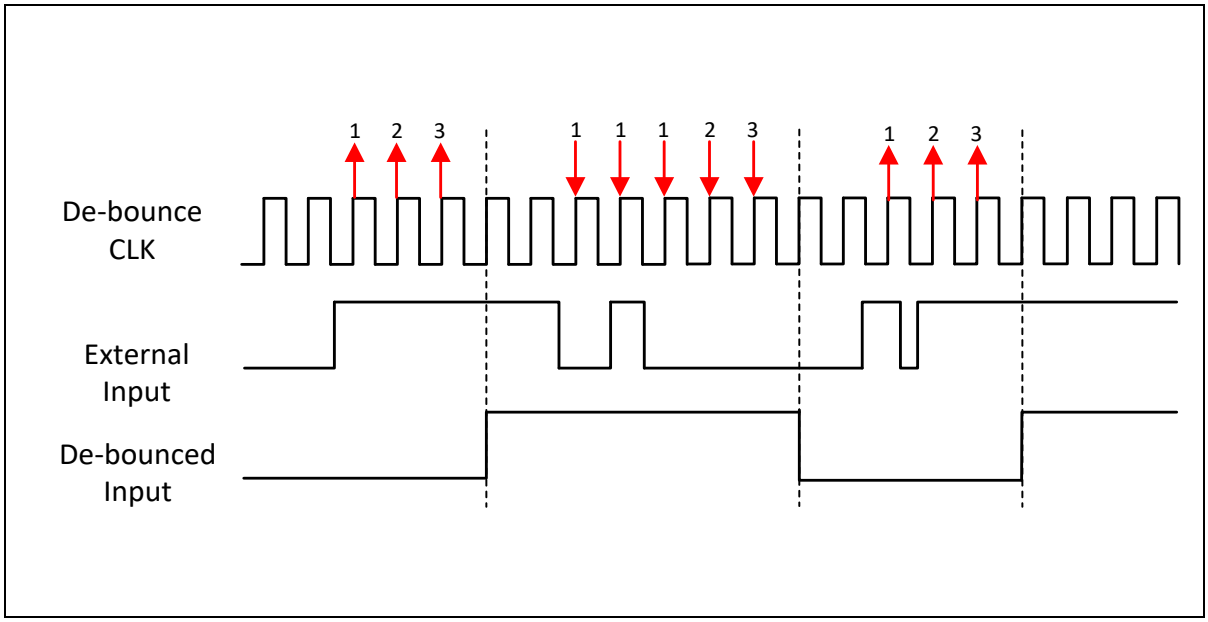
Filtering levels can be adjusted for each 16-pin port, and each individual pin is configured based on whether filtering is enabled or disabled. The debouncing clock used for each port is set in the Pn_DBCR registers.

When enabled, the debouncing logic is clocked at the frequency division ratio set in the Pn_DBCR.DBCLK[2:0] bits, depending on which register the port belongs to.

If a level is not detected on an enabled pin three or more times at a low level at the sampling clock, the signal is eliminated as noise.

Figure 29 shows a timing diagram of a port debouncing.

Figure 29. Example Timing Diagram of Port Debouncing



5.4 GPIO Registers

The base address and register map of PCU are described in the following:

Table 23. Base Address of PCU

Name	Base Address	Description
PA	0x3000_0000	General Port A
PB	0x3000_0100	General Port B
PC	0x3000_0200	General Port C
PD	0x3000_0300	General Port D
PE	0x3000_0400	Reserved
PF	0x3000_0500	General Port F

Table 24. PCU and GPIO Register Map

Name	Offset	Type	Description	Reset Value	Reference
Pn_MOD	0x00	RW	Port n Mode Register	0x0000_0000	5.4.1
Pn_TYP	0x04	RW	Port n Output Type Selection Register	0x0000_0000	5.4.2
Pn_AFSR1	0x08	RW	Port n Alternate Function Selection Register 1	0x0000_0000	5.4.3
Pn_AFSR2	0x0C	RW	Port n Alternate Function Selection Register 2	0x0000_0000	5.4.3
Pn_PUPD	0x10	RW	Port n Pull-up/down Resistor Selection Register	0x0000_0000	5.4.4
Pn_INDR	0x14	RO	Port n Input Data Register	0x0000_xxxx	5.4.5
Pn_OUTDR	0x18	RW	Port n Output Data Register	0x0000_0000	5.4.6
Pn_BSR	0x1C	WO	Port n Output Bit Set Register	0x0000_0000	5.4.7
Pn_BCR	0x20	WO	Port n Output Bit Clear Register	0x0000_0000	5.4.8
Pn_OUTDMSK	0x24	RW	Port n Output Data Mask Register	0x0000_0000	5.4.9
Pn_DBCR	0x28	RW	Port n Debounce Control Register	0x0000_0000	5.4.10

NOTE:

1. n = A, B, C, D, and F.

5.4.1 Pn_MOD: Port n Mode Register

The Pn_MOD register selects one from input mode and output mode for each port pin. Each pin can be configured as an input pin, an output pin or an Alternate Function pin.

The Pn_MOD register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = A to F).

PA_MOD=0x3000_0000, PB_MOD=0x3000_0100, PC_MOD=0x3000_0200
PD_MOD=0x3000_0300, PF_MOD=0x3000_0500

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
MODE15[1:0]	MODE14[1:0]	MODE13[1:0]	MODE12[1:0]	MODE11[1:0]	MODE10[1:0]	MODE9[1:0]	MODE8[1:0]	MODE7[1:0]	MODE6[1:0]	MODE5[1:0]	MODE4[1:0]	MODE3[1:0]	MODE2[1:0]	MODE1[1:0]	MODE0[1:0]
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	MODEx[1:0]	Port n Mode Selection bits, x: 0 to 15	
		00	Input mode
		01	Output mode
		10	Alternate function mode
		11	Reserved

NOTES:

1. The MODEx bits for PC[15:14] and PF[1:0] won't be changed during the corresponding clock (LSE/ HSE) is selected as the system clock (MCLK).
2. The MODEx bits for PA[14:13] and PD5 are set to "10" for alternate function by reset.
3. PA13: SWDIO, PA14: SWCLK, PD5: BOOT
4. The MODEx bits for PC[15:14] are cleared to '0' by the reset of POR but retained by the other reset.

5.4.2 Pn_TYP: Port n Output Type Selection Register

The Pn_TYP register selects an output type of a port pin from Push-pull output and Open-drain output.

The Pn_TYP register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_TYP=0x3000_0004, PB_TYP=0x3000_0104, PC_TYP=0x3000_0204
PD_TYP=0x3000_0304, PF_TYP=0x3000_0504

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																TYP15	TYP14	TYP13	TYP12	TYP11	TYP10	TYP9	TYP8	TYP7	TYP6	TYP5	TYP4	TYP3	TYP2	TYP1	TYP0
-																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	TYPx	Port n Output Type Selection bit, x: 0 to 15
		0 Push-pull output
		1 Open-drain output

5.4.3 Pn_AFSR1/2: Port n Alternate Function Selection Register 1/2

The Pn_AFSR1/2 register must be set properly before using the port. Otherwise, the port may not function properly.

The Pn_AFSR1/2 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

PA_AFSR1=0x3000_0008, PB_AFSR1=0x3000_0108, PC_AFSR1=0x3000_0208
PD_AFSR1=0x3000_0308, PF_AFSR1=0x3000_0508

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
AFSR7[3:0]	AFSR6[3:0]	AFSR5[3:0]	AFSR4[3:0]	AFSR3[3:0]	AFSR2[3:0]	AFSR1[3:0]	AFSR0[3:0]
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4x+3 4x	AFSRx[3:0]	Port n Alternate Function Selection bits, x: 0 to 7	
		0000	Alternate Function 0 (AF0)
		0001	Alternate Function 1 (AF1)
		0010	Alternate Function 2 (AF2)
		0011	Alternate Function 3 (AF3)
		0100	Alternate Function 4 (AF4)
		0101	Alternate Function 5 (AF5)
		0110	Alternate Function 6 (AF6)
		0111	Alternate Function 7 (AF7)
		Others	Reserved

NOTE:

1. The AFSRx bits for PF[1:0] won't be changed during the corresponding clock (HSE) is selected as the system clock (MCLK).

PA_AFSR2=0x3000_000C, PB_AFSR2=0x3000_010C, PC_AFSR2=0x3000_020C
PD_AFSR2=0x3000_030C, PF_AFSR2=0x3000_050C

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
AFSR15[3:0]	AFSR14[3:0]	AFSR13[3:0]	AFSR12[3:0]	AFSR11[3:0]	AFSR10[3:0]	AFSR9[3:0]	AFSR8[3:0]
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4(x-8)+3 4(x-8)	AFSRx[3:0]	Port n Alternate Function Selection bits, x: 8 to 15	
		0000	Alternate Function 0 (AF0)
		0001	Alternate Function 1 (AF1)
		0010	Alternate Function 2 (AF2)
		0011	Alternate Function 3 (AF3)
		0100	Alternate Function 4 (AF4)
		0101	Alternate Function 5 (AF5)
		0110	Alternate Function 6 (AF6)
		0111	Alternate Function 7 (AF7)
		Others	Reserved

NOTES:

1. The AFSRx bits for PC[15:14] won't be changed during the corresponding clock (LSE) is selected as the system clock (MCLK).
2. If LSE clock is used, both PC[15:14] pins should be selected as SXIN and SXOUT functions by the corresponding bits of PC_AFSR2 register.

5.4.4 Pn_PUPD: Port n Pull-Up/Pull-Down Resistor Selection Register

Every pin of the port has an on-chip pull-up/pull-down resistor, which can be configured by Pn_PUPD registers.

The Pn_PUPD register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_PUPD=0x3000_0010, PB_PUPD=0x3000_0110, PC_PUPD=0x3000_0210
PD_PUPD=0x3000_0310, PF_PUPD=0x3000_0510

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PUPD15[1:0]	PUPD14[1:0]	PUPD13[1:0]	PUPD12[1:0]	PUPD11[1:0]	PUPD10[1:0]	PUPD9[1:0]	PUPD8[1:0]	PUPD7[1:0]	PUPD6[1:0]	PUPD5[1:0]	PUPD4[1:0]	PUPD3[1:0]	PUPD2[1:0]	PUPD1[1:0]	PUPD0[1:0]
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1 2x	PUPDx[1:0]	Port n pull-up/pull-down resistor selection bits, x: 0 to 15	
		00	Disable pull-up/pull-down resistor
		01	Enable pull-up resistor
		10	Enable pull-down resistor
		11	Reserved

NOTES:

1. The pull-up/down resistor of PF[1:0] and PC[15:14] is automatically disabled regardless of the corresponding PUPDx value if the pins are configured as the alternate function for X-tal (XIN, XOUT, SXIN, and SXOUT).
2. The PUPDx bits for PA13, PA14, and PD5 are set to “01”, “10”, and “01” for SWDIO/SWCLK/BOOT by reset, respectively.
3. PA13: SWDIO, PA14: SWCLK, PD5: BOOT

5.4.5 Pn_INDR: Port n Input Data Register

Each pin level status can be read in the Pn_INDR register. Except for analog input and alternate mode output, the pin level can be detected in the Pn_INDR register.

The Pn_INDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_INDR=0x3000_0014, PB_INDR=0x3000_0114, PC_INDR=0x3000_0214
PD_INDR=0x3000_0314, PF_INDR=0x3000_0514

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		INDR15	INDR14	INDR13	INDR12	INDR11	INDR10	INDR9	INDR8	INDR7	INDR6	INDR5	INDR4	INDR3	INDR2	INDR1	INDR0
-		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
-		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

x	INDRx	Port n Input Data bit, x: 0 to 15

5.4.6 Pn_OUTDR: Port n Output Data Register

When a pin is set as an output in GPIO mode, the output level of the pin is defined by Pn_OUTDR registers.

The Pn_OUTDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_OUTDR=0x3000_0018, PB_OUTDR=0x3000_0118, PC_OUTDR=0x3000_0218
PD_OUTDR=0x3000_0318, PF_OUTDR=0x3000_0518

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		OUTDR15	OUTDR14	OUTDR13	OUTDR12	OUTDR11	OUTDR10	OUTDR9	OUTDR8	OUTDR7	OUTDR6	OUTDR5	OUTDR4	OUTDR3	OUTDR2	OUTDR1	OUTDR0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDRx	Port n Output Data bit, x: 0 to 15. The OUTDR bits can be individually set/cleared by writing to the Pn_BSR/Pn_BCR register.

5.4.7 Pn_BSR: Port n Output Bit Set Register

The Pn_BSR are used for controlling each bit of the Pn_OUTDR register. Writing a ‘1’ into the specific bit position will set a corresponding bit of Pn_OUTDR to ‘1’. Writing ‘0’ in the register has no effect.

The Pn_BSR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_BSR=0x3000_001C, PB_BSR=0x3000_011C, PC_BSR=0x3000_021C
PD_BSR=0x3000_031C, PF_BSR=0x3000_051C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		BSR15	BSR14	BSR13	BSR12	BSR11	BSR10	BSR9	BSR8	BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

x	BSRx	Port n Output Set bit, x: 0 to 15. These bits are always read to 0x00.	
		0	No effect
		1	Set the corresponding OUTDRx bit (Automatically cleared to 0)

5.4.8 Pn_BCR: Port n Output Bit Clear Register

The Pn_BCR are used for controlling each bit of the Pn_OUTDR register. Writing a ‘1’ into the specific bit will set a corresponding bit of Pn_OUTDR to ‘0’. Writing ‘0’ in this register has no effect.

The Pn_BCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_BCR=0x3000_0020, PB_BCR=0x3000_0120, PC_BCR=0x3000_0220
PD_BCR=0x3000_0320, PF_BCR=0x3000_0520

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

x	BCRx	Port n Output Clear bit, x: 0 to 15. These bits are always read to 0x00.	
		0	No effect
		1	Clear the corresponding OUTDRx bit (Automatically cleared to 0)

5.4.9 Pn_OUTDMSK: Port n Output Data Mask Register

The Pn_OUTDMSK are used for protecting each bit of Pn_OUTDR register. Writing a '1' into the specific bit will protect the corresponding bit of Pn_OUTDR. Writing '0' in this register is unmasked.

The Pn_OUTDMSK register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_OUTDMSK=0x3000_0024, PB_OUTDMSK=0x3000_0124, PC_OUTDMSK=0x3000_0224
PD_OUTDMSK=0x3000_0324, PF_OUTDMSK=0x3000_0524

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		OUTDMSK15	OUTDMSK14	OUTDMSK13	OUTDMSK12	OUTDMSK11	OUTDMSK10	OUTDMSK9	OUTDMSK8	OUTDMSK7	OUTDMSK6	OUTDMSK5	OUTDMSK4	OUTDMSK3	OUTDMSK2	OUTDMSK1	OUTDMSK0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	OUTDMSKx	Port n Output Data Mask bit, x: 0 to 15.	
		0	Unmask. The corresponding OUTDRx bit can be changed.
		1	Mask. The corresponding OUTDRx bit is protected.

5.4.10 Pn_DBCR: Port n Debounce Control Register

The Pn_DBCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = A to F).

PA_DBCR=0x3000_0028, PB_DBCR=0x3000_0128, PC_DBCR=0x3000_0228
PD_DBCR=0x3000_0328, PF_DBCR=0x3000_0528

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DBCLK[2:0]	DBEN15	DBEN14	DBEN13	DBEN12	DBEN11	DBEN10	DBEN9	DBEN8	DBEN7	DBEN6	DBEN5	DBEN4	DBEN3	DBEN2	DBEN1	DBEN0
-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

18 16	DBCLK[2:0]	Port n Debounce Filter Sampling Clock Selection.	
		000	HCLK/1
		001	HCLK/4
		010	HCLK/16
		011	HCLK/64
		100	HCLK/256
		101	HCLK/1024
		110	Reserved
		111	Reserved
x	DBENx	Port n Debounce Enable bit, x: 0 to 15.	
		0	Disable debounce filter
		1	Enable debounce filter

NOTES:

1. If a level is not detected on an enabled pin three or more times in a row at the sampling clock, the signal is eliminated as noise.
2. The port debounce should be disabled before DEEP SLEEP mode.
3. The debounce of the PD5 (BOOT) Pin will be enabled on system reset.

5.4.11 PCU and GPIO Register Map Summary

Table 25. PCU and GPIO Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	Pn_MOD	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	Pn_TYP	Res		Res		Res		Res		Res		Res		Res		Res		TYP15	TYP14	TYP13	TYP12	TYP11	TYP10	TYP9	TYP8	TYP7	TYP6	TYP5	TYP4	TYP3	TYP2	TYP1	TYP0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	Pn_AFSR1	AFSR7[3:0]				AFSR6[3:0]				AFSR5[3:0]				AFSR4[3:0]				AFSR3[3:0]				AFSR2[3:0]				AFSR1[3:0]				AFSR0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	Pn_AFSR2	AFSR15[3:0]				AFSR14[3:0]				AFSR13[3:0]				AFSR12[3:0]				AFSR11[3:0]				AFSR10[3:0]				AFSR9[3:0]				AFSR8[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	Pn_PUPD	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	Pn_INDR	Res		Res		Res		Res		Res		Res		Res		Res		INDR15	INDR14	INDR13	INDR12	INDR11	INDR10	INDR9	INDR8	INDR7	INDR6	INDR5	INDR4	INDR3	INDR2	INDR1	INDR0
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x18	Pn_OUTDR	Res		Res		Res		Res		Res		Res		Res		Res		OUTDR15	OUTDR14	OUTDR13	OUTDR12	OUTDR11	OUTDR10	OUTDR9	OUTDR8	OUTDR7	OUTDR6	OUTDR5	OUTDR4	OUTDR3	OUTDR2	OUTDR1	OUTDR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	Pn_BSR	Res		Res		Res		Res		Res		Res		Res		Res		BSR15	BSR14	BSR13	BSR12	BSR11	BSR10	BSR9	BSR8	BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	Pn_BCR	Res		Res		Res		Res		Res		Res		Res		Res		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	Pn_OUTDMSK	Res		Res		Res		Res		Res		Res		Res		Res		OUTDMSK15	OUTDMSK14	OUTDMSK13	OUTDMSK12	OUTDMSK11	OUTDMSK10	OUTDMSK9	OUTDMSK8	OUTDMSK7	OUTDMSK6	OUTDMSK5	OUTDMSK4	OUTDMSK3	OUTDMSK2	OUTDMSK1	OUTDMSK0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	Pn_DBCR	Res		Res		Res		Res		Res		Res		Res		DBCLK[2:0]		DBEN15	DBEN14	DBEN13	DBEN12	DBEN11	DBEN10	DBEN9	DBEN8	DBEN7	DBEN6	DBEN5	DBEN4	DBEN3	DBEN2	DBEN1	DBEN0
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.5 Interrupt Registers

The base address and register map of the interrupt registers are shown in the following:

Table 26. Base Address of Interrupt Register

Name	Base address	Description
Interrupt Register	0x4000_1000	Interrupt Register

Table 27. Interrupt Register Map

Name	Offset	Type	Description	Reset value	Reference
INTC_PnTRIG	0x0000	RW	Port n Interrupt Trigger Selection Register	0x0000_0000	5.5.1
INTC_PnCR	0x0100	RW	Port n Interrupt Control Register	0x0000_0000	5.5.2
INTC_PnFLAG	0x0200	RW	Port n Interrupt Flag Register	0x0000_0000	5.5.3
INTC_EINTxCONF1 INTC_EINTxCONF2	0x0300	RW	External Interrupt Configuration Register1, 2	0x0000_0000	5.5.4 5.5.5
INTC_MSK	0x0400	RW	Interrupt Source Mask Register	0x0000_0000	5.5.6

NOTE:

1. n = A, B, C, D, and F, X = 0 to 3.

5.5.1 INTC_PnTRIG: Port n Interrupt Trigger Selection Register

The INTC_PnTRIG register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_PATRIG=0x4000_1000, INTC_PBTRIG=0x4000_1004, INTC_PCTRIG=0x4000_1008
INTC_PDTRIG=0x4000_100C, INTC_PFTRIG=0x4000_1014

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ITRIG15	ITRIG14	ITRIG13	ITRIG12	ITRIG11	ITRIG10	ITRIG9	ITRIG8	ITRIG7	ITRIG6	ITRIG5	ITRIG4	ITRIG3	ITRIG2	ITRIG1	ITRIG0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	ITRIGx	Port n Interrupt Trigger Selection bits, x: 0 to 15	
		0	Edge trigger interrupt
		1	Level trigger interrupt

5.5.2 INTC_PnCR: Port n Interrupt Control Register

The INTC_PnCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_PACR=0x4000_1100, INTC_PBCR=0x4000_1104, INTC_PCCR=0x4000_1108
INTC_PDCR=0x4000_110C, INTC_PFCR=0x4000_1114

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
INTCTL15[1:0]	INTCTL14[1:0]	INTCTL13[1:0]	INTCTL12[1:0]	INTCTL11[1:0]	INTCTL10[1:0]	INTCTL9[1:0]	INTCTL8[1:0]	INTCTL7[1:0]	INTCTL6[1:0]	INTCTL5[1:0]	INTCTL4[1:0]	INTCTL3[1:0]	INTCTL2[1:0]	INTCTL1[1:0]	INTCTL0[1:0]
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

2x+1: 2x	INTCTLx[1:0]	Port n Interrupt Control bits, x: 0 to 15	
		00	Disable external interrupt (The flag bit won't be set)
		01	Interrupt on falling edge or on low level
		10	Interrupt on rising edge or on high level
		11	Interrupt on both of falling and rising edge or no level interrupt

NOTE:

1. Do not write "11" to the corresponding INTCTLx[1:0] bits during the ITRIGx bit of INTC_PnTRIG is '1'. Otherwise, a malfunction may occur.

5.5.3 INTC_PnFLAG: Port n Interrupt Flag Register

The INTC_PnFLAG register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_PAFLAG=0x4000_1200, INTC_PBFLAG=0x4000_1204, INTC_PCFLAG=0x4000_1208
INTC_PDFLAG=0x4000_120C, INTC_PFFLAG=0x4000_1214

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FLAG15	FLAG14	FLAG13	FLAG12	FLAG11	FLAG10	FLAG9	FLAG8	FLAG7	FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	FLAGx	Port n Interrupt Flag bits, x: 0 to 15	
		0	No request occurred
		1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.

5.5.4 INTC_EINTnCONF1: External Interrupt Configuration Register 1

The INTC_EINTnCONF1 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_EINT0CONF1=0x4000_1300, INTC_EINT1CONF1=0x4000_1304
INTC_EINT2CONF1=0x4000_1308, INTC_EINT3CONF1=0x4000_130C

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
CONF7[3:0]	CONF6[3:0]	CONF5[3:0]	CONF4[3:0]	CONF3[3:0]	CONF2[3:0]	CONF1[3:0]	CONF0[3:0]
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4x+3: 4x	CONFx[3:0]	Configuration bits for External Interrupt Group n, x: 0 to 7	
		0000	PAx
		0001	PBx
		0010	PCx
		0011	PDx
		0101	PFx

5.5.5 INTC_EINTnCONF2: External Interrupt Configuration Register 2

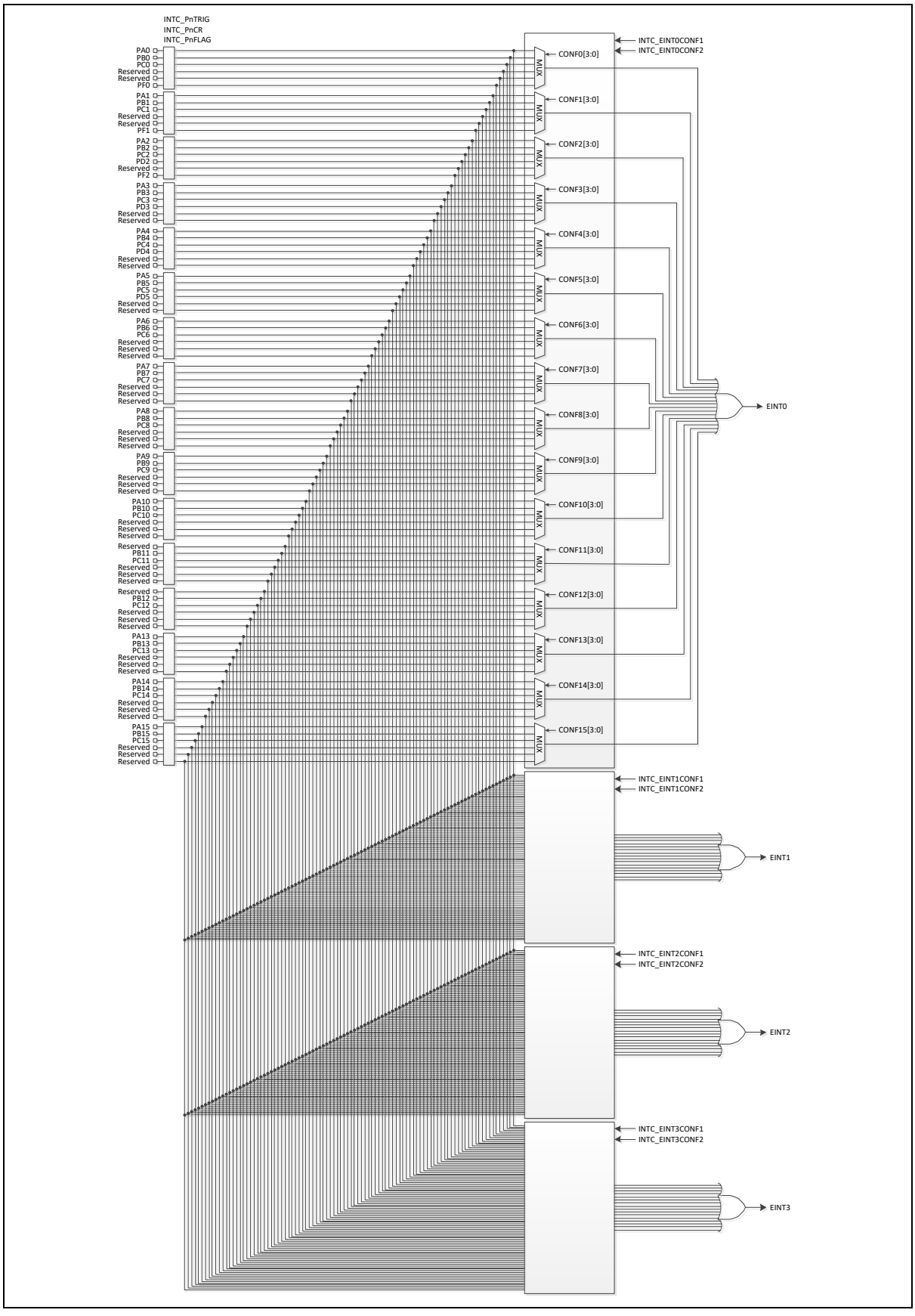
The INTC_EINTnCONF2 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_EINT0CONF2=0x4000_1310, INTC_EINT1CONF2=0x4000_1314
INTC_EINT2CONF2=0x4000_1318, INTC_EINT3CONF2=0x4000_131C

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
CONF15[3:0]	CONF14[3:0]	CONF13[3:0]	CONF12[3:0]	CONF11[3:0]	CONF10[3:0]	CONF9[3:0]	CONF8[3:0]
0000	0000	0000	0000	0000	0000	0000	0000
RW	RW	RW	RW	RW	RW	RW	RW

4(x-8)+3 :4(x-8)	CONFx[3:0]	Configuration bits for External Interrupt Group n, x: 8 to 15	
		0000	PAx
		0001	PBx
		0010	PCx
		0011	PDx
		0101	PFx

Figure 30. Configuration Map for External Interrupt Group 0/1/2/3



5.5.6 INTC_MSK: Interrupt Source Mask Register

The INTC_MSK register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

INTC_MSK=0x4000_1400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMSK31	IMSK30	IMSK29	IMSK28	IMSK27	IMSK26	IMSK25	IMSK24	IMSK23	IMSK22	IMSK21	IMSK20	IMSK19	IMSK18	IMSK17	IMSK16	IMSK15	IMSK14	IMSK13	IMSK12	IMSK11	IMSK10	IMSK9	IMSK8	IMSK7	IMSK6	IMSK5	IMSK4	IMSK3	IMSK2	IMSK1	IMSK0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	IMSKx	Interrupt Source Mask bit, x: 0 to 31
		0 Mask. The corresponding interrupt is disabled.
		1 Unmask

NOTE:

1. A masked interrupt source isn't used for a wake-up source from SLEEP and DEEP-SLEEP mode.

Table 28. Corresponding Interrupts of IMSKx

Source Mask	Interrupt Source Name	Source Mask	Interrupt Source Name
IMSK0	LVI, ECCC	IMSK16	TIMER21
IMSK1	WUT	IMSK17	USART11
IMSK2	WDT	IMSK18	ADC
IMSK3	EINT0	IMSK19	CMP[1:0]
IMSK4	EINT1	IMSK20	TS
IMSK5	EINT2	IMSK21	TIMER13
IMSK6	EINT3	IMSK22	TIMER14
IMSK7	TIMER10	IMSK23	TIMER15
IMSK8	TIMER11	IMSK24	DMACH[1:0]
IMSK9	TIMER12	IMSK25	I2C2
IMSK10	I2C0	IMSK26	USART12
IMSK11	USART10	IMSK27	USART13
IMSK12	RTCC	IMSK28	DMACH[3:2]
IMSK13	Reserved	IMSK29	SPI0
IMSK14	I2C1	IMSK30	SPI1
IMSK15	TIMER20	IMSK31	USB

5.5.7 Interrupt Register Map Summary

Table 29. Interrupt Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000	INTC_PnTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIG15	TRIG14	TRIG13	TRIG12	TRIG11	TRIG10	TRIG9	TRIG8	TRIG7	TRIG6	TRIG5	TRIG4	TRIG3	TRIG2	TRIG1	TRIG0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1100	INTC_PnCR	INTCTL15[1:0]		INTCTL14[1:0]		INTCTL13[1:0]		INTCTL12[1:0]		INTCTL11[1:0]		INTCTL10[1:0]		INTCTL9[1:0]		INTCTL8[1:0]		INTCTL7[1:0]		INTCTL6[1:0]		INTCTL5[1:0]		INTCTL4[1:0]		INTCTL3[1:0]		INTCTL2[1:0]		INTCTL1[1:0]		INTCTL0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1200	INTC_PnFLAG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLAG15	FLAG14	FLAG13	FLAG12	FLAG11	FLAG10	FLAG9	FLAG8	FLAG7	FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1300	INTC_EINTnCONF1	CONF7[3:0]				CONF6[3:0]				CONF5[3:0]				CONF4[3:0]				CONF3[3:0]				CONF2[3:0]				CONF1[3:0]				CONF0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1310	INTC_EINTnCONF2	CONF15[3:0]				CONF14[3:0]				CONF13[3:0]				CONF12[3:0]				CONF11[3:0]				CONF10[3:0]				CONF9[3:0]				CONF8[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1400	INTC_MSK	IMSK31	IMSK30	IMSK29	IMSK28	IMSK27	IMSK26	IMSK25	IMSK24	IMSK23	IMSK22	IMSK21	IMSK20	IMSK19	IMSK18	IMSK17	IMSK16	IMSK15	IMSK14	IMSK13	IMSK12	IMSK11	IMSK10	IMSK9	IMSK8	IMSK7	IMSK6	IMSK5	IMSK4	IMSK3	IMSK2	IMSK1	IMSK0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6. Embedded Flash Memory

6.1 Introduction

The Flash memory controller is an interface controller for embedded Flash memory. It manages data stored on Flash memory.

The Flash memory of the A31G336 has several key features as listed below:

- Code Flash memory for 256 KB with write protection configure option
- 256 KB bulk erase
- 2 KB sector erase
- 512 Bytes page erase
- 32-bit unit write
- Flash Read Protection

6.2 Flash Memory Main Features

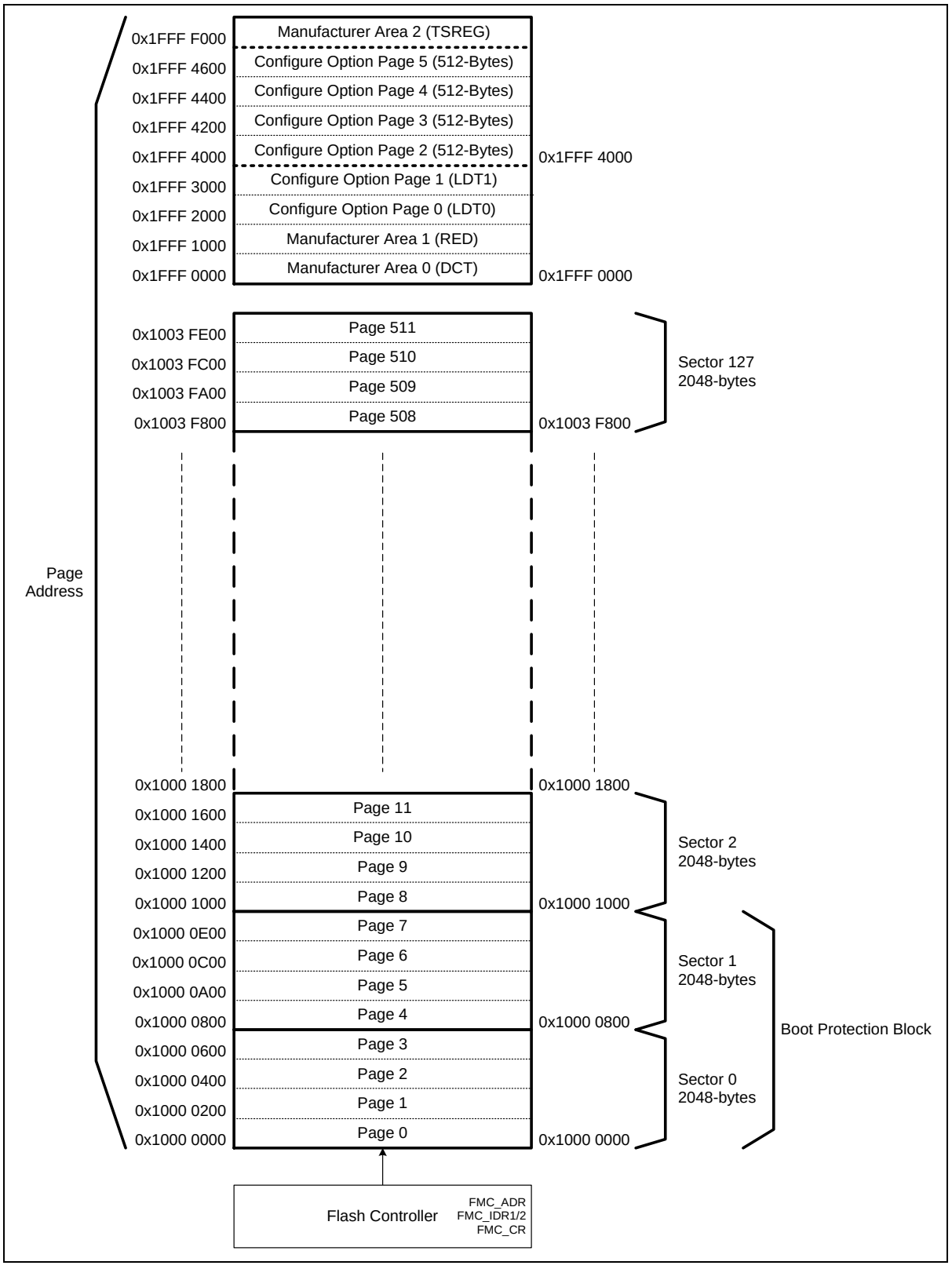
The A31G336 has built-in Flash memory with the following features:

- 256 KB Flash memory with ECC
- 32-bit wide read data bus
- 512-byte sized page
- Page erase, sector erase, and bulk erase
- programming in 32-bit units

Table 30. Flash Memory Features

Item	Description
Size	256 KB
Start Address	0x1000_0000
End Address	0x1003_FFFF
Page Size	512 Bytes
Total Page Count	512 Pages
Sector Size	2 KB
Total Sector Count	128 Sectors
PGM Unit	32-bit
Erase Unit	page, sector, or bulk

Figure 31. Flash Memory Map (256 KB Code Flash)



6.2.1 Procedure For Flash Memory Operation

- The high-speed internal RC oscillator (HSI) should be enabled by software for Flash memory operation.
- The procedure will be cleared, the related registers will be reset, and a FMOPFLAG will be set if wrong sequence.
- Check whether the FMBUSY bit is busy or not. Flash memory operation must start when the bit is ‘0’.
- The address range is 0x10000000 to 0x17FFFFFF when “Flash memory area” is selected.
- The address range is 0x1FFF0000 to 0x1FFFFFFF when “configure option area” is selected.
- If the CPU is running in the Flash memory, the CPU will be held during the Flash memory operation is activated.
- The configure option pages won’t be erased at Flash bulk erase mode.
- The Boot protection block can be erased at Flash bulk erase mode if the BBLOCKBEN bit has correct values
- The CPU should not be in the Flash memory area in Flash bulk erase mode.
- When the read protection level is ‘1’ or ‘2’, the configure option page 1 must be erased if necessary after bulk erase.
- A write to the Flash related register is ignored during Flash operation.
- An NMI source should not be selected while the Flash memory operation is activated.
- The LVR should be enabled during the Flash memory operation is activated (Recommend: 1.93 V over).
- The global interrupt should be disabled.
- The CPU should not enter sleep and deep sleep mode during Flash erase/write mode.
- The FMOPFLAG bit of FMC_ERFLAG register must always be cleared before all Flash erase and write procedures.

6.2.1.1 Flash Page Erase Procedure (512 bytes)

1. Write 0x5FFFFFFF to FMC_ADR during the register is equal to 0x5FFFFF80.
2. Write 0x08192A3B to FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.
3. Write 0x4C5D6E7F to FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.
4. Write a page address to FMC_ADR register.
5. Read and check the FMC_IDR1 and FMC_IDR2 registers in turn.
6. Write 0x6C93A401 (Flash memory area) or 0x6C933801 (configure option area) to FMC_CR register.
7. Check the FMBUSY bit whether '0' or not.
8. Verify the erased page of Flash memory.

6.2.1.2 Flash Sector Erase Procedure (2048 bytes)

1. Write 0x5FFFFFFF to FMC_ADR during the register is equal to 0x5FFFFF80.
2. Write 0x08192A3B to FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.
3. Write 0x4C5D6E7F to FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.
4. Write a sector address to FMC_ADR register.
5. Read and check the FMC_IDR1 and FMC_IDR2 registers in turn.
6. Write 0x6C93A402 (Flash memory area) to FMC_CR register.
7. Check the FMBUSY bit whether '0' or not.
8. Verify the erased page of Flash memory.

6.2.1.3 Flash Write Procedure

1. Write 0x5FFFFFFF to FMC_ADR during the register is equal to 0x5FFFFF80.
2. Write 0x08192A3B to FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.
3. Write 0x4C5D6E7F to FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.
4. Write 0x6C93A404 (Flash memory area) or 0x6C933804 (configure option area) to FMC_CR register.
5. Read and check the FMC_IDR1 and FMC_IDR2 registers in turn.
6. Write word data to a physical Flash address.
7. Check the FMBUSY bit whether '0' or not.
8. Decide the following.
   A. Jump to "the clause G" for the next word program.
   B. Go to "the next clause, J" to finish Flash write procedure.
9. Write 0x5FFFFF80 to FMC_ADR to clear the Flash write procedure and reset the related registers.
10. Verify all data written to the Flash memory

6.2.1.4 Flash Bulk Erase Procedure

1. Write 0x5FFFFFFF to FMC_ADR during the register is equal to 0x5FFFFF80.
2. Write 0x08192A3B to FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.
3. Write 0x4C5D6E7F to FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.
4. Write the value 0x5F9A30D7 to FMC_ADR register.
5. Read and check the FMC_IDR1 and FMC_IDR2 register in turn.
6. Write 0x6C93A410 to FMC_CR register.
7. Check the FMBUSY bit whether '0' or not.
8. Verify all page of Flash memory.

6.2.1.5 Flash Bulk Erase Procedure Including Boot Protection Block

1. Write 0x5FFFFFFF to FMC_ADR during the register is equal to 0x5FFFFF80.
2. Write 0x08192A3B to FMC_IDR1 register during the FMC_ADR register is equal to 0x5FFFFFFF.
3. Write 0x4C5D6E7F to FMC_IDR2 register during the FMC_ADR register is equal to 0x5FFFFFFF.
4. Write the value 0xC1BE5000 to FMC_BCR register.
5. Write the value 0x5F9A30D7 to FMC_ADR register.
6. Read and check the FMC_IDR1 and FMC_IDR2 register in turn.
7. Write 0x6C93A410 to FMC_CR register.
8. Check the FMBUSY bit whether ‘0’ or not.
9. Verify all pages of Flash memory.

6.3 Flash Memory Controller Registers

The base address and register map of the Flash Memory Controller (FMC) are described in the following tables.

Table 31. Base Address of Flash Memory Controller

Name	Base Address
FMC	0x4000_1B00

Table 32. FMC Register Map

Name	Offset	Type	Description	Reset Value	Reference
FMC_ADR	0x0000	RW	Flash Memory Address Register	0x5FFF_FF80	6.3.1
FMC_IDR1	0x0004	RW	Flash Memory Identification Register 1	0x0000_0000	6.3.2
FMC_IDR2	0x0008	RW	Flash Memory Identification Register 2	0x0000_0000	6.3.3
FMC_CR	0x000C	RW	Flash Memory Control Register	0x0000_0000	6.3.4
FMC_BCR	0x0010	RW	Flash Memory Boot Protection Block Bulk Erase Control Register	0x0000_0000	6.3.5
FMC_ERFLAG	0x0014	RW	Flash Memory Error Flag	0x0000_0000	6.3.6
FMC_FETCHCR	0x0020	RW	Flash Memory Data Fetch Control Register	0x0000_0002	6.3.7

6.3.1 FMC_ADR: Flash Memory Address Register

The FMC_ADR register is used to remember the internal Flash memory address. This register is 32 bits wide.

FMC_ADR=0x4000_1B00

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ADDR[31:0]			
0x5FFF_FF80			
RW			

31 0	ADDR[31:0]	Flash Memory Address Pointer. This register is reset to 0x5FFFFF80 immediately after one time operation except “Flash write mode”.

NOTE:

1. The LSB-2byte of the target Flash address is always considered to “0x00”.

6.3.2 FMC_IDR1: Flash Memory Identification Register 1

The FMC_IDR1 register is an internal Flash memory identification register for Flash mode. This register is 32 bits wide.

FMC_IDR1=0x4000_1B04

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ID1[31:0]			
0x0000_0000			
RW			

31 0	ID1[31:0]	Flash Memory Identification 1	
		0x08192A3B	Identification value for a Flash mode
		Others	No identification value

6.3.3 FMC_IDR2: Flash Memory Identification Register 2

The FMC_IDR2 register is an internal Flash memory identification register for Flash mode. This register is 32 bits wide.

FMC_IDR2=0x4000_1B08

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ID2[31:0]			
0x0000_0000			
RW			

31 0	ID2[31:0]	Flash Memory Identification 2	
		0x4C5D6E7F	Identification value for a Flash mode
		Others	No identification value

NOTES:

1. The FMC_IDR1/2 registers are automatically cleared to logic 0x00000000 immediately after one time operation except “Flash Write Mode”.
2. The FMC_IDR1/2 registers should be written with correct values in turn.
3. If incorrect values are written to the FMC_IDR1/2 registers, the registers are cleared to logic 0x00000000.

6.3.4 FMC_CR: Flash Memory Control Register

The FMC_CR register is an internal Flash memory control register. This register is 32-bit size.

FMC_CR=0x4000_1B0C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5	4 3 2 1 0
WTIDKY[15:0]		FMKEY[7:0]	FMBUSY	Reserved	FMOD[4:0]
0x0000		0x00	0	-	0
WO		RW	RO	-	RW

31 16	WTIDKY[15:0]	Write Identification Key.	
		When writing, write 0x6C93 to these bits, or else writing is ignored.	
15 8	FMKEY[7:0]	Flash Memory Operation Area Selection.	
		0x00	Selects no area but for Flash page buffer reset mode.
		0x38	Selects “configure option area” for Flash memory erase/write.
		0xA4	Selects “Flash memory area” for Flash memory erase/write.
		Others	Not allowed. FMOPFLAG will be set.
7	FMBUSY	Flash Memory Operation Mode Busy.	
		0	No effect.
		1	Busy.
4 0	FMOD[4:0]	Flash Memory Operation Mode Selection.	
		00001	“Flash page erase mode” and start when the Flash operation rule is satisfied.
		00010	“Flash sector erase mode” and start when the Flash operation rule is satisfied
		00100	“Flash write mode” and start when the Flash operation rule is satisfied.
		10000	“Flash bulk erase mode” and start when the Flash operation rule is satisfied.
		Others	Not allowed. FMOPFLAG will be set.

NOTES:

1. During a flash memory operation mode, all interrupts are on disable regardless of enable bits.
2. The FMKEY[7:0] and FMOD[3:0] bits are automatically cleared to logic 0x00 immediately after a single operation except “Flash Write Mode”.

6.3.5 FMC_BCR: Flash Memory Boot Protection Block Bulk Erase Control Register

The FMC_BCR register is used to permit bulk erase. This register is 32-bit size.

FMC_BCR=0x4000_1B10

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0
WTIDKY[15:0]		BBLOCKBEN [3:0]		Reserved
0x0000		0000		-
WO		RW		-

31 16	WTIDKY[15:0]	Write Identification Key. When writing, write 0xC1BE to these bits, or else writing is ignored.
15 12	BBLOCKBEN[3:0]	Boot Protection Block Bulk Erase Enable 0x5 Permit “boot protection block” erase when bulk erase Others Protect “boot protection block” erase when bulk erase

NOTE:

1. This register is automatically cleared to logic “0x00” immediately after one time operation.

6.3.6 FMC_ERFLAG: Flash Memory Error Flag Register

The FMC_ERFLAG register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

FMC_ERFLAG=0x4000_1B14

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1	0
Reserved				FMOPFLAG
-				0
-				RW

0	FMOPFLAG	Error bit of Flash Memory Operation Procedure. This bit is set to logic 1 if there is a wrong procedure for Flash memory operation. 0 No wrong procedure. 1 A wrong procedure occurred. The bit is cleared to ‘0’ when ‘1’ is written.

NOTE:

1. This bit may be set to ‘1’ when a breakpoint is inserted or removed in debug mode.

6.3.7 FMC_FETCHCR: Flash Memory Data Fetch Control Register

The FMC_FETCHCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

FMC_FETCHCR=0x4000_1B20

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
WTIDKY[15:0]		Reserved		FWAIT[1:0]
0x0000		-		10
WO		-		RW

31 16	WTIDKY[15:0]	Write Identification Key. When writing, write 0xD7A4 to these bits, or else writing is ignored.
1 0	FWAIT	Wait Fetch Mode Selection bits
		00 0-wait fetch mode (The system clock is up to 20 MHz)
		01 1-wait fetch mode (The system clock is up to 40 MHz)
		10 2-wait fetch mode (The system clock is up to 48 MHz)
		11 3-wait fetch mode (The system clock is up to 48 MHz)

6.3.8 FMC Register Map Summary

Table 33. FMC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FMC_ADR	ADDR[31:0]																															
	Reset value	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0x04	FMC_IDR1	ID1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	FMC_IDR2	ID2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	FMC_CR	WTIDKY[15:0]																FMKEY[7:0]								FMBUSY	Res	Res	FMOD[4:0]				
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0
0x10	FMC_BCR	WTIDKY[15:0]																	BBLOCKBEN[3:0]				Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x14	FMC_ERFLAG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FMOPFLAG
	Reset value																																0
0x20	FMC_FETCHCR	WTIDKY[15:0]																Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FWAIT[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															1	0

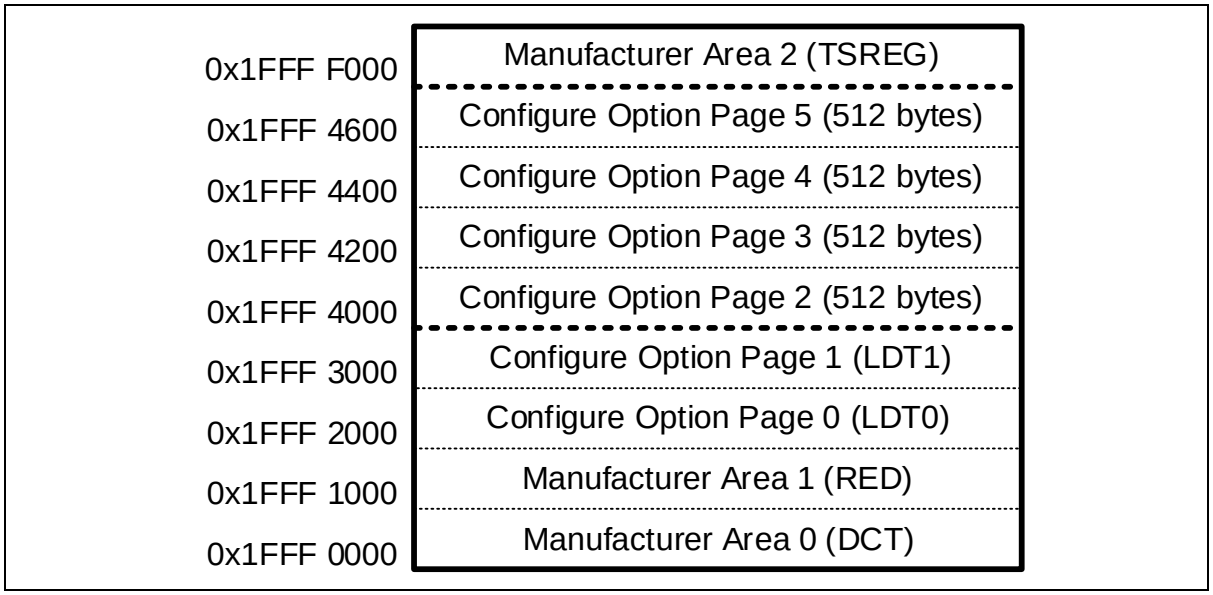
6.4 Configure Option and Manufacturer Areas

Configuration options and manufacturer areas of the A31G336 are used for system related trimming values, user option, and user data. The configure option area consists of six pages in the Flash memory, which can be erased and written by the Flash memory controller. These areas can be read by any instruction.

Six pages of the configure option area are listed below:

- Page 0: System related trimming values
- Page 1: User option for Read Protection, watchdog timer, and LVR voltage level configurations
- Page 2: User data 0 area
- Page 3: User data 1 area
- Page 4: User data 2 area
- Page 5: User data 3 area

Figure 32. Configure Option Area Structure



6.5 Manufacturer Area

The base address of the manufacturer area ranges from 0x1FFF_0000 to 0x1FFF_1FFF and 0x1FFF_F000 to 0x1FFF_FFFF. There are 128-bit unique device ID registers in area 0.

The area map is shown in Table 34.

Table 34. Manufacturer Area Map

Area	Name	Address	Description
0	CONF_MF1CNFIG	0x1FFF_0080	Manufacturer Information 1
	CONF_MF2CNFIG	0x1FFF_0084	Manufacturer Information 2
	CONF_MF3CNFIG	0x1FFF_0088	Manufacturer Information 3
	CONF_MF4CNFIG	0x1FFF_008C	Manufacturer Information 4
	CONF_MFCONFIRM	0x1FFF_0090	Manufacturer Information Confirm Data
1	-	0x1FFF_1000 to 0x1FFF_1FFF	-
2	-	0x1FFF_F000 to 0x1FFF_FFFF	-

6.5.1 CONF_MF1CNFIG: Configuration for Manufacture Information 1

The Configuration for Manufacture Information 1 is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF1CNFIG=0x1FFF_0080

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
LOTNO[31:0]			
Reset Value: Unknown			
R0			

31 0	LOTNO[31:0]	Lot Number

6.5.2 CONF_MF2CNFIG: Configuration for Manufacture Information 2

The Configuration for Manufacture Information 2 is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF2CNFIG=0x1FFF_0084

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
LOTNO[47:32]		WAFNO[15:0]	
Reset Value: Unknown		Reset Value: Unknown	
R0		R0	

31 16	LOTNO[47:32]	Lot Number
15 0	WAFNO[15:0]	Wafer Number

6.5.3 CONF_MF3CNFIG: Configuration for Manufacture Information 3

The Configuration for Manufacture Information 3 is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF3CNFIG=0x1FFF_0088

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
XYCDN[31:0]			
Reset Value: Unknown			
R0			

31 0	XYCDN[31:0]	X and Y Coordinates.

6.5.4 CONF_MF4CNFIG: Configuration for Manufacture Information 4

The Configuration for Manufacture Information 4 is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MF4CNFIG=0x1FFF_008C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
YYMD[31:0]			
Reset Value: Unknown			
R0			

31 0	YYMD[31:0]	Year, Month, and Date.

6.5.5 CONF_MFCONFIRM: Manufacture Information Confirm Data

The Configuration for Manufacture Information Confirm Data is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_MFCONFIRM=0x1FFF_0090

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
UIDFLAG[15:0]		UIDCKS[15:0]	
Reset Value: Unknown		Reset Value: Unknown	
R0		R0	

31 16	UIDFLAG[15:0]	Unique ID Program Confirm Flag
		0xAA55 Unique ID was written normally on the CONF_MFxCNFIG registers
		Others Unique ID wasn't written on the CONF_MFxCNFIG registers
15 0	UIDCKS[15:0]	Unique ID Checksum. The checksum is sum of CONF_MFxCNFIG by byte units.

6.5.6 Manufacturer Area Register Map Summary

Table 35. Manufacturer Area 0 Configuration Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x80	CONF_MF1CNFIG	LOTNO[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x84	CONF_MF2CNFIG	LOTNO[47:32]																WAFNO[15:0]															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x88	CONF_MF3CNFIG	XYCDN[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x8C	CONF_MF4CNFIG	YYMD[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x90	CONF_MFCONFIRM	UIDFLAG[15:0]																UIDCKS[15:0]															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

6.6 Configure Option Pages

The base address of the configure option area ranges from 0x1FFF_2000 to 0x1FFF_4600.

The area map is shown in Table 36.

Table 36. Configure Option Area Map

Page	Name	Address	Description
0	-	0x1FFF_2000 to 0x1FFF_2047 0x1FFF_2050 to 0x1FFF_21FF	System Trimming Values
	TS_FREQ_T30	0x1FFF 2048	Temperature Sensor Output Frequency acquired at 30°C [Hz]
	TS_FREQ_T85	0x1FFF 204C	Temperature Sensor Output Frequency acquired at 85°C [Hz]
1	CONF_RPCNFIG	0x1FFF_3000	Configuration for Read Protection
	CONF_WDTCNFIG	0x1FFF_300C	Configuration for Watch-Dog Timer
	CONF_LVRCNFIG	0x1FFF_3010	Configuration for Low-Voltage Reset
	CONF_CNFIGWTP 1	0x1FFF_3014	Configure and User Data Page Erase/Write Protection 1
	CONF_FMWTP1	0x1FFF_3040	Erase/Write Protection 1 for Flash Memory
	CONF_FMWTP2	0x1FFF_3044	Erase/Write Protection 2 for Flash Memory
	CONF_FMWTP3	0x1FFF_3048	Erase/Write Protection 3 for Flash Memory
	CONF_FMWTP4	0x1FFF_304C	Erase/Write Protection 4 for Flash Memory
2	-	0x1FFF_4000 to 0x1FFF_41FF	User Data Area 0
3	-	0x1FFF_4200 to 0x1FFF_43FF	User Data Area 1
4	-	0x1FFF_4400 to 0x1FFF_45FF	User Data Area 2
5	-	0x1FFF_4600 to 0x1FFF_47FF	User Data Area 3

6.6.1 TS_FREQ_T30: Temperature Sensor Output Frequency 1

The Temperature Sensor Output Frequency 1 is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

TS_FREQ_T30=0x1FFF_2048

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	F_T30[23:0]		
-	Reset Value: Unknown		
.	RW		

23 0	F_T30[23:0]	Temperature Sensor Output Frequency acquired at 30°C [Hz]

6.6.2 TS_FREQ_T85: Temperature Sensor Output Frequency 2

The Temperature Sensor Output Frequency 2 is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

TS_FREQ_T85=0x1FFF_204C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	F_T85[23:0]		
-	Reset Value: Unknown		
.	RW		

23 0	F_T85[23:0]	Temperature Sensor Output Frequency acquired at 85°C [Hz]

6.6.3 CONF_RPCNFIG: Configuration for Read Protection

The configuration for the Flash Memory Read Protection is 32-bit Flash memory. This register is 32-bit size.

CONF_RPCNFIG=0x1FFF_3000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
WTIDKY[27:0]			Reserved	READP[1:0]
0xFFFFFFF			-	11
RW			-	RW

31 4	WTIDKY[27:0]	Write Identification Key These bits are the write key for “Read Protection”. So, The WTIDKY[27:0] should be kept with the 0x69C8A27. Otherwise, the Read Protection will be on level 2.
1 0	READP[1:0]	Read Protection for Flash Memory and Data Flash Memory Area.
		11 Read Protection level 0, No restriction for read/erase/write.
		10 Read Protection level 1, Not readable/erasable/writable by “Debug” Bulk erasable only by ”Debug” Readable/erasable/writable by “Instruction from Flash Memory and RAM”
		0x Read Protection level 2, Where x is don’t care Not readable/erasable/writable by “Debug”/”Instruction from RAM” Bulk erasable only by “Instruction from RAM”/”Debug” Readable/erasable/writable by “Instruction from Flash Memory”

NOTES:

1. The Read Protection level can be changed from lower level to higher level only.
2. The “Configure Option Page 1” cannot be erased by “Debug” unit on “Read Protection level 1/2” and by “Instruction from RAM” on “Read Protection level 2.
3. The configure option area may be read even if the “Read Protection” is on level 1 and 2.
4. An erase/write except a bulk erase isn’t executable by “Instruction from RAM” regardless of the CONF_FMWTP1/2/3/4 registers on read protection level 2.
5. An erase/write except a bulk erase isn’t executable by “Debug” regardless of the CONF_FMWTP1/2/3/4 registers on read protection level 1/2.
6. When the read protection level is ‘1’ or ‘2’, the configure option page 1 must be erased if necessary after bulk erase.

6.6.4 CONF_WDTCNFIG: Configuration for Watchdog Timer

The configuration for the Watchdog Timer is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_WDTCNFIG=0x1FFF_300C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1	0
Reserved		LSIMF[11:0]		Reserved	WCLKMF	WRSTMF	WCNTMF
-		0xFFF		-	1	1	1
-		RW		-	RW	RW	RW

15 4	LSIMF[11:0]	Low-Speed Internal RC Oscillator Master Configuration
	0x96D	The LSI oscillation is decided by the LSIEN of SCU_CLKSRCR register.
	0x2A7	Master enable LSI but disabled at DEEP SLEEP mode.
	Others	Master enable LSI.
	NOTE: 1. If the LSI is selected for MCLK by SCU_SCCR register when the bits are not 0x96D, the CPU cannot wake up at DEEP SLEEP mode. So, only SLEEP mode on the above case should be used for power down.	
2	WCLKMF	Watchdog Timer Clock Selection Master Configuration
	0	Watchdog Timer clock is selected by the WDTCLK of SCU_PPCLKSR register.
	1	Master selection LSI for Watchdog Timer clock
1	WRSTMF	Watchdog Timer Reset Enable Master Configuration
	0	Master enable WDT reset
	1	Disable/Enable of WDT reset is decided by the RSTEN[5:0] of WDT_CR register.
0	WCNTMF	Watchdog Timer Counter Enable Master Configuration
	0	Master enable WDT counter
	1	Disable/Enable WDT counter is decided by the CNTEN[5:0] of WDT_CR register.

6.6.5 CONF_LVRCNFIG: Configuration for Low-Voltage Reset

The configuration for the Low-Voltage Reset is 32-bit Flash memory. This register is 32-bit size.

CONF_LVRCNFIG=0x1FFF_3010

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3 2 1 0
WTIDKY[15:0]		LVRENM[7:0]	Reserved	LVRVS[3:0]
0xFFFF		0xFF	-	1111
RW		RW	-	RW

31 16	WTIDKY[15:0]	Write Identification Key: 0x9D58	
		These bits are the write key for “LVR controller”.	
15 8	LVRENM[7:0]	LVR Reset Operation Control Master Configuration	
		0xAA	LVR operation is decided by the LVREN of SCU_LVRCR register
		Others	Master enable LVR operation
3 0	LVRVS[3:0]	LVR Voltage Selection.	
		1111	1.68 V
		1100	1.93 V
		1011	2.00 V
		1010	2.13 V
		1001	2.31 V
		1000	2.48 V
		0111	2.68 V
		0110	3.05 V
		0101	3.19 V
		0100	3.60 V
		0011	3.74 V
		0010	4.04 V
		0001	4.22 V
		0000	4.50 V

6.6.6 CONF_CNFIGWTP1: Configure and User Data Page Erase/Write Protection 1

The Configure and User Data Page Erase/Write Protection 1 is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_CNFIGWTP1=0x1FFF_3014

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5	4	3	2	1	0
Reserved				UDP3WP	UDP2WP	UDP1WP	UDP0WP	CP1WP
-				1	1	1	1	1
.				RW	RW	RW	RW	RW

4	UDP3WP	User Data Page 3 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction
		1	Disable protection (Erasable/writable by instruction)
3	UDP2WP	User Data Page 2 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction
		1	Disable protection (Erasable/writable by instruction)
2	UDP1WP	User Data Page 1 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction
		1	Disable protection (Erasable/writable by instruction)
1	UDP0WP	User Data Page 0 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction
		1	Disable protection (Erasable/writable by instruction)
0	CP1WP	Configure Option Page 1 Erase/Write Protection	
		0	Enable protection (Not erasable/writable by instruction)
		1	Disable protection (Erasable/writable by instruction)

NOTE:

1. The special area which is protected cannot be erased by page unit.

6.6.7 CONF_FMWTP1: Erase/Write Protection 1 for Flash Memory

The Erase/Write Protection 1 for Flash memory is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_FMWTP1=0x1FFF_3040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWTP31	SWTP30	SWTP29	SWTP28	SWTP27	SWTP26	SWTP25	SWTP24	SWTP23	SWTP22	SWTP21	SWTP20	SWTP19	SWTP18	SWTP17	SWTP16	SWTP15	SWTP14	SWTP13	SWTP12	SWTP11	SWTP10	SWTP9	SWTP8	SWTP7	SWTP6	SWTP5	SWTP4	SWTP3	SWTP2	SWTP1	SWTP0
0xFFFFFFFF																															
RW																															

n	SWTPn	Flash Memory Erase/Write Protection bits, n: 0 to 31 (Sector 0 to Sector 31)
		0 Protect “Flash memory sector n erase/write”
		1 Permit “Flash memory sector n erase/write”

6.6.8 CONF_FMWTP2: Erase/Write Protection 2 for Flash Memory

The Erase/Write Protection 2 for Flash memory is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_FMWTP2=0x1FFF_3044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWTP63	SWTP62	SWTP61	SWTP60	SWTP59	SWTP58	SWTP57	SWTP56	SWTP55	SWTP54	SWTP53	SWTP52	SWTP51	SWTP50	SWTP49	SWTP48	SWTP47	SWTP46	SWTP45	SWTP44	SWTP43	SWTP42	SWTP41	SWTP40	SWTP39	SWTP38	SWTP37	SWTP36	SWTP35	SWTP34	SWTP33	SWTP32
0xFFFFFFFF																															
RW																															

n-32	SWTPn	Flash Memory Erase/Write Protection bits, n: 32 to 63 (Sector 32 to Sector 63)
		0 Protect “Flash memory sector n erase/write”
		1 Permit “Flash memory sector n erase/write”

6.6.9 CONF_FMWTP3: Erase/Write Protection 3 for Flash Memory

The Erase/Write Protection 3 for Flash memory is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_FMWTP2=0x1FFF_3048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWTP95	SWTP94	SWTP93	SWTP92	SWTP91	SWTP90	SWTP89	SWTP88	SWTP87	SWTP86	SWTP85	SWTP84	SWTP83	SWTP82	SWTP81	SWTP80	SWTP79	SWTP78	SWTP77	SWTP76	SWTP75	SWTP74	SWTP73	SWTP72	SWTP71	SWTP70	SWTP69	SWTP68	SWTP67	SWTP66	SWTP65	SWTP64
0xFFFFFFFF																															
RW																															

n-64	SWTPn	Flash Memory Erase/Write Protection bits, n: 64 to 95 (Sector 64 to Sector 95)	
		0	Protect “Flash memory sector n erase/write”
		1	Permit “Flash memory sector n erase/write”

6.6.10 CONF_FMWTP4: Erase/Write Protection 4 for Flash Memory

The Erase/Write Protection 4 for Flash memory is 32-bit Flash memory. This is available at 32-/ 16-/ 8-bit access.

CONF_FMWTP2=0x1FFF_304C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWTP127	SWTP126	SWTP125	SWTP124	SWTP123	SWTP122	SWTP121	SWTP120	SWTP119	SWTP118	SWTP117	SWTP116	SWTP115	SWTP114	SWTP113	SWTP112	SWTP111	SWTP110	SWTP109	SWTP108	SWTP107	SWTP106	SWTP105	SWTP104	SWTP103	SWTP102	SWTP101	SWTP100	SWTP99	SWTP98	SWTP97	SWTP96
0xFFFFFFFF																															
RW																															

n-96	SWTPn	Flash Memory Erase/Write Protection bits, n: 96 to 127 (Sector 96 to Sector 127)	
		0	Protect “Flash memory sector n erase/write”
		1	Permit “Flash memory sector n erase/write”

6.6.11 Configure Option Page Register Map Summary

Table 37. Configure Option Page 0 Configuration Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48	TS_FREQ_T30	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F_T30[23:0]																							
	Reset value									x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x4C	TS_FREQ_T85	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F_T85[23:0]																							
	Reset value									x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Table 38. Configure Option Page 1 Configuration Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CONF_RPCNFIG	WTIDKY[27:0]																												Res.	Res.	READP[1:0]	
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			1	1
0x0C	CONF_WDTCNFIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSIMF[11:0]												Res.	WCLKMF	WRSTMF	WCNTMF
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1
0x10	CONF_LVRCNFIG	WTIDKY[15:0]																LVRENM[7:0]								Res.	Res.	Res.	Res.	LVRVS[3:0]			
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					1	1	1	1
0x14	CONF_CNFIGWTP1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDP3WP	UDP2WP	UDP1WP	UDP0WP	CP1WP
	Reset value																												1	1	1	1	1
0x40	CONF_FMWTP1	SWTP31	SWTP30	SWTP29	SWTP28	SWTP27	SWTP26	SWTP25	SWTP24	SWTP23	SWTP22	SWTP21	SWTP20	SWTP19	SWTP18	SWTP17	SWTP16	SWTP15	SWTP14	SWTP13	SWTP12	SWTP11	SWTP10	SWTP9	SWTP8	SWTP7	SWTP6	SWTP5	SWTP4	SWTP3	SWTP2	SWTP1	SWTP0
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x44	CONF_FMWTP2	SWTP63	SWTP62	SWTP61	SWTP60	SWTP59	SWTP58	SWTP57	SWTP56	SWTP55	SWTP54	SWTP53	SWTP52	SWTP51	SWTP50	SWTP49	SWTP48	SWTP47	SWTP46	SWTP45	SWTP44	SWTP43	SWTP42	SWTP41	SWTP40	SWTP39	SWTP38	SWTP37	SWTP36	SWTP35	SWTP34	SWTP33	SWTP32
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x48	CONF_FMWTP3	SWTP95	SWTP94	SWTP93	SWTP92	SWTP91	SWTP90	SWTP89	SWTP88	SWTP87	SWTP86	SWTP85	SWTP84	SWTP83	SWTP82	SWTP81	SWTP80	SWTP79	SWTP78	SWTP77	SWTP76	SWTP75	SWTP74	SWTP73	SWTP72	SWTP71	SWTP70	SWTP69	SWTP68	SWTP67	SWTP66	SWTP65	SWTP64
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x4C	CONF_FMWTP4	SWTP127	SWTP126	SWTP125	SWTP124	SWTP123	SWTP122	SWTP121	SWTP120	SWTP119	SWTP118	SWTP117	SWTP116	SWTP115	SWTP114	SWTP113	SWTP112	SWTP111	SWTP110	SWTP109	SWTP108	SWTP107	SWTP106	SWTP105	SWTP104	SWTP103	SWTP102	SWTP101	SWTP100	SWTP99	SWTP98	SWTP97	SWTP96
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

7. Direct Memory Access Controller (DMA)

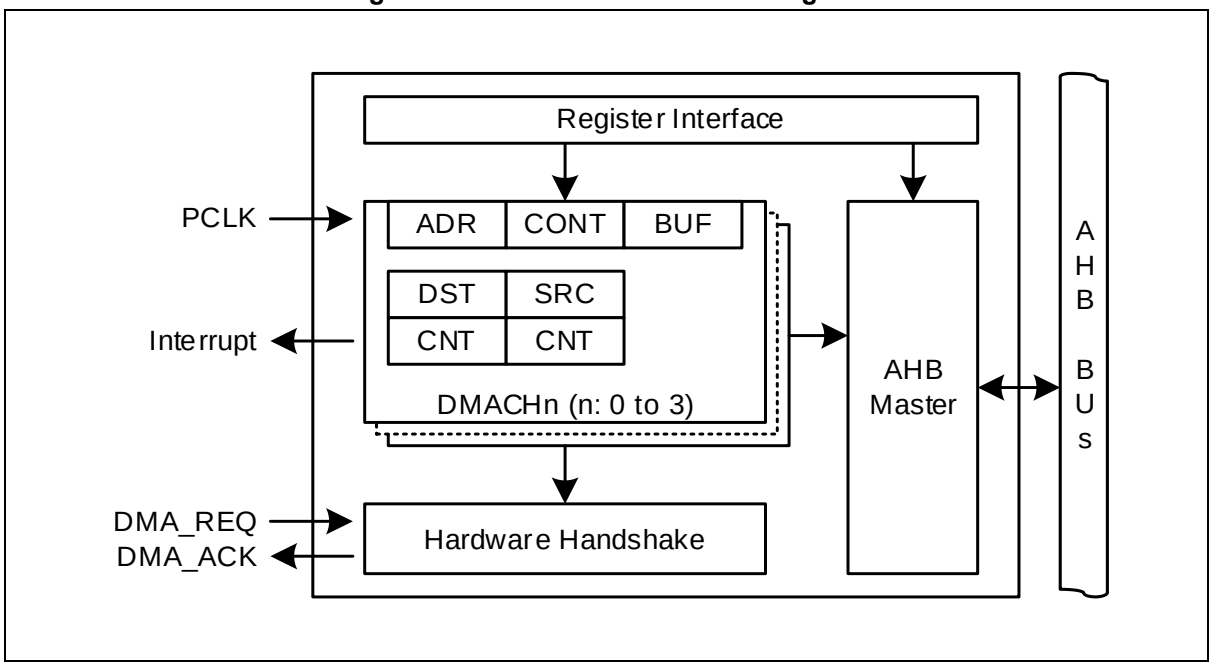
7.1 DMA Introduction

The Direct Memory Access (DMA) controller is used for high-speed data transfers between peripherals and memories. DMA enables quick data transfers of memory-to-peripheral or peripheral-to-memory.

- Four channels
- 8-/ 16-/ 32-bit data transfers supported
- Memory to peripheral transmission
- Peripheral to memory transmission
- Various buffers with the same size supported
- DMA transfers are triggered through peripheral interrupts

Figure 33 shows a DMA block diagram.

Figure 33. DMA Controller Block Diagram



7.2 DMA Registers

The base address of DMA channel 0/1/2/3 is described in the following tables:

Table 39. Base Address of DMA Channel 0/1/2/3

Name	Base Address
DMACH0	0x4000_5D00
DMACH1	0x4000_5D20
DMACH2	0x4000_5D40
DMACH3	0x4000_5D60

Table 40. DMA Channel n Register Map (n = 0 and 3)

Name	Offset	Type	Description	Reset Value	Reference
DMACHn_CR	0x0000	RW	DMA Channel n Control Register	0x0000_0000	7.2.1
DMACHn_IESR	0x0004	RW	DMA Channel n Interrupt Enable and Status Register	0x0000_0000	7.2.2
DMACHn_PAR	0x0008	RW	DMA Channel n Peripheral Address Register	0x4000_0000	7.2.3
DMACHn_MAR	0x000C	RW	DMA Channel n Memory Address Register	0x2000_0000	7.2.4

7.2.1 DMACHn_CR: DMA Channel n Controller Register

The DMACHn_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 to 3)

DMACH0_CR=0x4000_5D00, DMACH1_CR=0x4000_5D20
DMACH2_CR=0x4000_5D40, DMACH3_CR=0x4000_5D60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCNTDIS	Reserved							TRANSCNT[11:0]								ERFGSTP	Reserved				PERISEL[4:0]			MINCDIS	Reserved			SIZE[1:0]		DIR	CHnEN
0	-							0x000								0	-				00000			0	-			00		0	0
RW	-							RW								RW	-				RW			RW	-			RW		RW	RW

31	TCNTDIS	Transfer Counter Decrement Disable bit. The TRANSCNT[11:0] bits won't be decremented during transfer if this bit is '1'.
	0	Enable transfer counter decrement
	1	Disable transfer counter decrement

NOTE:

1. If this bit is set to '1', the MINCDIS bit should be set to '1' by software.

27 16	TRANSCNT[11:0]	The number of times to transfer. These bits should be written out except '0' before the DMA channel is activated.
	0x000	All transmissions have been completed.
	Others	The number of times left to transfer. The value is decremented by '1' every transfer and the transfer ends when it reaches zero.
15	ERFGSTP	Error Flag Stop bit. This bit is used to stop the DMA transfer when an error flag of a selected peripheral is set to '1'.
	0	Disable DMA stop function by an error of peripheral
	1	Enable DMA stop function by an error of peripheral
12 8	PERSEL[4:0]	Peripheral Selection.
	00000	Channel idle
	00001	ADC
	00010	SPI0 Rx
	00011	SPI0 Tx
	00100	SPI1 Rx
	00101	SPI1 Tx
	00110	USART10 Rx
	00111	USART10 Tx
	01000	I2C0 Rx
	01001	I2C0 Tx
	01010	I2C1 Rx
	01011	I2C1 Tx
	01100	USART11 Rx
	01101	USART11 Tx
	01110	USART12 Rx
	01111	USART12 Tx
	10000	USART13 Rx
	10001	USART13 Tx
	10010	I2C2 Rx
	10011	I2C2 Tx

		Others	Reserved
7	MINCDIS	Memory Increment Disable bit. The DMACHn_MAR register won't be incremented during transfer if this bit is '1'.	
		0	Enable memory increment
		1	Disable memory increment
3 2	SIZE[1:0]	Transfer Size Selection.	
		00	8-bits.
		01	16-bits.
		10	32-bits.
		11	Not used.
1	DIR	Transfer Direction.	
		0	Transfer is from memory to peripheral.
		1	Transfer is from peripheral to memory.
0	CHnEN	DMA Channel Enable. This bit is automatically cleared to '0' immediately after transfer completion or error.	
		0	Disable channel n.
		1	Enable channel n.

NOTE:

1. All DMA channels must be disabled by software before entering SLEEP and DEEP SLEEP mode.

NOTES:

1. When ADC is DMA transfer,
   - The EOCIFLAG bit of ADC_IESR register is the request signal of DMA transfer.
   - If the OVRUNIFLAG bit of ADC_IESR register is set during transfer on the ERFGSTP = 1, the corresponding transfer error flag bit is set, and the transfer will be stopped.
2. When SPIn is DMA transfer. Where n = 0 and 1,
   - The SPInIFLAG bit of SPIn_SR register is the request signal of DMA transfer.
3. When USARTn is DMA transfer. Where n = 10, 11, 12, and 13,
   - The DREn and RXCn bits of USARTn_ST register are the request signal for Tx and Rx of DMA transfer.
   - If the DORn, FEn, and PEn bits of USARTn_ST register are set during transfer on the ERFGSTP = 1, the corresponding transfer error interrupt flag bit is set and the transfer will be stopped.
4. When I2Cn is DMA transfer. Where n = 0, 1 and 2,
   - The values of I2Cn_ST register are the request signal for Tx and Rx of DMA transfer.
   - Abbreviated terms
     > "SnDA" is Start and Device address.
     > "rSnDA" is Restart and Device address.
     > "CSnSP" is Clear status and stop.
     > "CS" is Clear status.
     > N is the number of bytes to be received or transmitted.
   - On the master Tx: "SnDA" by software + Transmit(N) by DMA + "CSnSP" by software
   - On the master Rx: Up to "rSnDA" by software + Receive (N-1) by DMA + Receive(1+NACK) and "CSnSP" by software
   - On the slave Tx: Up to "rSnDA" by software + Transmit (N) by DMA + "CS" by software
   - On the slave Rx: "SnDA" by software + Receive(N) by DMA + "CS" by software
   - The corresponding DMA channel should be enabled immediately before "SnDA" on a master Tx or slave Rx.
   - The corresponding DMA channel should be enabled immediately before "rSnDA" on a master Rx or slave Tx.

7.2.2 DMACHn_IESR: DMA Channel n Interrupt Enable and Status Register

The DMACHn_IESR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access (n = 0 to 3).

DMACH0_IESR=0x4000_5D04, DMACH1_IESR=0x4000_5D24
DMACH2_IESR=0x4000_5D44, DMACH3_IESR=0x4000_5D64

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5	4	3 2	1	0
Reserved				TRERIENn	TRCIENn	Reserved	TRERIFGn	TRCIFGn
-				0	0	-	0	0
.				RW	RW	.	RW	RW

5	TRERIENn	Transfer Error Interrupt Enable.
		0 Disable transfer error interrupt.
		1 Enable transfer error interrupt.
4	TRCIENn	Transfer Complete Interrupt Enable.
		0 Disable transfer complete interrupt.
		1 Enable transfer complete interrupt.
1	TRERIFGn	Transfer Error Interrupt Flag bit. This bit is set to ‘1’ when an error occurs on the transfer.
		0 No request occurred.
		1 Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
		NOTE:
		1. This bit will be set by an error occurring during the ERFGSTP bit of DMACHn_CR register is set to 1.
0	TRCIFGn	Transfer Complete Interrupt Flag. This bit is set to ‘1’ when the transfer is finished.
		0 No request occurred.
		1 Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
		NOTE:
		1. On the DIR bit of DMACHn_CR is ‘0’ (Tx of an interface), this bit is set when there is no data to transmit in DMA memory. That is, since the last data is being transmitted, the next Tx should be started after the corresponding Tx is completed. Check the corresponding flag of Tx interrupt to see if the last Tx is complete.

7.2.3 DMACHn_PAR: DMA Channel n Peripheral Address Register

The DMACHn_PAR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 to 3)

DMACH0_PAR=0x4000_5D08, DMACH1_PAR=0x4000_5D28
DMACH2_PAR=0x4000_5D48, DMACH3_PAR=0x4000_5D68

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
PBADR[15:0]		POADR[15:0]	
0x4000		0x0000	
RO		RW	

31 16	PBADR[15:0]	Peripheral Base Address. This is fixed at 0x4000 for APB peripherals.
15 0	POADR[15:0]	Peripheral Offset Address. If the DIR bit is '0', this is the destination offset address of data transfer. If the DIR bit is '1', this is the source offset address of data transfer.

7.2.4 DMACHn_MAR: DMA Channel n Memory Address Register

The DMACHn_MAR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 to 3)

DMACH0_MAR=0x4000_5D0C, DMACH1_MAR=0x4000_5D2C
DMACH2_MAR=0x4000_5D4C, DMACH3_MAR=0x4000_5D6C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
MAR[31:16]		MAR[15]	MAR[14:0]	
0x2000		0	0x0000	
RO		RO	RW	

31 16	MAR[31:16]	Memory Base Address. These bits are fixed with 0x2000.
15	MAR[15]	This bit is fixed with '0'
14 0	MAR[14:0]	Memory Offset Address. This is the address of the memory area from/to which the data will be read/written. This register will be incremented by $2^{SIZE[1:0]}$ for every transfer. When SIZE[1:0] is "01", the MADR[0] bit is ignored. When SIZE[1:0] is "10", the MAR[1:0] bits are ignored. If the DIR bit is '0', this is the source memory address of data transfer. If the DIR bit is '1', this is the destination memory address of data transfer.

7.2.5 DMA Register Map Summary

Table 41. DMA Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DMACHn_CR	TCNTDIS	Res.	Res.	Res.	TRANSCNT[11:0]												ERFGSTP	Res.	Res.	PERISEL[4:0]					MINCDIS	Res.	Res.	Res.	SIZE[1:0]		DIR	CHnEN
	Reset value	0				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0					0	0	0	0
0x04	DMACHn_IESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRERIENn	TRCIENn	Res.	Res.	TRERIFGn	TRCIFGn
	Reset value																											0	0			0	0
0x08	DMACHn_PAR	PBADR[15:0]																POADR[15:0]															
	Reset value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DMACHn_MAR	MAR[31:16]																MAR[15]	MAR[14:0]														
	Reset value	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8. Watchdog Timer (WDT)

8.1 WDT Introduction

Watchdog Timer (WDT) rapidly detects CPU malfunctions such as endless loops caused by noise and returns the CPU to the normal state. WDT signals for malfunction detection can be used as either a CPU reset or an interrupt request.

When the WDT is not being used for malfunction detection, it can be used as a timer to generate interrupts at fixed intervals. When WDT_CNT value reaches WDT_WINDR value, a watchdog interrupt can be generated. The underflow time of the WDT can be set by WDT_DR. If an underflow occurs, an internal reset may be generated. The WDT operates at 40kHz which is the embedded RC oscillator's clock.

8.2 WDT Main Features

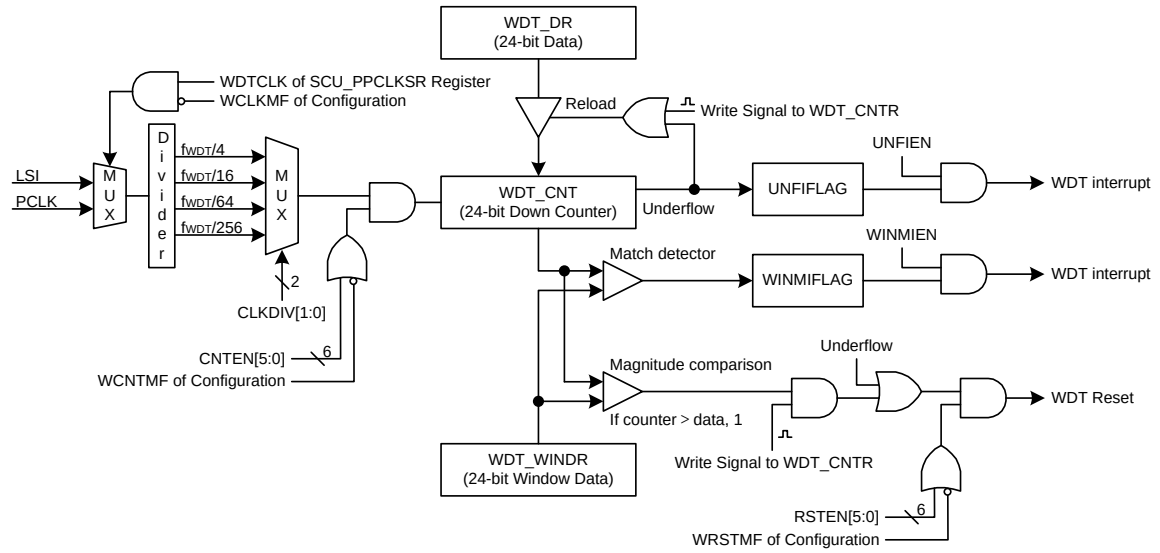
The WDT module has main features as listed below:

- 24-bit down counter (WDT_CNT)
- Reset or periodic interrupt selection
- Count clock selection
- Watchdog overflow output signal
- Counter window function

8.3 WDT Block Diagram

Figure 34 shows a block diagram for the WDT.

Figure 34. WDT Block Diagram



NOTES:

1. When a reload signal is generated, the WDT_DR value is reloaded to WDT_CNT register, after 1-clock delay, with the clock selected by the CLKDIV bits.
2. The WDT_WINDR register is a register that determines the range in which the value of WDT_DR register can be reloaded into the WDT_CNT register without reset occurring in the WDT. In the picture above, it must be reloaded when the value of WDT_CNT register is less than the value of WDT_ WINDR register.

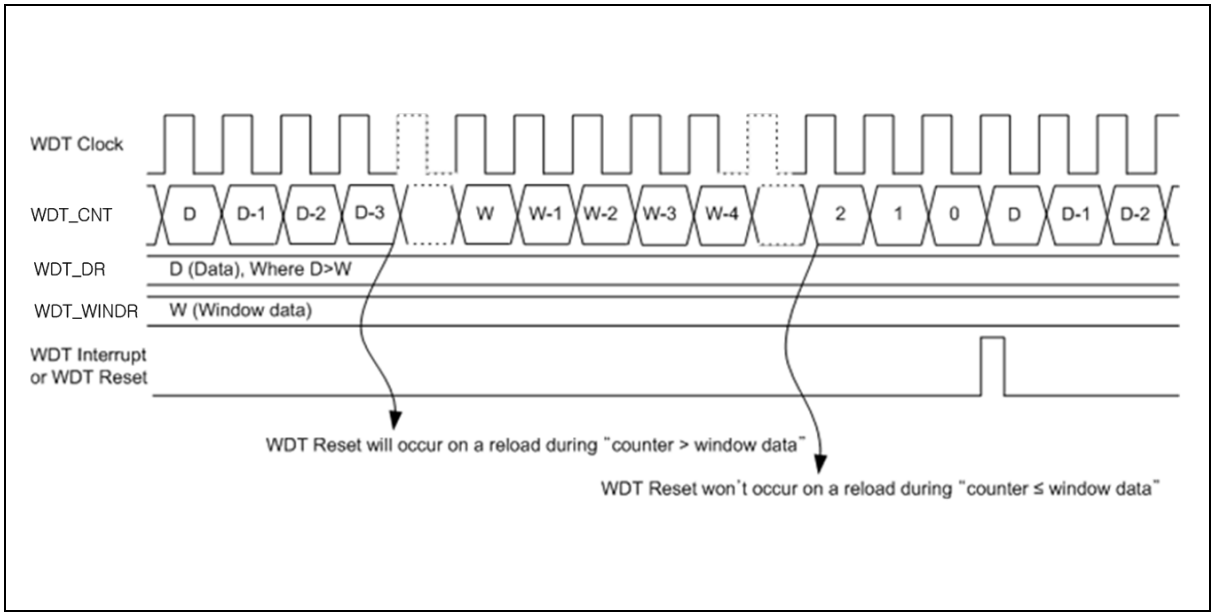
8.4 WDT Functional Description

Watchdog timer count can be enabled by CNTEN (WDT_CR[9:4]) set as any value other than 0x1A. As the WDT activates, the down counter will start counting from the load value. If the RSTEN (WDT_CR[15:10]) is set as any value other than 0x25, WDT reset would be asserted when the WDT counter value reaches 0 (underflow event) from WDT_DR value.

Before WDT counter reaches 0, software can write 0x6A to WDT_CNTR register to reload WDT counter when the counter value is less than or equal to the value of window data register. WDT reset may be asserted if the reload occurs when counter > window data.

8.4.1 Timing Diagram

Figure 35. WDT Interrupt and WDT Reset Timing Diagram



8.4.2 Pre-Scale Table

The WDT includes a 24-bit down counter with programmable pre-scaler to define different time-out intervals.

Clock sources of the WDT can be LSI or PCLK. The PCLK can be selected by setting WDTCLK (SCU_PPCLKSR[0]) to ‘1’ when the CONF_WDTCNFIG[2] bit of configure option page 1 is '0'.

A WDT counter can be set as a base clock by controlling a 2-bit pre-scaler CLKDIV [1:0] in the WDT_CR register. The maximum pre-scaled value is “clock source frequency÷256”. The pre-scaled WDT counter clock frequency values are listed in Table 42.

Selectable clock source (40 kHz ~ 48 MHz) and time-out interval at a single count

Time-out period = (Load Value + 1) × (1/pre-scaled WDT counter clock frequency)

*Time-out period (when the Load Value reaches zero, underflow flag is set to ‘1’)

Table 42. Prescaled WDT Counter Clock Frequency

Clock Source	WDT CLKIN	WDTCLKIN / 4	WDTCLKIN / 16	WDTCLKIN / 64	WDTCLKIN / 256
LSI	40 kHz	10 kHz	2.5 kHz	0.625 kHz	0.156 kHz
PCLK	PCLK	PCLK/4	PCLK/16	PCLK/64	PCLK/256

8.5 WDT Registers

The base address and register map of the WDT are described in the following:

Table 43. Base Address of WDT

Name	Base Address
WDT	0x4000_1A00

Table 44. WDT Register Map

Name	Offset	Type	Description	Reset Value	Reference
WDT_CR	0x0000	RW	Watchdog Timer Control Register	0x0000_0000	8.5.1
WDT_SR	0x0004	RW	Watchdog Timer Status Register	0x0000_0080	8.5.2
WDT_DR	0x0008	RW	Watchdog Timer Data Register	0x0000_0FFF	8.5.3
WDT_CNT	0x000C	RO	Watchdog Timer Counter Register	0x0000_0FFF	8.5.4
WDT_WINDR	0x0010	RW	Watchdog Timer Window Data Register	0x0000_1FFF	8.5.5
WDT_CNTR	0x0014	WO	Watchdog Timer Counter Reload Register	0x0000_0000	8.5.6

8.5.1 WDT_CR: Watchdog Timer Control Register

WDT module should be configured properly before running. The WDT module can reset the system or assert an interrupt signal to the system. This register is 32-bit size.

WDT_CR=0x4000_1A00

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1 0
WTIDKY[15:0]		RSTEN[5:0]	CNTEN[5:0]	WINMIEN	UNFIEN	CLKDIV[1:0]
0x0000		0	0	0	0	0
WO		RW	RW	RW	RW	RW

31 16	WTIDKY[15:0]	Write Identification Key. When writing, write 0x5A69 to these bits, or else writing is ignored.	
15 10	RSTEN[5:0]	Watchdog Timer Reset Enable.	
		0x25	Disable watchdog timer reset.
		Others	Enable watchdog timer reset.
9 4	CNTEN[5:0]	Watchdog Timer Counter Enable.	
		0x1A	Disable watchdog timer counter.
		Others	Enable watchdog timer counter.
3	WINMIEN	Watchdog Timer Window Match Interrupt Enable.	
		0	Disable window data match interrupt.
		1	Enable window data match interrupt.
2	UNFIEN	Watchdog Timer Underflow Interrupt Enable.	
		0	Disable watchdog timer underflow interrupt.
		1	Enable watchdog timer underflow interrupt.
1 0	CLKDIV[1:0]	Watchdog Timer Clock Divider. The watchdog timer clock is selected by SCU_PPCLKSR[0] bit of clock generation and CONF_WDTCNFIG[2] bit of Configure Option Page 1.	
		00	fWDT/4
		01	fWDT/16
		10	fWDT/64
		11	fWDT/256

8.5.2 WDT_SR: Watchdog Timer Status Register

The WDT_SR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

WDT_SR=0x4000_1A04

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4 3 2	1	0
Reserved			DBGCNTEN	Reserved	WINMIFLAG	UNFIFLAG
-			1	-	0	0
-			RW	-	RW	RW

7	DBGCNTEN	Watchdog Timer Counter Enable bit when the core is halted in debug mode.
	0	The watchdog timer counter continues operation even if the core is halted.
	1	The watchdog timer counter stops when the core is halted.

NOTE:

1. This bit is set to '1' by POR reset.

1	WINMIFLAG	Watchdog Timer Window Match Interrupt Flag.
	0	No request occurred.
	1	Request occurred. The bit is cleared to '0' when '1' is written.
0	UNFIFLAG	Watchdog Timer Underflow Interrupt Flag.
	0	No request occurred.
	1	Request occurred. The bit is cleared to '0' when '1' is written.

8.5.3 WDT_DR: Watchdog Timer Data Register

WDT_DR register is used to update WDT_CNT register. This register is 32-bit size.

WDT_DR=0x4000_1A08

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	DATA[23:0]		
-	0x000FFF		
-	RW		

23 0	DATA[23:0]	Watchdog Timer Data. The range is 0x000000 to 0xFFFFFF

NOTE:

1. Once any value is written to this data register, the register cannot be changed until system reset.

8.5.4 WDT_CNT: Watchdog Timer Counter Register

WDT_CNT register represents current count value of the 32-bit down counter. When the counter value reaches 0, an interrupt or a reset will be asserted. This register is 32-bit size.

WDT_CNT=0x4000_1A0C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	CNT[23:0]		
-	0x000FFF		
-	RO		

23 0	CNT[23:0]	Watchdog Timer Counter

8.5.5 WDT_WINDR: Watchdog Timer Window Data Register

WDT_WINDR register is used to compare to WDT_CNT for WINDOW function. This register is 32-bit size.

WDT_WINDR=0x4000_1A10

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	WDATA[23:0]		
-	0x001FFF		
.	RW		

23 0	WDATA[23:0]	Watchdog Timer Window Data. The range is 0x000000 to 0xFFFFFF.

NOTE:

1. Once any value is written to this window data register, the register cannot be changed until the system is reset.

8.5.6 WDT_CNTR: Watchdog Timer Counter Reload Register

WDT_CNTR register is used to generate a reload signal. When a reload signal is generated, the WDT_DR value is reloaded to WDT_CNT. This register is 32-bit size.

WDT_CNTR=0x4000_1A14

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			CNTR[7:0]
-			0x00
.			WO

7 0	CNTR[7:0]	Watchdog Timer Counter Reload bits.	
		0x6A	Reload the WDT_DR value to watchdog timer counter and re-start. (Automatically cleared to “0x00” after operation)
		Others	No effect

8.5.7 WDT Register Map Summary

Table 45. WDT Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WDT_CR	WTIDKY[15:0]																RSTEN[5:0]						CNTEN[5:0]						WINMIEN	UNFIEN	CLKDIV[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	WDT_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGCNTEN	Res.	Res.	Res.	Res.	Res.	WINMIFLAG	UNFIFLAG
	Reset value																									1						0	0
0x08	WDT_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	WDT_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0x10	WDT_WINDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0x14	WDT_CNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

9. REAL TIME CLOCK AND CALENDAR (RTCC)

9.1 RTCC Introduction

Real Timer Clock and Calendar (RTCC) has a function for RTC (Real Time Clock) and calendar operations. Internal structure of the RTCC is implemented with the clock source select circuit, second/minute/hour/day/week/month/year counter circuits, alarm circuit, output select circuit, and error correction circuit.

RTCC is an independent BCD counter. The RTCC circuitry and the related control bits are not reset by a system reset other than POR.

9.2 RTCC Main Features

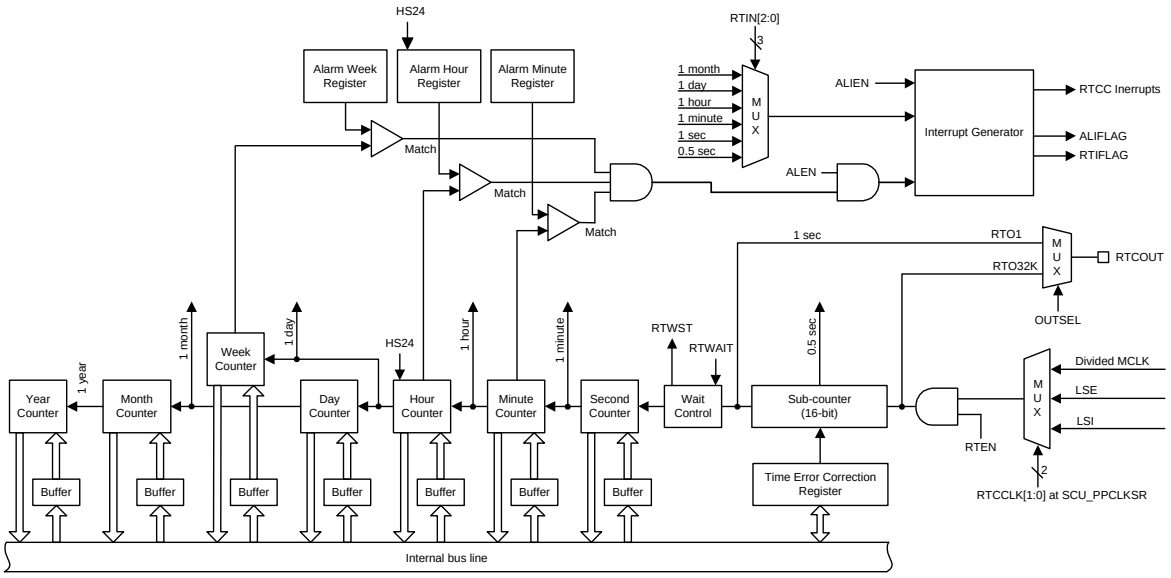
The RTCC module has main features as listed below:

- Calendar with 0.5 seconds, seconds, minutes, hours, day, week, month, and year up to 2099
- Time error correction function
- Alarm function with interrupt
- Wake-up possible from DEEP SLEEP mode

9.3 RTCC Block Diagram

Figure 36 shows a block diagram for the RTCC.

Figure 36. RTCC Block Diagram



NOTE:

1. Time in the block diagram is based on 32.768 kHz.

9.4 RTCC Registers

The base address and register map of the RTCC are described in the following:

Table 46. Base Address of WT

Name	Base Address
RTCC	0x4000_5200

Table 47. WT Register Map

Name	Offset	Type	Description	POR Reset Value (Retained at Other Resets)	Reference
RTC_CR	0x0000	RW	RTCC Control Register	0x0000_0000	9.4.1
RTC_ECR	0x0004	RW	RTCC Time Error Correction Register	0x0000_0000	9.4.2
RTC_SCNT	0x0008	RO	RTCC Sub-counter Register	0x0000_0000	9.4.3
RTC_SEC	0x000C	RW	RTCC Second Counter Register	0x0000_0000	9.4.4
RTC_MIN	0x0010	RW	RTCC Minute Counter Register	0x0000_0000	9.4.5
RTC_HOUR	0x0014	RW	RTCC Hour Counter Register	0x0000_0012	9.4.6
RTC_DAY	0x0018	RW	RTCC Day Counter Register	0x0000_0001	9.4.7
RTC_WEEK	0x001C	RW	RTCC Week Counter Register	0x0000_0000	9.4.8
RTC_MONTH	0x0020	RW	RTCC Month Counter Register	0x0000_0001	9.4.9
RTC_YEAR	0x0024	RW	RTCC Year Counter Register	0x0000_0000	9.4.10
RTC_ALMIN	0x0028	RW	RTCC Alarm Minute Register	0x0000_0000	9.4.11
RTC_ALHOUR	0x002C	RW	RTCC Alarm Hour Register	0x0000_0012	9.4.12
RTC_ALWEEK	0x0030	RW	RTCC Alarm Week Register	0x0000_0000	9.4.13

9.4.1 RTC_CR: RTCC Control Register

The RTC_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_CR=0x4000_5200

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14 13 12	11	10	9	8	7	6	5	4 3 2	1	0
Reserved		RTEN	RTIN[2:0]	RTIFLAG	HS24	Reserved	OUTSEL	ALEN	ALIEN	ALIFLAG	Reserved	RTWST	RTWAIT
-		0	0	0	0	-	0	0	0	0	-	0	0
-		RW	RW	RW	RW	-	RW	RW	RW	RW	-	RO	RW

15	RTEN	RTCC Enable.
	0	RTCC Disable
	1	RTCC Enable
14 12	RTIN[2:0]	RTCC Interrupt Interval Selection bits.
	000	Disable RTCC Interval Interrupt
	001	Once per 0.5 sec
	010	Once per 1 sec
	011	Once per 1 min
	100	Once per 1 hour
	101	Once per 1 day
	110	Once per 1 month
	111	Reserved. Value is not changed.
	NOTE: 1. When changing the values of RTIN[2:0] while the counter operates (RTEN = 1), rewrite the values of RTIN[2:0] after disabling interrupt servicing RTCC Interrupt by using the Interrupt Source Mask Register (INTC_MSK). Furthermore, after rewriting the values of RTIN[2:0], enable interrupt servicing after clearing the RTIFLAG flag.	
11	RTIFLAG	RTCC Interval Interrupt Flag
	0	No request occurred
	1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
10	HS24	12/24-hour System Selection.
	0	12-hour system
	1	24-hour system
	NOTE: 1. Rewrite the HS24 value after setting RTWAIT (bit 0 of RTC_CR) to 1. If the HS24 value is changed, the values of the RTCC hour counter register (RTC_HOUR) and RTCC alarm hour register (RTC_ALHOUR) change according to the specified time system. ‘Value of RTC_HOUR/RTC_ALHOUR by HS24 bit’ Table shows the displayed time digits.	
8	OUTSEL	RTCOUT Selection.
	0	RTO 1 (1 Hz)
	1	RTO 32K (32 kHz)
7	ALEN	RTCC Alarm Match Operation Enable.
	0	Disable RTCC alarm match operation
	1	Enable RTCC alarm match operation

		NOTE: 1. When setting a value to the ALEN bit while the counter operates (RTEN = 1) and ALIEN = 1, rewrite the ALEN bit after disabling interrupt servicing RTCC Interrupt by using Interrupt Source Mask Register (INTC_MSK). Furthermore, clear the ALIFLAG flag after rewriting the ALEN bit. When setting each alarm register (ALIEN flag of RTC_CR, the RTC_ALMIN register, the RTC_ALHOUR register, and the RTC_ALWEEK register), set match operation to be invalid (“0”) for the ALEN bit
6	ALIEN	RTCC Alarm Match Interrupt Enable. 0 Disable 1 Enable
5	ALIFLAG	RTCC Alarm Match Interrupt Flag. 0 No request occurred 1 Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
1	RTWST	RTCC Wait Status Flag. 0 Counter is operating 1 Mode to read/write counter value **NOTE:** 1. This status flag indicates whether the setting of RTWAIT is valid. Before reading or writing the counter value, confirm that the value of this flag is ‘1’.
0	RTWAIT	RTCC Wait Control. 0 Set counter Operation 1 Stop RTSEC to RTYEAR counters for read/write counter value. **NOTE:** 1. This bit controls the operation of the counter. Be sure to write ‘1’ to it to read or write the counter value. Because the RTCC sub-counter (RTC_SCNT) continues operation, complete reading or writing of it in 1 second, and clearing this bit back to ‘0’. When RTWAIT = 1, it takes up to 2 clocks (RTCC clock) until the counter value can be read or written. If the RTCC sub-counter (RTC_SCNT) overflows when RTWAIT = 1, it counts up after RTWAIT = 0. If the RTCC second counter register (RTC_SEC) is written, however, it does not count up because RTCC sub-counter (RTC_SCNT) is cleared.

9.4.2 RTC_ECR: RTCC Time Error Correction Register

The RTC_ECR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_ECR=0x4000_5204

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5 4 3 2 1 0
Reserved			ECTM	ECSIGN	ECV[5:0]
-			0	0	0
-			RW	RW	RW

7	ECTM	Time Error Correction Timing Selection
	0	Corrects time error when the second digits are at 00H, 20H, or 40H. (every 20 seconds)
	1	Corrects watch error only when the second digits are at 00H. (every 60 seconds)
	NOTE: 1. Do not write to the RTC_ECR register at the following timing. A. When ECTM = 0 is set: For a period of SEC = 00H, 20H, 40H B. When ECTM = 1 is set: For a period of SEC = 00H	
6	ECSIGN	Time Error Correction Data Sign
	0	Increases by {(ECV5, ECV4, ECV3, ECV2, ECV1, ECV0) – 1} × 2.
	1	Decreases by {(/ECV5, /ECV4, /ECV3, /ECV2, /ECV1, /ECV0) + 1} × 2.
	NOTES: 1. When (ECSIGN, ECV5, ECV4, ECV3, ECV2, ECV1, ECV0) = (n, 0, 0, 0, 0, 0, n), the watch error is not corrected. (Where n = 0 or 1.) 2. /ECV5 to /ECV0 are the inverted values of the corresponding bits (000011 when 111100). 3. Range of correction value: (when ECSIGN = 0) 2, 4, 6, 8, … , 120, 122, 124 (when ECSIGN = 1) –2, –4, –6, –8, … , –120, –122, –124	
5 0	ECV[5:0]	Time Error Correction Data

9.4.3 RTC_SCNT: RTCC Sub Control Register

The RTC_SCNT register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_SCNT=0x4000_5208

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		RTCNT[15:0]	
-		0	
-		RO	

15 0	RTCNT[15:0]	RTCC Sub-counter.

NOTES:

1. When a correction is made by using the RTC_ECR register, the value of RTC_SCNT may become 8000H or more. The RTC_SCNT is also cleared by writing the RTCC second counter register.
2. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

9.4.4 RTC_SEC: Second Counter Register

The RTC_SEC register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_SEC=0x4000_520C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RSEC[6:0]
-			0
-			RW

6 0	RSEC[6:0]	RTCC Second Counter.

NOTES:

1. The RTC_SEC register takes a value of 0 to 59 (decimal) and indicates the count value of seconds. It counts up when the RTCC sub-counter overflows.
2. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is tried to be written in the register, the value is ignored.

9.4.5 RTC_MIN: RTCC Minute Counter Register

The RTC_MIN register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_MIN=0x4000_5210

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RMIN[6:0]
-			0
-			RW

6 0	RMIN[6:0]	Watch Timer Operation Enable.

NOTES:

1. The RTC_MIN register has a value of 0 to 59 (decimal) and indicates the count value of minutes. It counts up when the RTCC second counter register overflows.
2. Even if the RTCC second counter register overflows while this register is being written, this register ignores the overflow and is set to the value written.
3. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is tried to be written in the register, the value is ignored.

9.4.6 RTC_HOUR: RTCC Hour Counter Register

The RTC_HOUR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_HOUR=0x4000_5214

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4 3 2 1 0
Reserved				RHOUR[5:0]
-				0x12
-				RW

5 0	RHOUR[5:0]	RTCC Hour Counter.

NOTES:

1. The RTC_HOUR register has a value of 00 to 23 or 01 to 12, 21 to 32 (decimal) and indicates the count value of hours. It counts up when the RTCC minute counter register overflows.
2. Even if the RTCC minute counter register overflows while this register is being written, this register ignores the overflow and is set to the written value. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using HS24 bit of RTC_CR(RTCC control register). If a value outside this range is tried to be written in the register, the value is ignored.
3. RTHOUR5 bit of RTC_HOUR indicates AM(0)/PM(1) if HS24(RTC_CR[10]) = 0 (if the 12-hour system is selected).

Table 48. Value of RTC_HOUR/RTC_ALHOUR by HS24 bit

24-Hour Display (HS24 bit = 1)		12-Hour Display (HS24 bit = 0)	
Time	**RTC_HOUR Register RTC_ALHOUR Register**	**Time**	**RTC_HOUR Register RTC_ALHOUR Register**
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

9.4.7 RTC_DAY: RTCC Day Counter Register

The RTC_DAY register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_DAY=0x4000_5218

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RDAY[5:0]
-			1
-			RW

5 0	RDAY[5:0]	RTCC Day Counter.

NOTES:

1. The RTC_DAY register has a value of 1 to 31 (decimal) and indicates the count value of days. It counts up when the RTCC hour counter register overflows.
2. Even if the RTCC hour counter register overflows while this register is being written, this register ignores the overflow and is set to the value written.
3. Set a decimal value of 01 to 31 to this register in BCD code. If a value outside this range is tried to be written in the register, the value is ignored.
4. The RTC_DAY register counts as follows
   A. 01 to 31 (January, March, May, July, August, October, December)
   B. 01 to 30 (April, June, September, November)
   C. 01 to 29 (February of leap year)
   D. 01 to 28 (February of normal year)

9.4.8 RTC_WEEK: RTCC Week Counter Register

The RTC_WEEK register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_WEEK=0x4000_521C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3	2 1 0
Reserved				RWEEK[2:0]
-				0
-				RW

2 0	RWEEK[2:0]	RTCC Week Counter.	
		000	Sunday
		001	Monday
		010	Tuesday
		011	Wednesday
		100	Thursday
		101	Friday
		110	Saturday
		other	Other values are ignored. Value has not changed.

NOTES:

1. The RTC_WEEK register has a value of 0 to 6 (decimal) and indicates the count value of weekdays. It counts up in synchronization with the RTCC day counter register.
2. Set a decimal value of 00 to 06 to this register in BCD code. If a value outside this range is tried to be written in the register, the value is ignored.
3. Values corresponding to the month count register and day count register are not automatically stored to the RTCC week counter register.

9.4.9 RTC_MONTH: RTCC Month Counter Register

The RTC_MONTH register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_MONTH=0x4000_5220

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5	4 3 2 1 0
Reserved				RMONTH[4:0]
-				1
-				RW

4 0	RMONTH[4:0]	RTCC Month Counter.

NOTES:

1. The RTC_MONTH register has a value of 1 to 12 (decimal) and indicates the count value of months. It counts up when the RTCC day counter register overflows.
2. Even if the RTCC day counter register overflows while this register is being written, this register ignores the overflow and is set to the value written.
3. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside this range is tried to be written in the register, the value is ignored.

9.4.10 RTC_YEAR: RTCC Year Counter Register

The RTC_YEAR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_YEAR=0x4000_5224

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RYEAR[7:0]
-			0
-			RW

7 0	RYEAR[7:0]	RTCC Year Counter.

NOTES:

1. The RTC_YEAR register has a value of 0 to 99 (decimal) and indicates the count value of years. It counts up when the RTCC month counter register overflows. Values 00, 04, 08, …, 92, and 96 indicate a leap year.
2. Even if the RTCC month counter register overflows while this register is being written, this register ignores the overflow and is set to the value written.
3. Set a decimal value of 00 to 99 to this register in BCD code. If a value outside this range is tried to be written in the register, the value is ignored.

9.4.11 RTC_ALMIN: RTCC Alarm Minute Counter Register

The RTC_ALMIN register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_ALMIN=0x4000_5228

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			AMIN[6:0]
-			0
-			RW

6 0	AMIN[6:0]	RTCC Alarm Minute Counter.

NOTE:

1. This register is used to set minutes of alarm. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is tried to be written in the register, the value is ignored.

9.4.12 RTC_ALHOUR: RTCC Alarm Hour Counter Register

The RTC_ALHOUR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_ ALHOUR=0x4000_522C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4 3 2 1 0
Reserved				AHOUR[5:0]
-				0x12
-				RW

5 0	AHOUR[5:0]	RTCC Alarm Hour Counter.

NOTES:

1. This register is used to set hours of alarm. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using HS24 bit of RTC_CR(RTCC control register). If a value outside this range is tried to be written in the register, the value is ignored.
2. AHOUR5 bit of RTC_ALHOUR indicates AM(0)/PM(1) if HS24 = 0 (if the 12-hour system is selected).

9.4.13 RTC_ALWEEK: RTCC Alarm Week Counter Register

The RTC_ALWEEK register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

RTC_ALWEEK=0x4000_5230

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved				AWEEK6	AWEEK5	AWEEK4	AWEEK3	AWEEK2	AWEEK1	AWEEK0
-				0	0	0	0	0	0	0
-				RW	RW	RW	RW	RW	RW	RW

6	AWEEK6	Saturday Alarm Setting	
		0	Disable Saturday Alarm
		1	Enable Saturday Alarm
5	AWEEK5	Friday Alarm Setting	
		0	Disable Friday Alarm
		1	Enable Friday Alarm
4	AWEEK4	Thursday Alarm Setting	
		0	Disable Thursday Alarm
		1	Enable Thursday Alarm
3	AWEEK3	Wednesday Alarm Setting	
		0	Disable Wednesday Alarm
		1	Enable Wednesday Alarm
2	AWEEK2	Tuesday Alarm Setting	
		0	Disable Tuesday Alarm
		1	Enable Tuesday Alarm
1	AWEEK1	Monday Alarm Setting	
		0	Disable Monday Alarm
		1	Enable Monday Alarm
0	AWEEK0	Sunday Alarm Setting	
		0	Disable Sunday Alarm
		1	Enable Sunday Alarm

9.4.14 RTCC Register Map Summary

Table 49. RTCC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	RTC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RTEN	RTIN[2:0]			RTIFLAG	HS24	Res	OUTSEL	ALEN	ALIEN	ALIFLAG	Res	Res	Res	RTWST	RTWAIT
	Reset value																	0	0	0	0	0	0		0	0	0	0				0	0
0x04	RTC_ECR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ECTM	ECSIGN	ECV[5:0]					
	Reset value																									0	0	0	0	0	0	0	0
0x08	RTC_SCNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RTCNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	RTC_SEC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RSEC[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x10	RTC_MIN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RMIN[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x14	RTC_HOUR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RHOUR[5:0]					
	Reset value																											0	1	0	0	1	0
0x18	RTC_DAY	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RDAY[5:0]					
	Reset value																											0	0	0	0	0	1
0x1C	RTC_WEEK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RWEEK[2:0]		
	Reset value																														0	0	0
0x20	RTC_MONTH	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RMONTH[4:0]				
	Reset value																												0	0	0	0	1
0x24	RTC_YEAR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RYEAR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x28	RTC_ALMIN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AMIN[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x2C	RTC_ALHOUR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AHOUR[5:0]					
	Reset value																											0	1	0	0	1	0
0x30	RTC_ALWEEK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AWEEK6	AWEEK5	AWEEK4	AWEEK3	AWEEK2	AWEEK1	AWEEK0
	Reset value																										0	0	0	0	0	0	0

10. Timer Counter 10/11/12/13/14/15

10.1 Timer Counter 10/11/12/13/14/15 Introduction

The timer block comprises six channels of 16-bit general purpose timers. Each has an independent 16-bit counter and a dedicated prescaler that feeds the counting clock. They support periodic timer, PWM pulse, one-shot and capture mode.

One more optional free-run timer is provided. The main purpose of this timer is a periodical tick timer or a wake-up source.

10.2 Timer Counter 10/11/12/13/14/15 Main Features

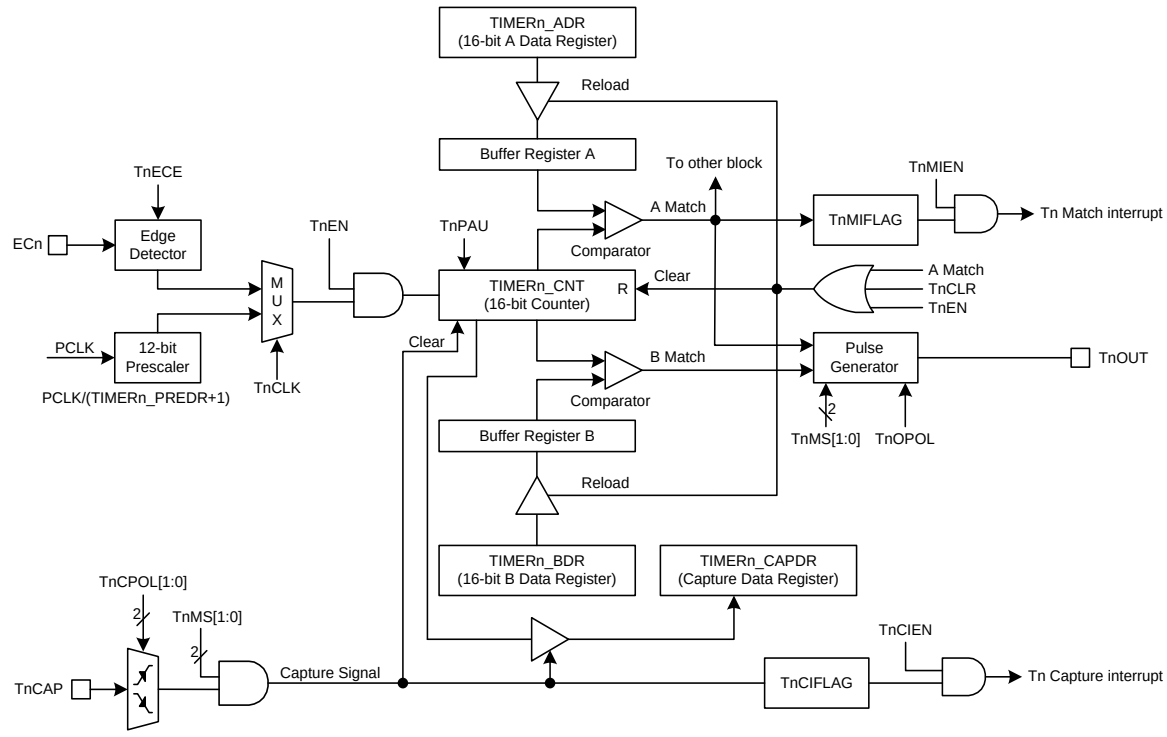
Timer counter 10/11/12/13/14/15 includes the features below:

- 16-bit up-counter and 12-bit prescaler
- Four operating modes:
  - Periodic Mode
  - One-shot Mode
  - PWM Mode
  - Capture Mode
- Various interrupts:
  - Match interrupt
  - Capture interrupt
- Synchronous start and clear function

10.3 Timer Counter 10/11/12/13/14/15 Functional Description

10.3.1 Block Diagram

Figure 37. Timer Counter n Timer Block Diagram (n = 10, 11, 12, 13, 14 and 15)



NOTES:

1. TnEN bit is automatically cleared to logic '0' after one pulse is generated at a PPG one-shot mode.
2. After the TnEN bit is set, counting of TIMERn_CNT may be delayed 2 clocks or more.

10.4 Pin Description for Timer Counter 10/11/12/13/14/15

Table 50. Pins and External Signals for Timer Counter n (n = 10, 11, 12, 13, 14 and 15)

Pin Name	Type	Description
ECn	I	External clock input
TnCAP	I	Capture input
TnOUT	O	PWM/one-shot output

10.5 Functional Description

10.5.1 Timer Counter 10/11/12/13/14/15

Timer/counter n can use an internal or an external clock source (ECn). A clock selection logic can select a clock source, and it is controlled by clock selection bits (TnCLK).

- Timer n clock source: {PCLK/(TIMERn_PREDR+1)}, ECn

In Capture mode, by TnCAP, data is captured into a corresponding capture data register (TIMERn_CAPDR). Timer n outputs the comparison result between counter and data register through TnOUT port in Timer/counter mode. Also, timer n output PWM waveform through TnOUT port under PPG mode. (n = 10, 11, 12, 13, 14 and 15)

Table 51. Timer n Operating Modes (n = 10, 11, 12, 13, 14 and 15)

TnEN	Alternate Mode	TnMS	TIMERn_PREDR	Timer n
1	TIMER10: T10OUT TIMER11: T11OUT TIMER12: T12OUT TIMER13: T13OUT TIMER14: T14OUT TIMER15: T15OUT	00	0xXXX	Timer/Counter Mode
1	TIMER10: T10CAP TIMER11: T11CAP TIMER12: T12CAP TIMER13: T13CAP TIMER14: T14CAP TIMER15: T15CAP	01	0xXXX	Capture Mode
1	TIMER10: T10OUT TIMER11: T11OUT TIMER12: T12OUT TIMER13: T13OUT TIMER14: T14OUT TIMER15: T15OUT	10	0xXXX	PPG Mode (one-shot mode)
1		11	0xXXX	PPG Mode (repeat mode)

10.5.2 16-bit Timer/Counter Mode

16-bit Timer/counter mode is selected by control register as shown in Figure 38. The 16-bit timer has a counter register and a data register. The counter register is increased by internal or external clock input. Timer n can use an input clock with 12-bit prescaler division rates (TIMERn_PREDR) and an external clock (ECn). When the values of TIMERn_CNT and TIMERn_ADR are the same in the timer n, a match signal is generated, and the interrupt of Timer n takes place.

The TIMERn_CNT values are automatically cleared by the match signal. It can also be cleared by software (TnCLR).

Figure 38. 16-Bit Timer/Counter Mode for Timer n (n = 10, 11, 12, 13, 14 and 15)

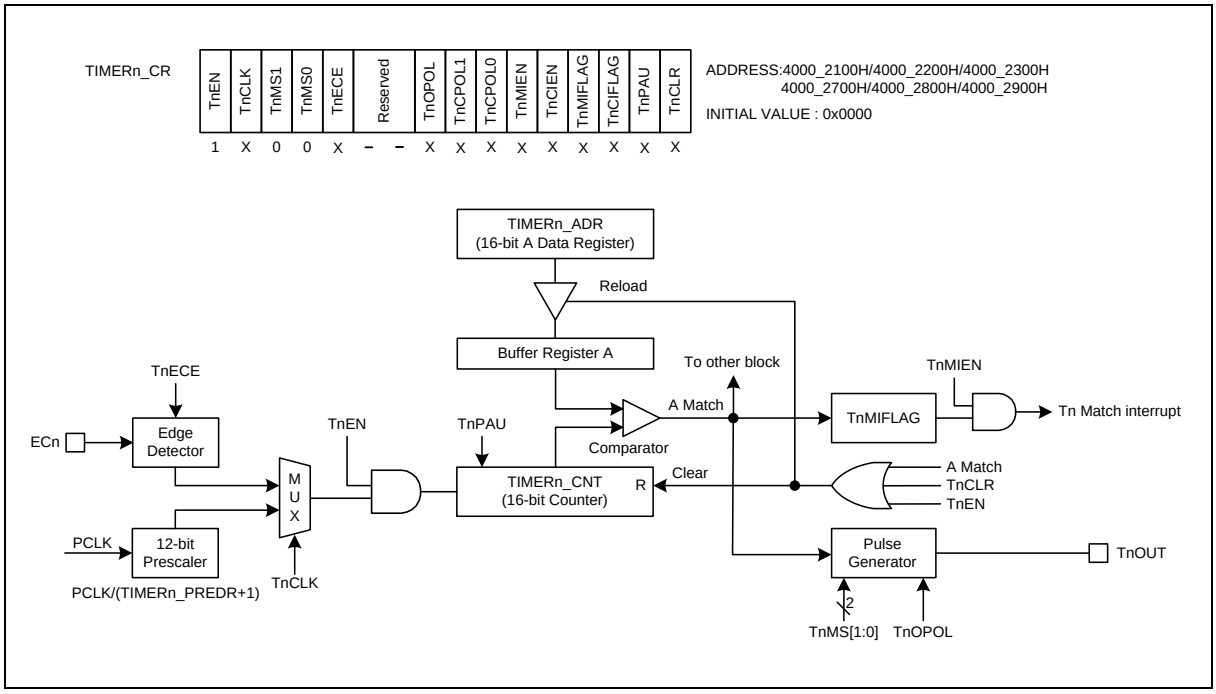
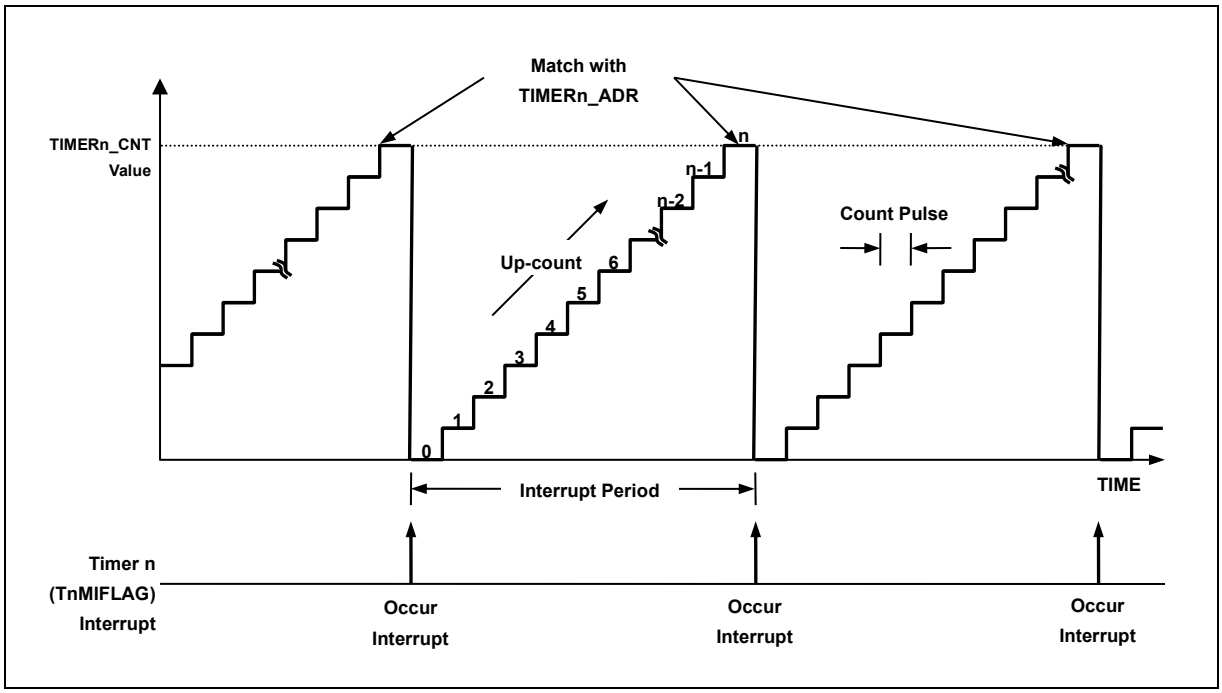


Figure 39. 16-Bit Timer/Counter n Example (n = 10, 11, 12, 13, 14 and 15)



10.5.3 16-bit Capture Mode

Timer n Capture mode is evoked by configuring TnMS[1:0] as '01'. The internal clock can be used as a clock source. It basically has the same function as the 16-bit timer/counter mode and an interrupt takes place when TIMERn_CNT becomes equal to TIMERn_ADR.

This timer interrupt in Capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. The capture result is loaded into TIMERn_CAPDR. In the timer n capture mode, timer n output (TnOUT) waveform is not available.

Figure 40. 16-Bit Capture Mode for Timer n (n = 10, 11, 12, 13, 14 and 15)

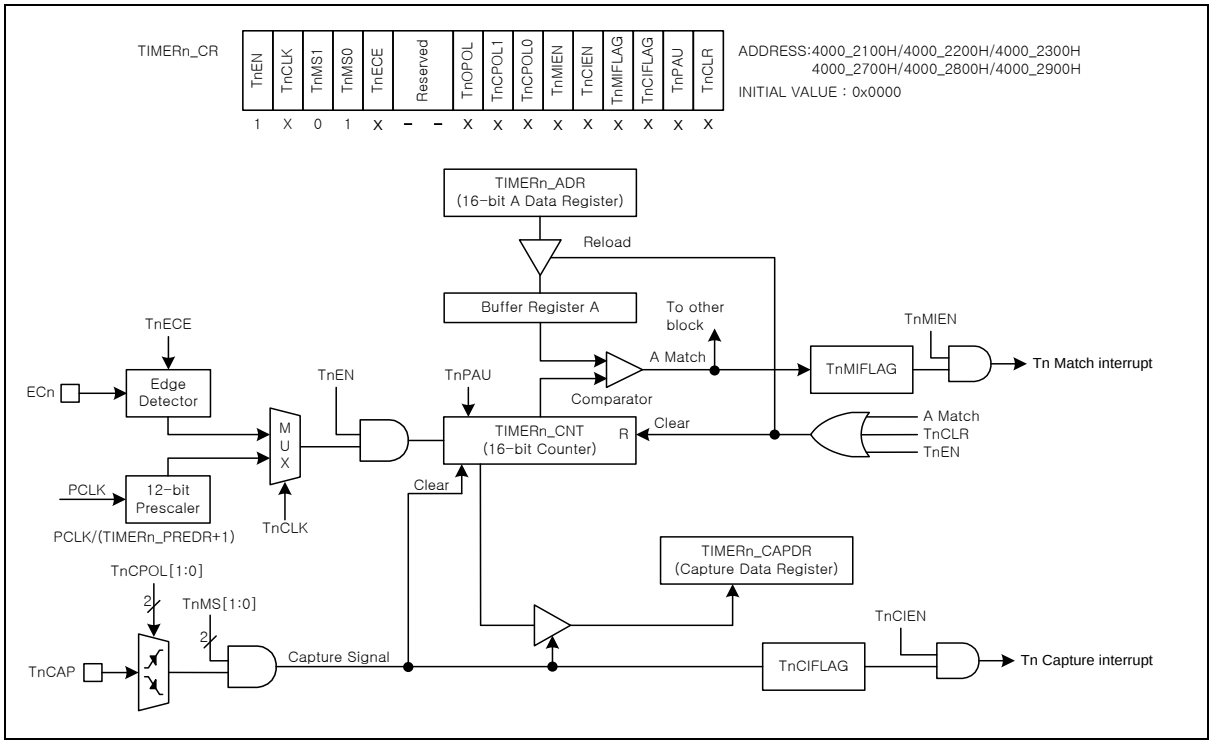


Figure 41. 16-Bit Capture Mode for Timer n (n = 10, 11, 12, 13, 14 and 15)

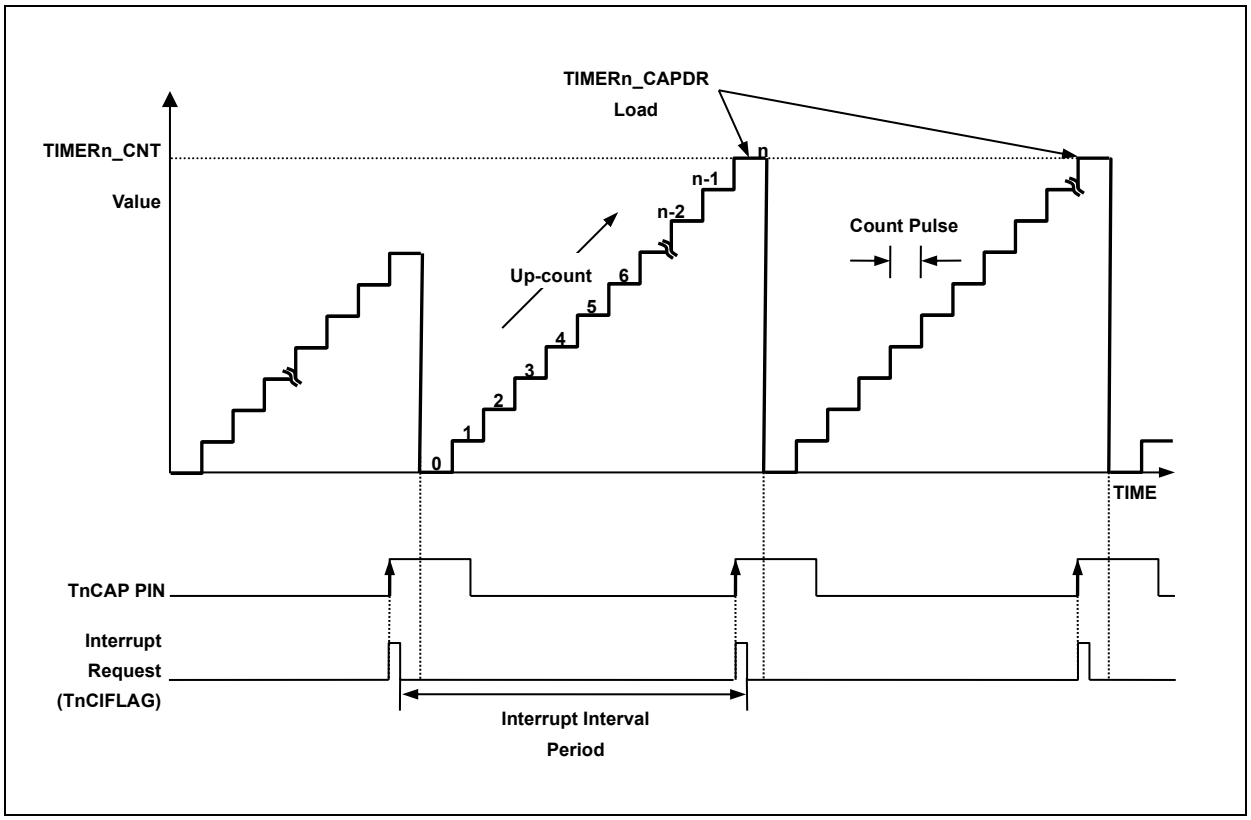
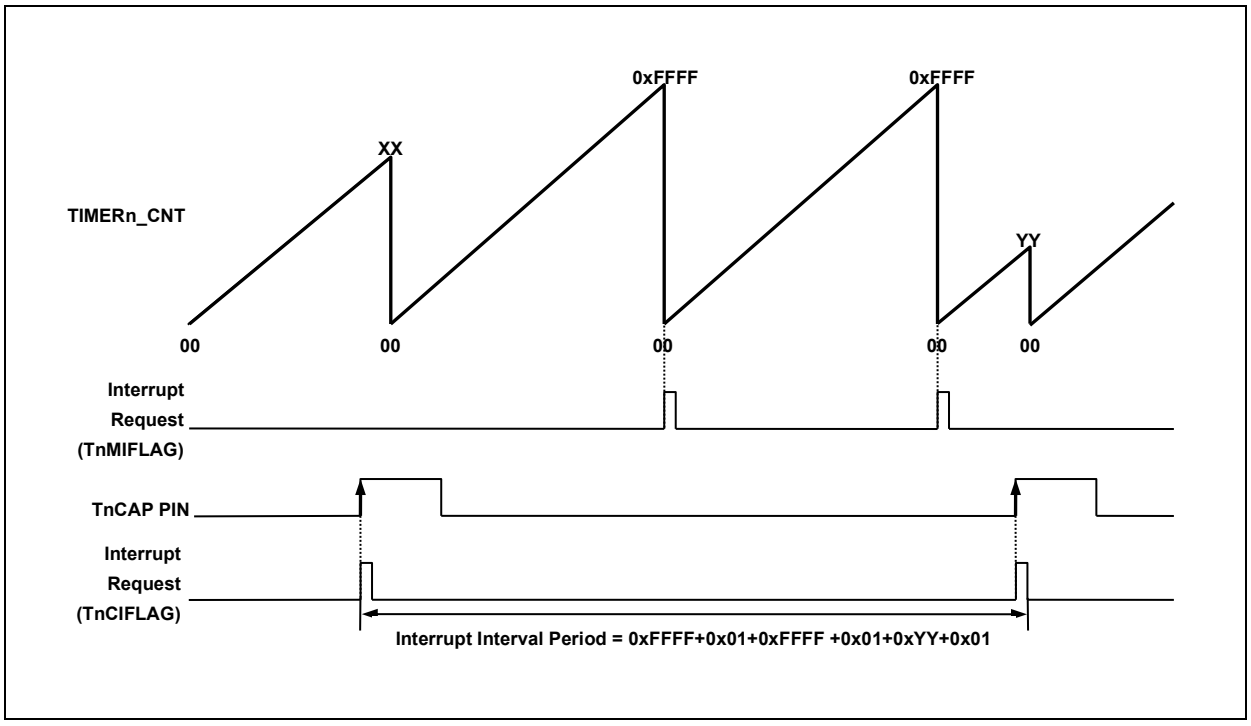


Figure 42. Express Timer Overflow in Capture Mode (n = 10, 11, 12, 13, 14 and 15)



10.5.4 16-bit PPG Mode

Timer n has a PPG (Programmable Pulse Generation) function. In PPG mode, the TnOUT pin generates PWM output of up to 16-bit resolution. This pin should be configured as a PWM output by setting Px_AFSR1, Px_AFSR2 to ‘AF0’. The period of PWM output is determined by the TIMERn_ADR. The duty of PWM output is determined by TIMERn_BDR. (x = A to F)

Figure 43. 16-Bit PPG Repeat and One-Shot Mode for Timer n (n = 10, 11, 12, 13, 14 and 15)

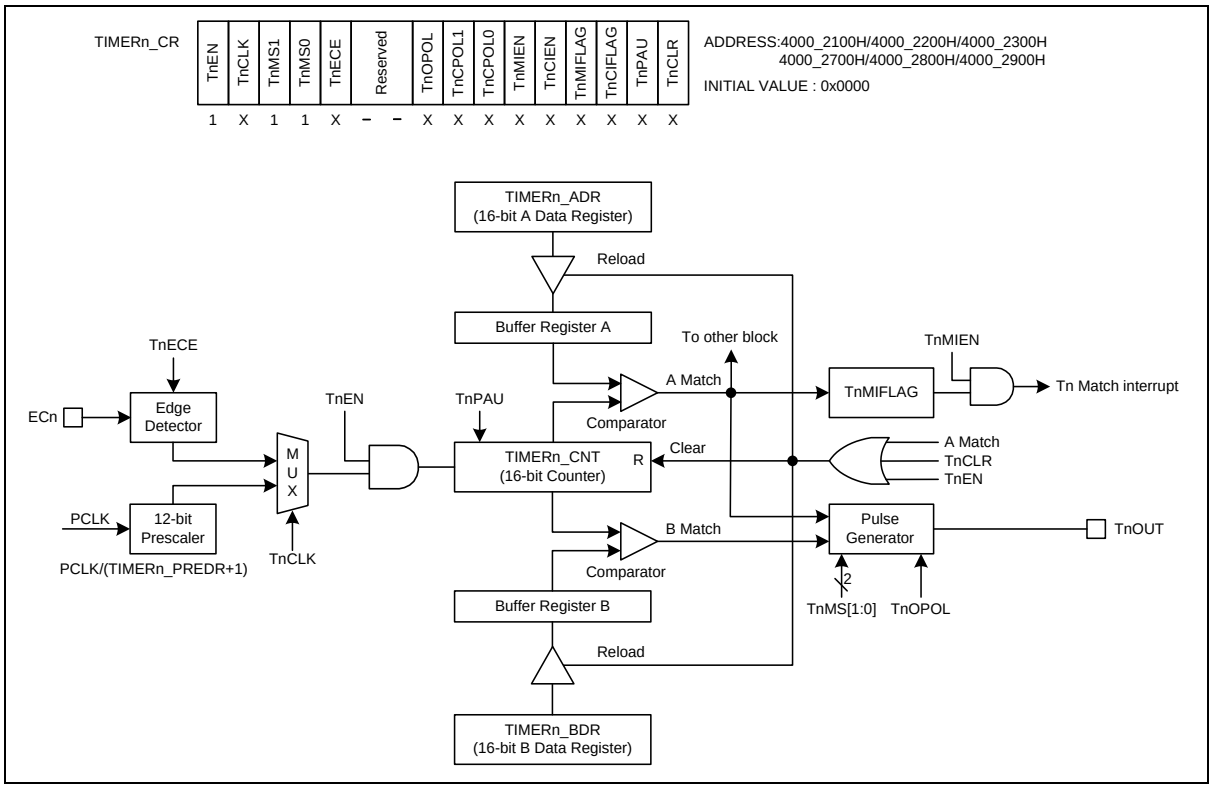
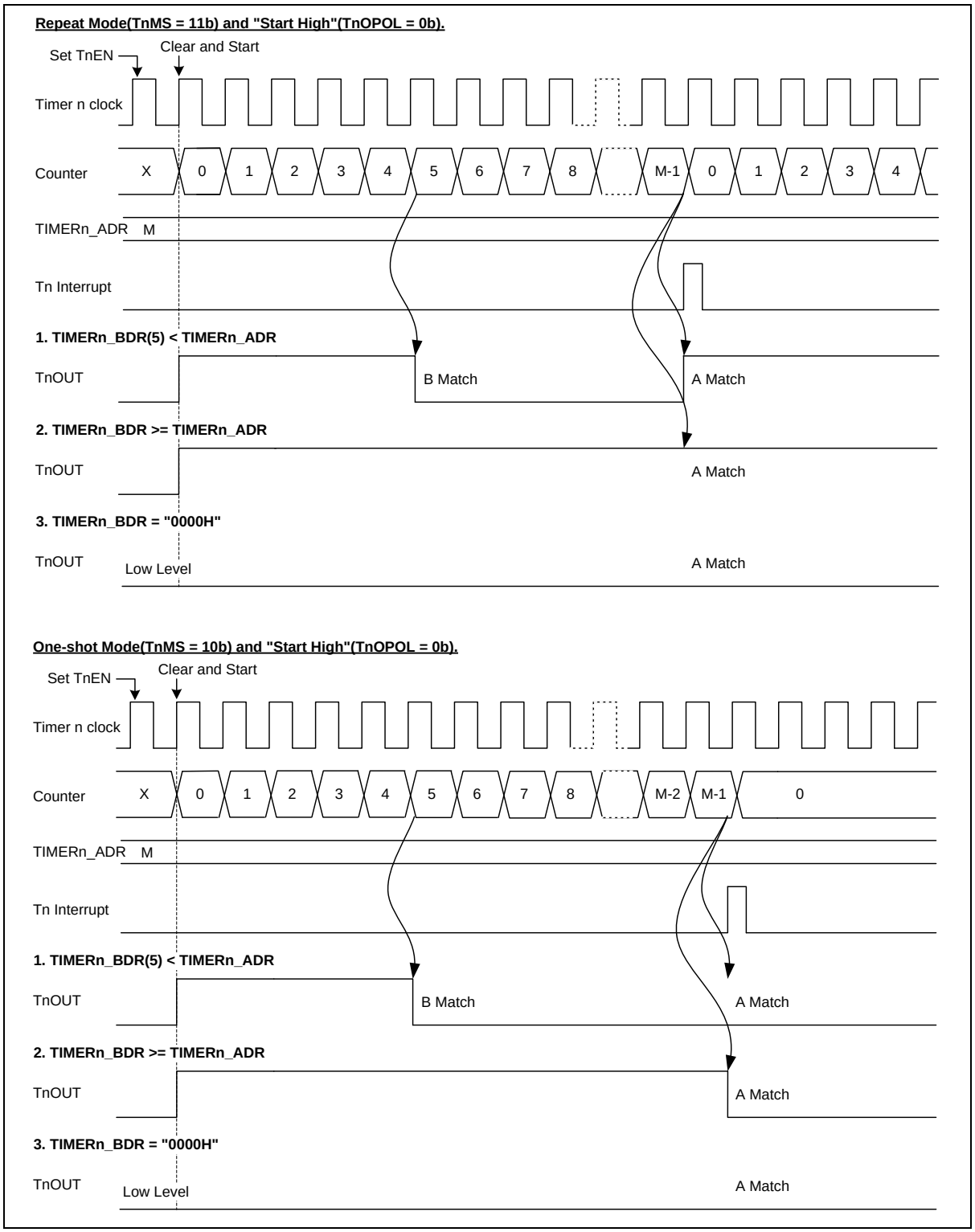


Figure 44. 16-Bit PPG Mode Timing Chart for Timer n (n = 10, 11, 12, 13, 14 and 15)



10.6 TIMER10/11/12/13/14/15 Registers

The base address of Timer 10/11/12/13/14/15 is described in the following:

Table 52. Base Address of Timer 10/11/12/13/14/15

Name	Base Address
TIMER10	0x4000_2100
TIMER11	0x4000_2200
TIMER12	0x4000_2300
TIMER13	0x4000_2700
TIMER14	0x4000_2800
TIMER15	0x4000_2900

Table 53. TIMER Register Map

Name	Offset	Type	Description	Reset Value	Reference
TIMERn_CR	0x00	RW	Timer/Counter n Control Register	0x0000_0000	10.6.1
TIMERn_ADR	0x04	RW	Timer/Counter n A Data Register	0x0000_FFFF	10.6.2
TIMERn_BDR	0x08	RW	Timer/Counter n B Data Register	0x0000_FFFF	10.6.3
TIMERn_CAPDR	0x0C	RO	Timer/Counter n Capture Data Register	0x0000_0000	10.6.4
TIMERn_PREDR	0x10	RW	Timer/Counter n Prescaler Data Register	0x0000_0FFF	10.6.5
TIMERn_CNT	0x14	RO	Timer/Counter n Counter Register	0x0000_0000	10.6.6

NOTE:

1. n = 10, 11, 12, 13, 14, and 15.

10.6.1 TIMERn_CR: Timer /Counter n Control Register

The timer module should be configured properly before running. The timer should be configured with the appropriate value in TIMERn_CR register for designated operating mode. After configuring this register, a user can start or stop the timer function by using this register.

TIMERn_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

(n = 10, 11, 12, 13, 14 and 15).

TIMER10_CR=0x4000_2100, TIMER11_CR=0x4000_2200, TIMER12_CR=0x4000_2300
TIMER13_CR=0x4000_2700, TIMER14_CR=0x4000_2800, TIMER15_CR=0x4000_2900

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12	11	10	9	8	7 6	5	4	3	2	1	0
Reserved		TnEN	TnCLK	TnMS[1:0]	TnECE	Reserved		TnOPOL	TnCPOL[1:0]	TnMIEN	TnCIEN	TnMIFLAG	TnCIFLAG	TnPAU	TnCLR
-		0	0	00	0	-		0	00	0	0	0	0	0	0
-		RW	RW	RW	RW	-		RW	RW	RW	RW	RW	RW	RW	RW

15	TnEN	Timer n Operation Enable.
	0	Disable timer n operation.
	1	Enable timer n operation. (Counter clear and start)
14	TnCLK	Timer n Clock Selection.
	0	Select an internal prescaler clock.
	1	Select an external clock.
	NOTE: 1. This bit should be changed while TnEN bit is '0'.	
13 12	TnMS[1:0]	Timer n Operation Mode Selection.
	00	Timer/Counter mode. (TnOUT: toggle at A-match)
	01	Capture mode. (The A-match interrupt can occur)
	10	PPG one-shot mode. (TnOUT: Programmable pulse output)
	11	PPG repeat mode. (TnOUT: Programmable pulse output)
	NOTE: 1. This bit should be changed while TnEN bit is '0'.	
11	TnECE	Timer n External Clock Edge Selection.
	0	Select falling edge of external clock.
	1	Select rising edge of external clock.
8	TnOPOL	TnOUT Polarity Selection.
	0	Start high. (TnOUT is low level at disable)
	1	Start low. (TnOUT is high level at disable)
7 6	TnCPOL[1:0]	Timer n Capture Polarity Selection.
	00	Capture on falling edge.
	01	Capture on rising edge.
	10	Capture on both falling and rising edge.
	11	Reserved.
5	TnMIEN	Timer n Match Interrupt Enable.
	0	Disable timer n match interrupt.
	1	Enable timer n match interrupt.
4	TnCIEN	Timer n Capture Interrupt Enable.

		0	Disable timer n capture interrupt.
		1	Enable timer n capture interrupt.
3	TnMIFLAG	Timer n Match Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
2	TnCIFLAG	Timer n Capture Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
1	TnPAU	Timer n Counter Temporary Pause Control.	
		0	Continue counting.
		1	Temporary pause.
0	TnCLR	Timer n Counter and Prescaler Clear.	
		0	No effect.
		1	Clear timer n counter and prescaler. (Automatically cleared to ‘0’ after operation)

10.6.2 TIMERn_ADR: Timer/Counter n A Data Register

TIMERn_ADR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, 13, 14 and 15).

TIMER10_ADR=0x4000_2104, TIMER11_ADR=0x4000_2204, TIMER12_ADR=0x4000_2304
TIMER13_ADR=0x4000_2704, TIMER14_ADR=0x4000_2804, TIMER15_ADR=0x4000_2904

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		ADATA[15:0]	
-		0xFFFF	
-		RW	

15 0	ADATA[15:0]	Timer/Counter n A Data. The range is 0x0002 to 0xFFFF.

NOTE:

1. Do not write “0x0000” in the TIMERn_ADR register under PPG mode.

10.6.3 TIMERn_BDR: Timer/Counter n B Data Register

TIMERn_BDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, 13, 14 and 15).

TIMER10_BDR=0x4000_2108, TIMER11_BDR=0x4000_2208, TIMER12_BDR=0x4000_2308
TIMER13_BDR=0x4000_2708, TIMER14_BDR=0x4000_2808, TIMER15_BDR=0x4000_2908

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		BDATA[15:0]	
-		0xFFFF	
-		RW	

15 0	BDATA[15:0]	Timer/Counter n B Data. The range is 0x0000 to 0xFFFF.

10.6.4 TIMERn_CAPDR: Timer/Counter n Capture Data Register

TIMERn_CAPDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, 13, 14 and 15).

TIMER10_CAPDR=0x4000_210C, TIMER11_CAPDR=0x4000_220C, TIMER12_CAPDR=0x4000_230C
TIMER13_CAPDR=0x4000_270C, TIMER14_CAPDR=0x4000_280C, TIMER15_CAPDR=0x4000_290C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		CAPD[15:0]	
-		0x0000	
-		RO	

15 0	CAPD[15:0]	Timer/Counter n Capture Data.

10.6.5 TIMERn_PREDR: Timer/Counter n Prescaler Data Register

TIMERn_PREDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, 13, 14 and 15).

TIMER10_PREDR=0x4000_2110, TIMER11_PREDR=0x4000_2210, TIMER12_PREDR=0x4000_2310
TIMER13_PREDR=0x4000_2710, TIMER14_PREDR=0x4000_2810, TIMER15_PREDR=0x4000_2910

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		PRED[11:0]	
-		0x0FFF	
-		RW	

11 0	PRED[11:0]	Timer/Counter n Prescaler Data.

10.6.6 TIMERn_CNT: Timer/Counter n Counter Register

TIMERn_CNT register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, 13, 14 and 15).

TIMER10_CNT=0x4000_2114, TIMER11_CNT=0x4000_2214, TIMER12_CNT=0x4000_2314
TIMER13_CNT=0x4000_2714, TIMER14_CNT=0x4000_2814, TIMER15_CNT=0x4000_2914

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		CNT[15:0]	
-		0x0000	
-		R0	

15 0	CNT[15:0]	Timer/Counter n Counter.

10.6.7 TIMERn Register Map Summary

Table 54. TIMERn Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMERn_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TnEN	TnCLK	TnMS[1:0]		TnECE	Res.	Res.	TnOPOL	TnCPOL[1:0]		TnMIEN	TnCIEN	TnMIFLAG	TnCIFLAG	TnPAU	TnCLR
	Reset value																	0	0	0	0	0			0	0	0	0	0	0	0	0	0
0x04	TIMERn_ADR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADATA[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	TIMERn_BDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDATA[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	TIMERn_CAPDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAPD[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	TIMERn_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	TIMERn_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOTE:

1. n = 10, 11, 12, 13, 14, and 15.

11. Timer Counter 20/21

11.1 Timer Counter 20/21 Introduction

The timer block comprises two channels 32-bit general purpose timers. Each timer has an independent 32-bit counter and a dedicated prescaler that feeds counting clock. It supports periodic timer, PWM pulse, one-shot timer and capture mode.

One more optional free-run timer is provided. Main purpose of this timer is a periodical tick timer or a wake-up source. The Timer counter 20/21 features the following.

11.2 Timer Counter 20/21 Main Features

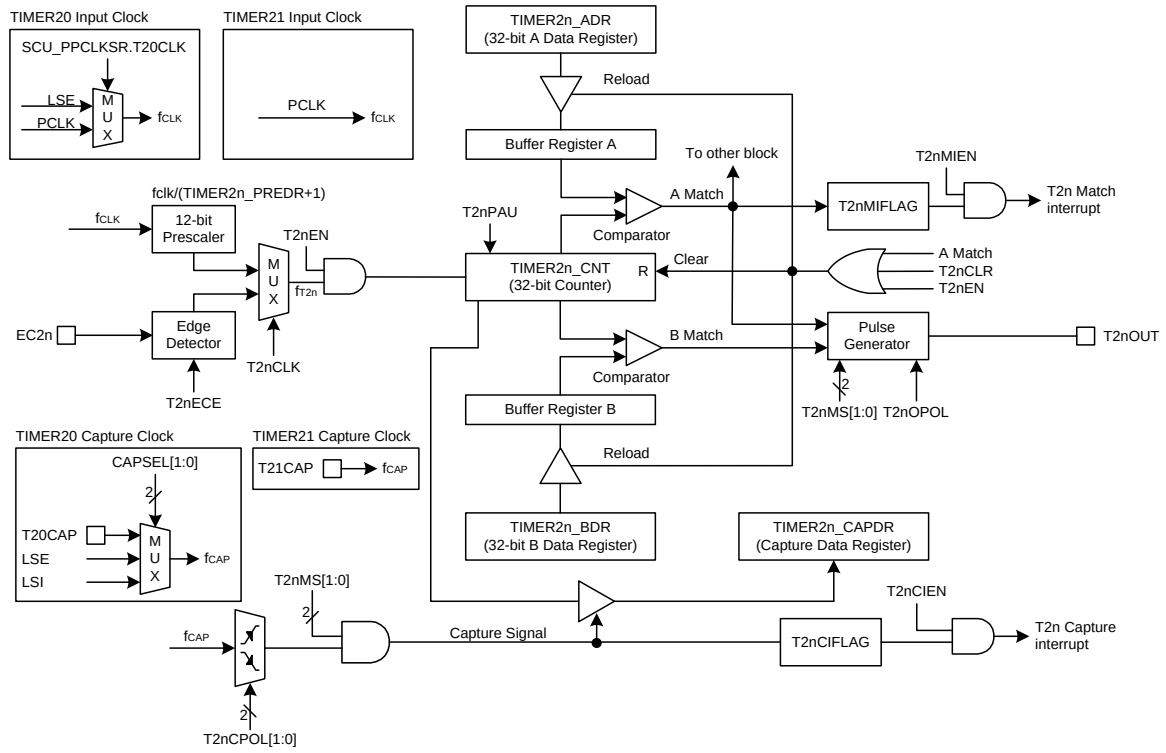
Timer counter 20/21 includes the features below:

- 32-bit up-counter and 12-bit prescaler
- Four operating modes:
  - Periodic Mode
  - One-shot Mode
  - PWM Mode
  - Capture Mode
- Various interrupts:
  - Match interrupt
  - Capture interrupt
- Synchronous start and clear function

11.3 Timer Counter 20/21 Functional Description

11.3.1 Block Diagram

Figure 45. Timer Counter 2n Timer Block Diagram (n = 0, 1)



NOTES:

1. T2nEN is automatically cleared to logic '0' after one pulse is generated in PPG one-shot mode.
2. After T2nEN bit is set, counting of TIMER2n_CNT may be delayed by two clocks or more.

11.4 Pin Description for Timer Counter 20/21

Table 55. Pins and External Signals for Timer Counter 2n (n = 0, 1)

Pin Name	Type	Description
EC2n	I	External clock input
T2nCAP	I	Capture input
T2nOUT	O	PWM/one-shot output

11.5 Functional Description

11.5.1 Timer Counter 20/21

Timer/Counter 2n (n = 0, 1) can use an internal or an external clock as a clock source (EC2n). A clock selection logic selects the clock source, and the clock selection logic is controlled by clock selection bits (T2nCLK).

- TIMER 2n clock sources are listed as followings (n = 0, 1):
  - PCLK/(TIMER2n_PREDR +1)
  - LSE/(TIMER2n_PREDR +1) (TIMER20 only)
  - EC2n

In capture mode, by T2nCAP, LSE or LSI data (LSE and LSI: TIMER20 only) is captured into input capture data register (TIMER2n_CAPDR). Timer 2n outputs the comparison result between counter and data register through T2nOUT port in Timer/counter mode. In addition, Timer 2n outputs PWM waveform through T2nOUT port in PPG mode.

Table 56. Timer 2n Operating Modes (n = 0, 1)

T2nEN	Alternate Mode	T2nMS	TIMER2n_PREDR	Timer 2n
1	T2nOUT	00	0xXXX	32-bit Timer/Counter Mode
1	T2nCAP	01	0xXXX	32-bit Capture Mode
1	T2nOUT	10	0xXXX	32-bit PPG Mode (one-shot mode)
1	T2nOUT	11	0xXXX	32-bit PPG Mode (repeat mode)

11.5.2 32-bit Timer/Counter Mode

32-bit Timer/counter mode is selected by control register as shown in Figure 46. The 32-bit timer has a counter register and a data register. The counter register is increased by internal or external clock input. Timer 2n can use an input clock with 12-bit prescaler division rates (TIMER2n_PREDR) and an external Clock (EC2n). When the values of TIMER2n_CNT and TIMER2n_ADR are the same in the timer 2n, a match signal is generated and the interrupt of Timer 2n takes place. The TIMER2n_CNT values are automatically cleared by match signal. It can also be cleared by software (T2nCLR) (n = 0, 1).

The TIMER2n_CNT values are automatically cleared by match signal. It can also be cleared by software (T2nCLR).

Figure 46. 32-Bit Timer/Counter Mode for Timer 2n (n = 0, 1)

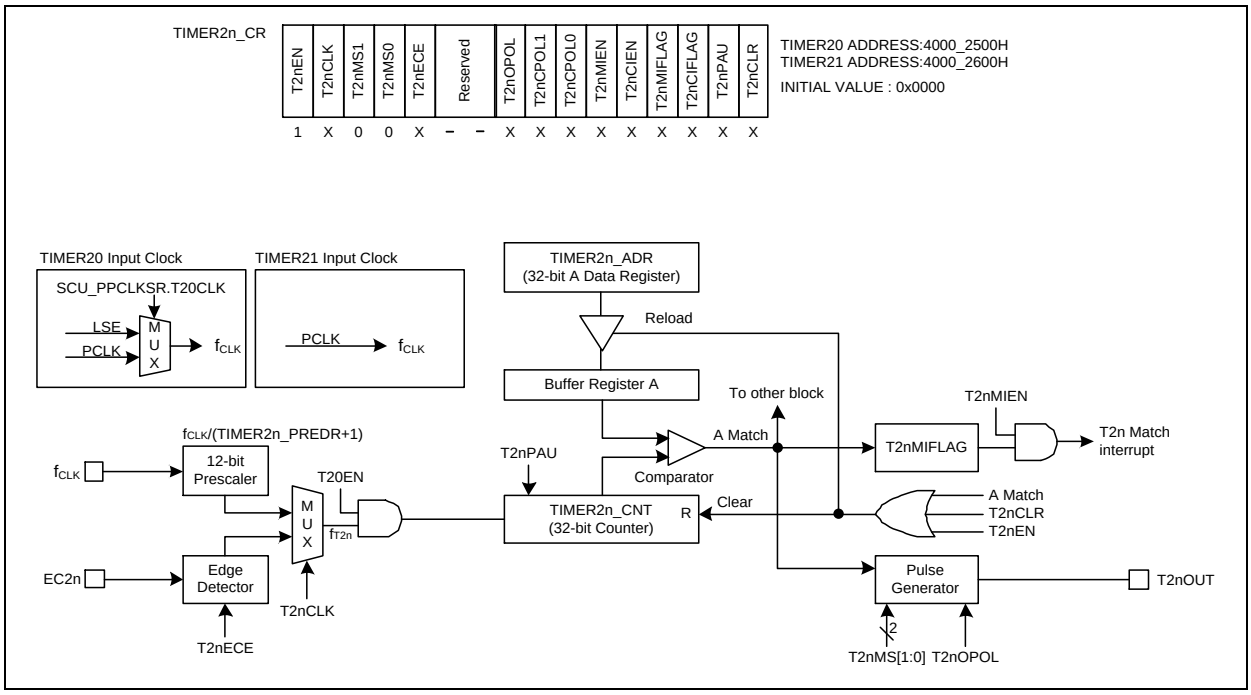
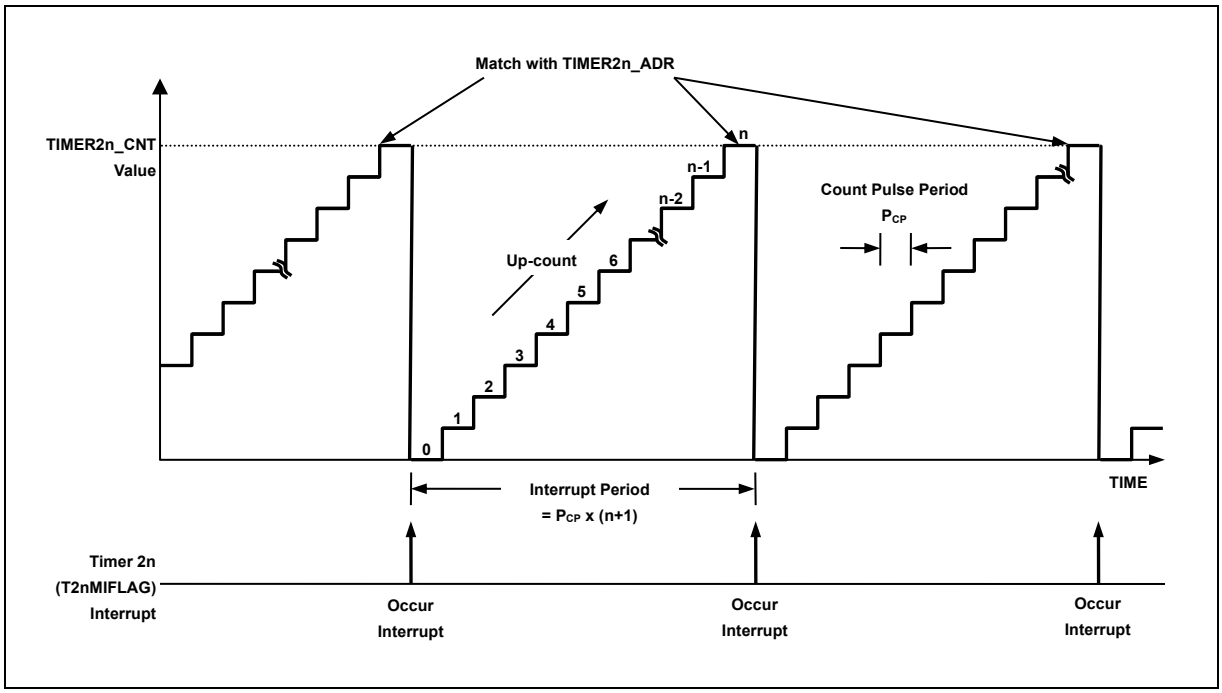


Figure 47. 32-Bit Timer/Counter 2n Example (n = 0, 1)



11.5.3 32-bit Capture Mode

Timer 2n Capture mode is evoked by configuring T2nMS[1:0] as '01'. The internal clock can be used as a clock source. It basically has the same function as the 32-bit timer/counter mode and an interrupt takes place when TIMER2n_CNT becomes equal to TIMER2n_ADR. TIMER2n_CNT values are cleared by software (T2nCLR) (n = 0, 1).

This timer interrupt in Capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer. The capture result is loaded into TIMER2n_CAPDR. In the Timer 2n capture mode, Timer 2n output (T2nOUT) waveform is not available.

Figure 48. 32-Bit Capture Mode for Timer 2n (n = 0, 1)

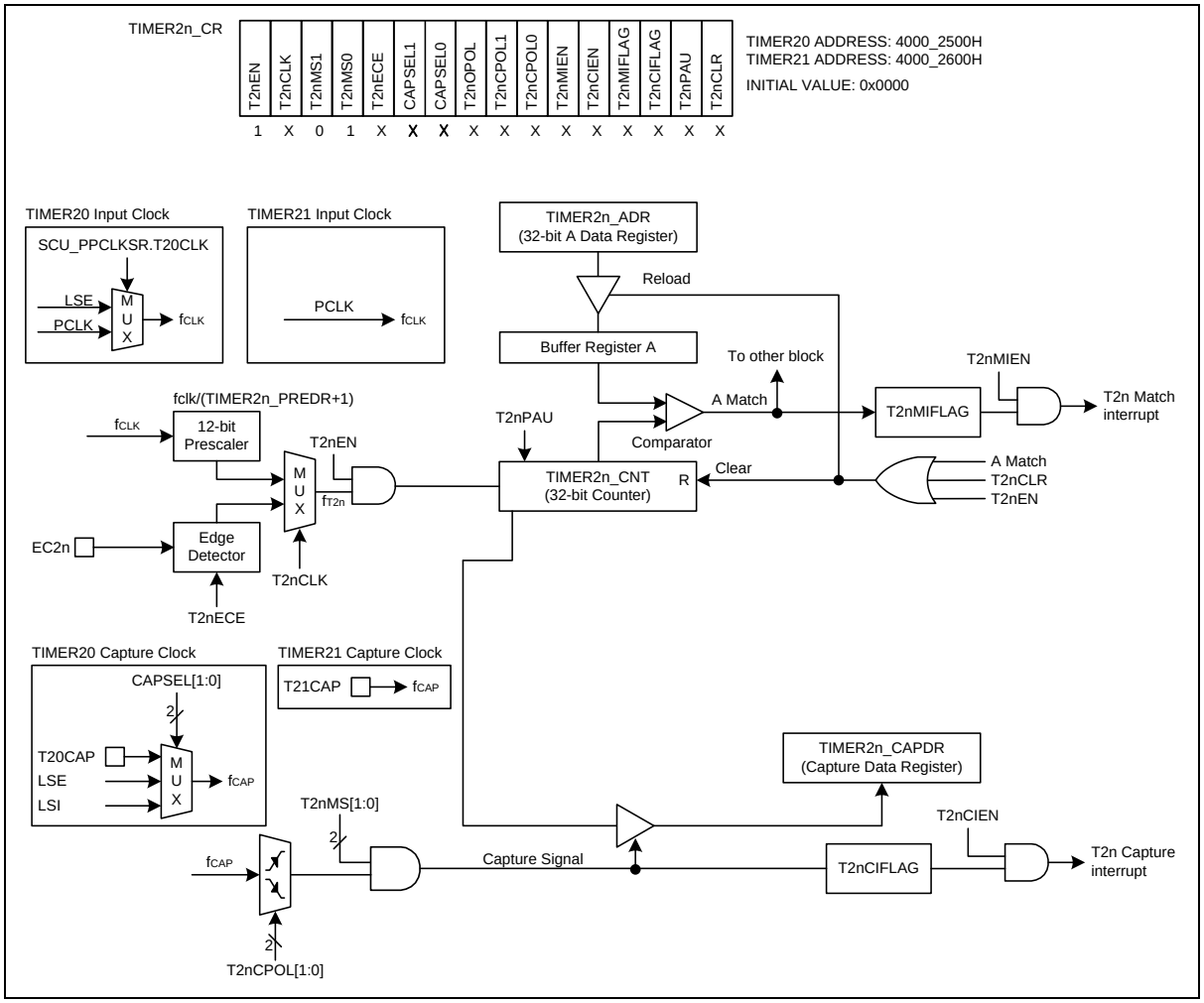


Figure 49. 32-Bit Capture Mode for Timer 2n (n = 0, 1)

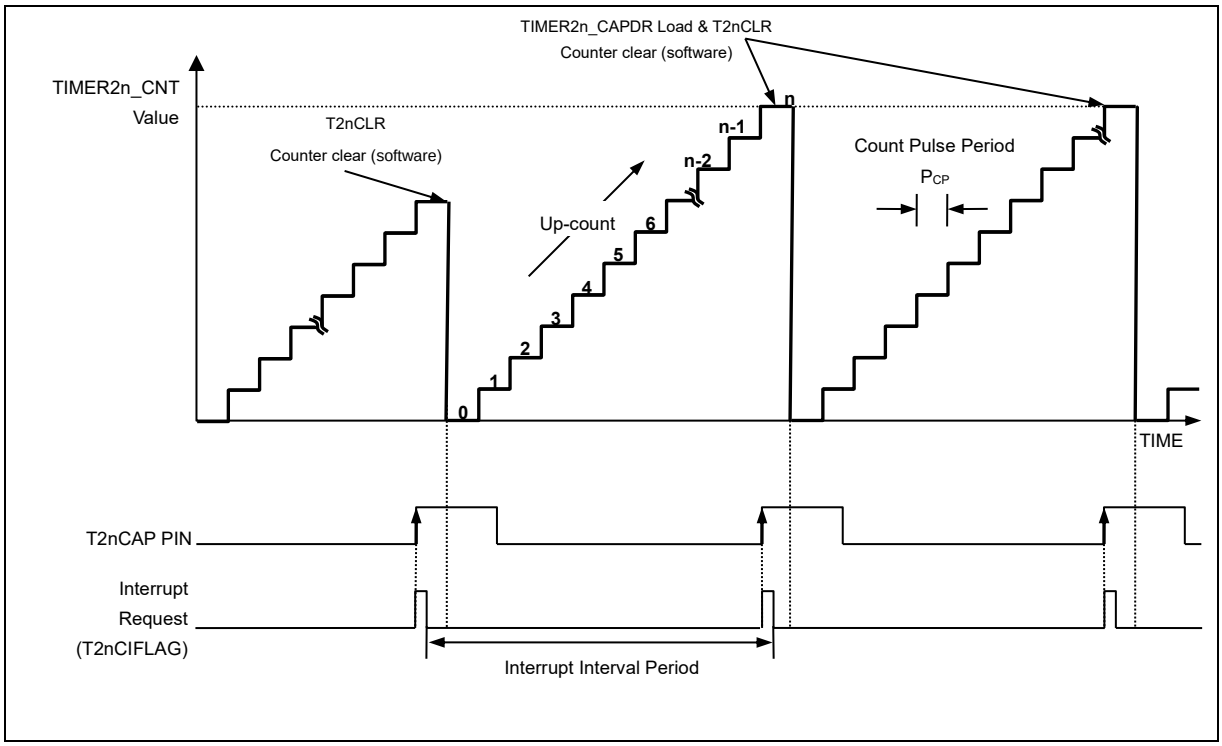
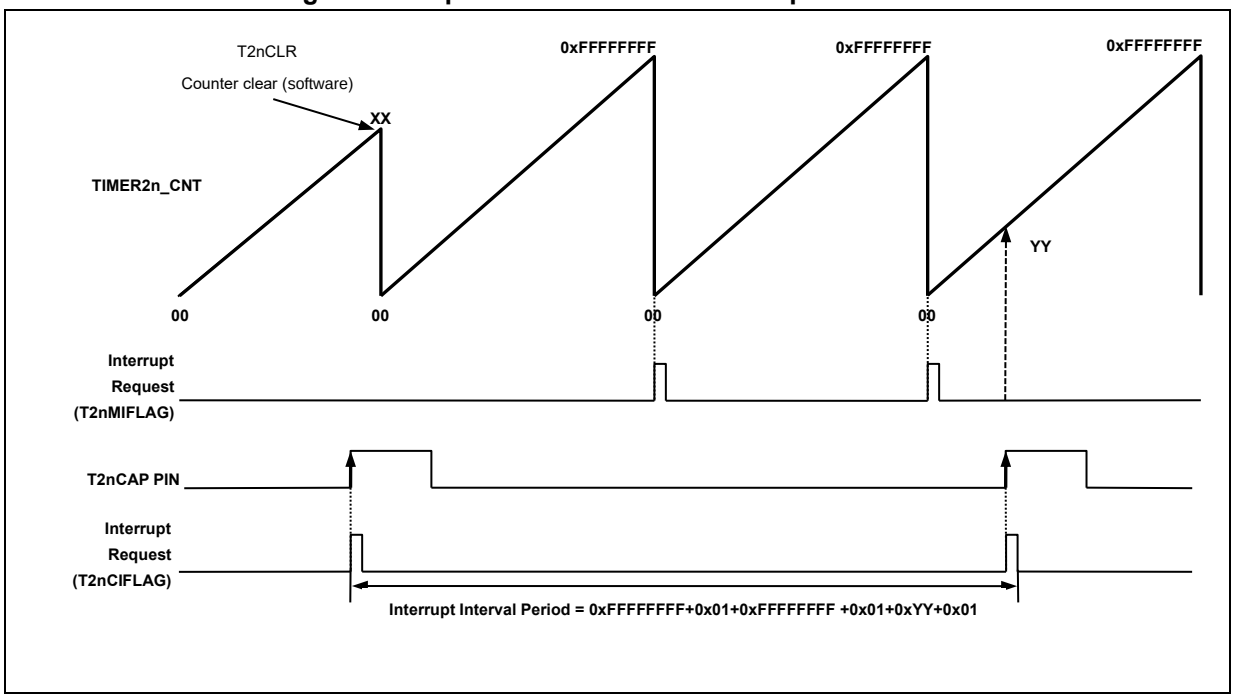


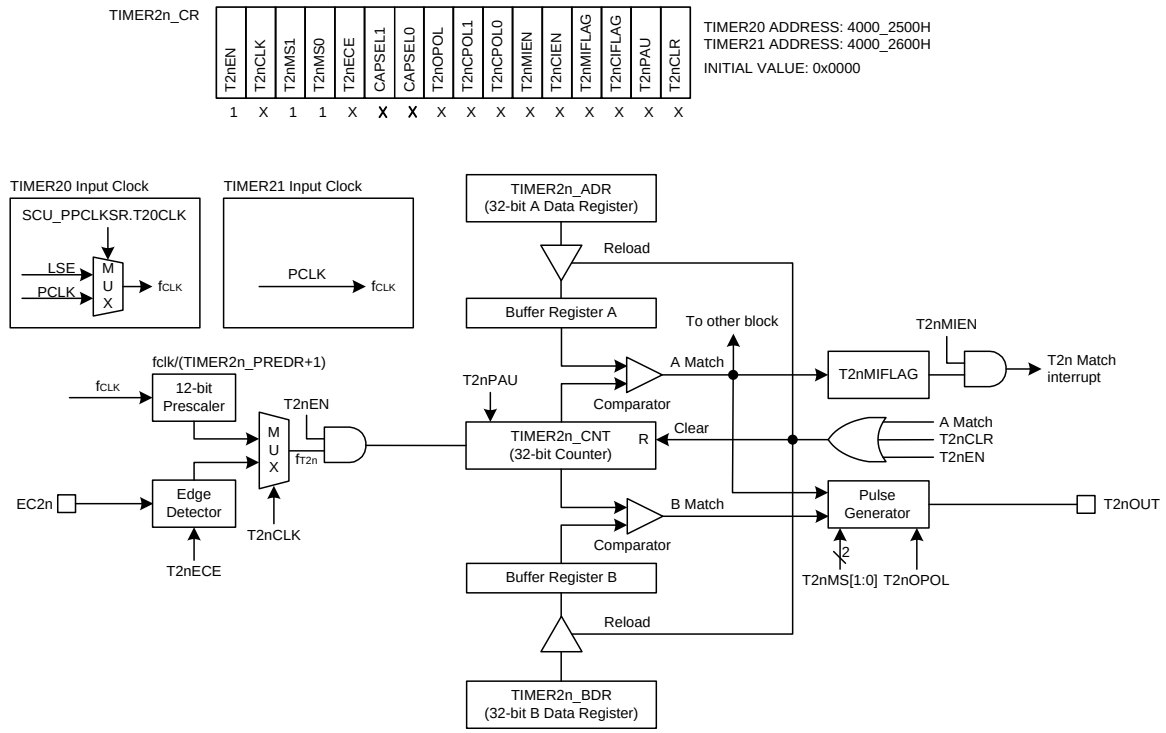
Figure 50. Express Timer Overflow in Capture Mode



11.5.4 32-bit PPG Mode

Timer 2n has a PPG (Programmable Pulse Generation) function. In PPG mode, the T2nOUT pin generates PWM output of up to 32-bit resolution. This pin should be configured as a PWM output by setting port's alternate function selection register. The period of PWM output is determined by the TIMER2n_ADR. The duty of PWM output is determined by TIMER2n_BDR.

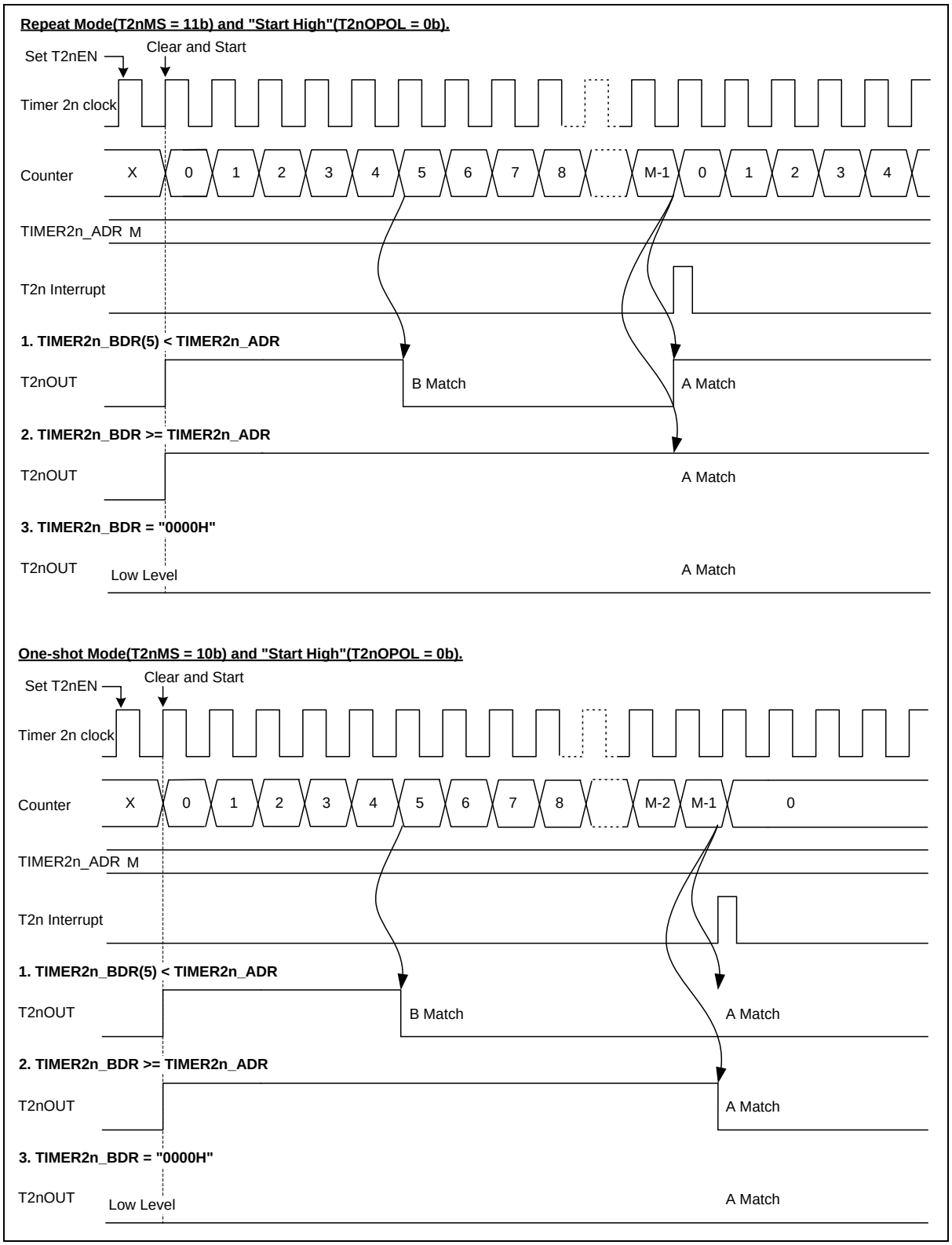
Figure 51. 32-Bit PPG Repeat and One-Shot Mode for Timer 2n (n = 0, 1)



NOTE:

1. The T2nEN is automatically cleared to logic '0' after one pulse is generated in PPG one-shot mode (n = 0, 1).

Figure 52. 32-Bit PPG Mode Timing Chart for Timer 2n (N = 0, 1)



11.6 TIMER 20/21 Registers

The base address of Timer 2n is described in the following:

Table 57. Base Address of Timer 20/21

Name	Base Address
TIMER20	0x4000_2500
TIMER21	0x4000_2600

Table 58. TIMER Register Map

Name	Offset	Type	Description	Reset Value	Reference
TIMER2n_CR	0x00	RW	Timer/Counter 2n Control Register	0x0000_0000	11.6.1
TIMER2n_ADR	0x04	RW	Timer/Counter 2n A Data Register	0xFFFF_FFFF	11.6.2
TIMER2n_BDR	0x08	RW	Timer/Counter 2n B Data Register	0xFFFF_FFFF	11.6.3
TIMER2n_CAPDR	0x0C	RO	Timer/Counter 2n Capture Data Register	0x0000_0000	11.6.4
TIMER2n_PREDR	0x10	RW	Timer/Counter 2n Prescaler Data Register	0x0000_0FFF	11.6.5
TIMER2n_CNT	0x14	RO	Timer/Counter 2n Counter Register	0x0000_0000	11.6.6

11.6.1 TIMER2n_CR: Timer/Counter 2n Control Register

The timer module should be configured properly before running. Once target purpose is defined, the timer can be configured in the TIMER2n_CR register. After configuring this register, a user can start or stop the timer function by using this register.

TIMER2n_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_CR=0x4000_2500, TIMER21_CR=0x4000_2600

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12	11	10 9	8	7 6	5	4	3	2	1	0
Reserved		T2nEN	T2nCLK	T2nMS[1:0]	T2nECE	CAPSEL[1:0]	T2nOPOL	T2nCPOL[1:0]	T2nMIEN	T2nCIEN	T2nMIFLAG	T2nCIFLAG	T2nPAU	T2nCLR
-		0	0	00	0	00	0	00	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	T2nEN	Timer 2n Operation Enable.
		0 Disable timer 2n operation.
		1 Enable timer 2n operation. (Counter clear and start)
14	T2nCLK	Timer 2n Clock Selection.
		0 Select an internal prescaler clock.
		1 Select an external clock.
		NOTE: 1. This bit should be changed while T2nEN bit is '0'.
13 12	T2nMS[1:0]	Timer 2n Operation Mode Selection.
		00 Timer/Counter mode. (T2nOUT: toggle at A-match)
		01 Capture mode. (The A-match interrupt can occur)
		10 PPG one-shot mode. (T2nOUT: Programmable pulse output)
		11 PPG repeat mode. (T2nOUT: Programmable pulse output)
		NOTE: 1. This bit should be changed while T2nEN bit is '0'.
11	T2nECE	Timer 2n External Clock Edge Selection.
		0 Select falling edge of external clock.
		1 Select rising edge of external clock.
10 9	CAPSEL[1:0]	Timer 2n Capture Signal Selection (TIMER20 only)
		00 Select an external capture signal.
		01 Select the LSE (External sub oscillator) signal.
		10 Select the LSI (Low-speed internal RC oscillator) signal.
		11 Not used
		NOTE: 1. This bit should be changed while T20EN bit is '0'.
8	T2nOPOL	T2nOUT Polarity Selection.
		0 Start high. (T2nOUT is low level at disable)
		1 Start low. (T2nOUT is high level at disable)
7 6	T2nCPOL[1:0]	Timer 2n Capture Polarity Selection.
		00 Capture on falling edge.
		01 Capture on rising edge.
		10 Capture on both falling and rising edge.

		11	Reserved.
5	T2nMIEN	Timer 2n Match Interrupt Enable.	
		0	Disable timer 2n match interrupt.
		1	Enable timer 2n match interrupt.
4	T2nCIEN	Timer 2n Capture Interrupt Enable.	
		0	Disable timer 2n capture interrupt.
		1	Enable timer 2n capture interrupt.
3	T2nMIFLAG	Timer 2n Match Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
2	T2nCIFLAG	Timer 2n Capture Interrupt Flag bit.	
		0	No request occurred.
		1	Request occurred. The bit is cleared to ‘0’ when ‘1’ is written.
1	T2nPAU	Timer 2n Counter Temporary Pause Control bit.	
		0	Continue counting.
		1	Temporary pause.
0	T2nCLR	Timer 2n Counter and Prescaler Clear bit.	
		0	No effect.
		1	Clear timer 2n counter and prescaler. (Automatically cleared to ‘0’ after operation)

11.6.2 TIMER2n_ADR: Timer/Counter 2n A Data Register

TIMER2n_ADR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_ADR=0x4000_2504, TIMER21_ADR=0x4000_2604

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADATA[31:0]																															
0xFFFFFFFF																															
RW																															

31 0	ADATA[31:0]	Timer/Counter 2n A Data. The range is 0x00000002 to 0xFFFFFFFF. (n = 0, 1)

NOTE:

1. Do not write “0x00000000” in the TIMER2n_ADR register under PPG mode.

11.6.3 TIMER2n_BDR: Timer/Counter 2n B Data Register

TIMER2n_BDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_BDR=0x4000_2508, TIMER21_BDR=0x4000_2608

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDATA[31:0]																															
0xFFFFFFFF																															
RW																															

31 0	BDATA[31:0]	Timer/Counter 2n B Data. The range is 0x00000000 to 0xFFFFFFFF.

11.6.4 TIMER2n_CAPDR: Timer/Counter 2n Capture Data Register

TIMER2n_CAPDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_CAPDR=0x4000_250C, TIMER21_CAPDR=0x4000_260C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPD[31:0]																															
0x00000000																															
RO																															

31 0	CAPD[31:0]	Timer/Counter 2n Capture Data (n = 0, 1).

11.6.5 TIMER2n_PREDR: Timer/Counter 2n Prescaler Data Register

TIMER2n_PREDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_PREDR=0x4000_2510, TIMER21_PREDR=0x4000_2610

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				PRED[11:0]											
-																				0xFFF											
-																				RW											

11 0	PRED[11:0]	Timer/Counter 2n Prescaler Data (n = 0, 1).

11.6.6 TIMER2n_CNT: Timer/Counter 2n Counter Register

TIMER2n_CNT register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TIMER20_CNT=0x4000_2514, TIMER21_CNT=0x4000_2614

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[31:0]																															
0x00000000																															
R0																															

31 0	CNT[31:0]	Timer/Counter 21 Counter (n = 0, 1).

11.6.7 TIMER20 Register Map Summary

Table 59. TIMER20 Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMER20_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T20EN	T20CLK	T20MS[1:0]		T20ECE	CAPSEL[1:0]		T20OPOL	T20CPOL[1:0]		T20MIEN	T20CIEN	T20MIFLAG	T20CIFLAG	T20PAU	T20CLR
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	TIMER20_ADR	ADATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	TIMER20_BDR	BDATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	TIMER20_CAPDR	CAPD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	TIMER20_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	TIMER20_CNT	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.8 TIMER21 Register Map Summary

Table 60. TIMER21 Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMER21_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T21EN	T21CLK	T21MS[1:0]		T21ECE	Res.	Res.	T21OPOL	T21CPOL[1:0]		T21MIEN	T21CIEN	T21MIFLAG	T21CIFLAG	T21PAU	T21CLR
	Reset value																	0	0	0	0	0			0	0	0	0	0	0	0	0	0
0x04	TIMER21_ADR	ADATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	TIMER21_BDR	BDATA[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	TIMER21_CAPDR	CAPD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	TIMER21_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	TIMER21_CNT	CNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12. USART (UART and SPI)

12.1 Introduction

USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter) is a highly flexible serial communication device. USART supports UART and SPI modes. The A31G336 has four channel USART modules.

12.2 Main Features

The USART of A31G336 features the following:

- Full duplex operation. (independent serial receive and transmit registers)
- Asynchronous or synchronous operation
- Baud rate generator
- Supports serial frames with 5,6,7,8, or 9 Data bits and 1 or 2 Stop bits
- Odd or even parity generation, and parity check supported by hardware.
- Supports receive character detection and receive time out function
- Supports Local Interconnection Network (LIN)
- Data OverRun Detection
- Framing Error Detection
- Three separate interrupts on TX completion, TX data register empty and RX completion
- Double Speed Asynchronous communication mode
- Up to 8 MHz data transfer for SPI

12.3 USART 10/11/12/13 Block Diagram

Figure 53 shows a block diagram of the UART and LIN block.

Figure 53. UART and LIN Block Diagram of USART (n = 10, 11, 12, and 13)

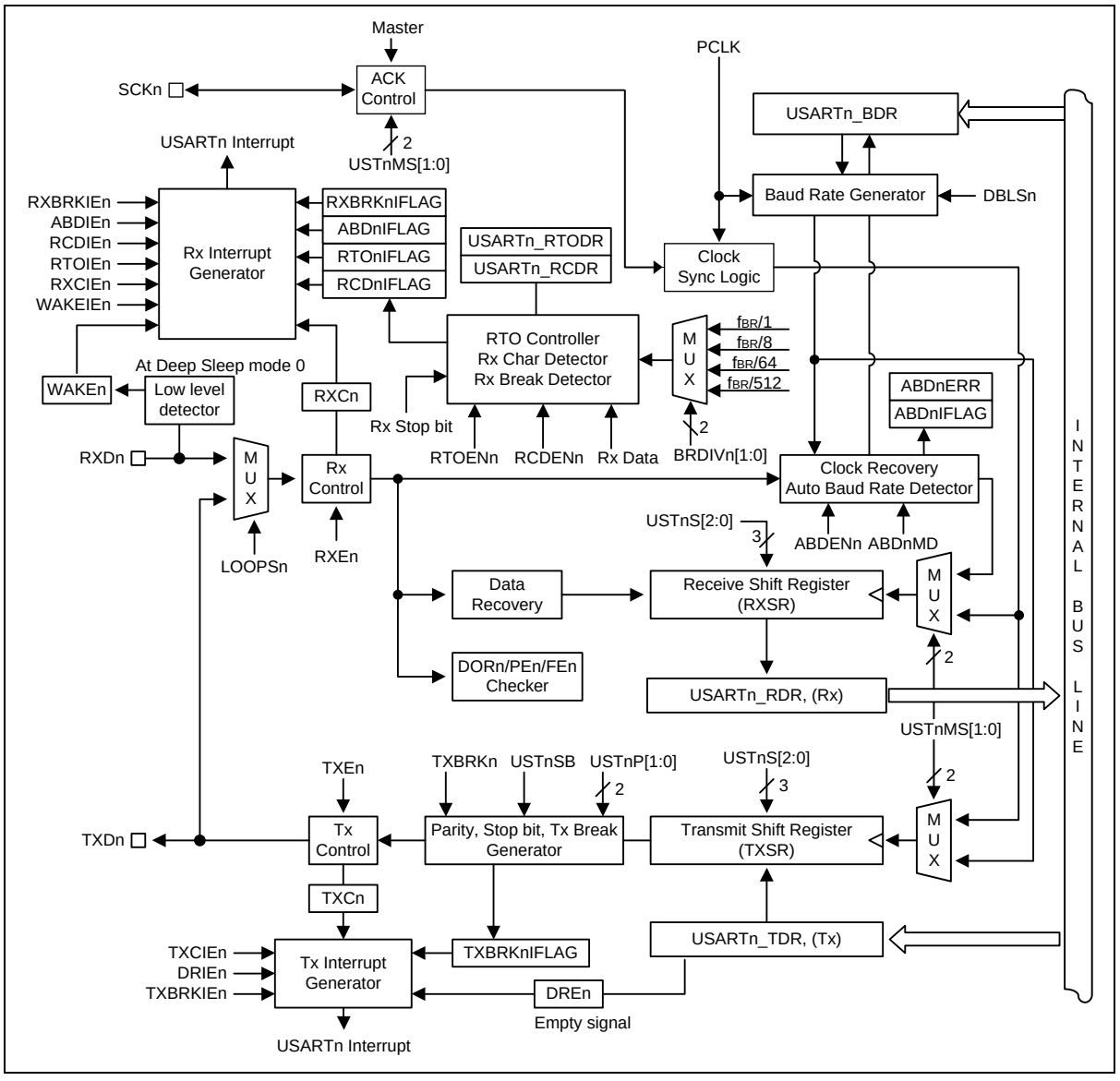
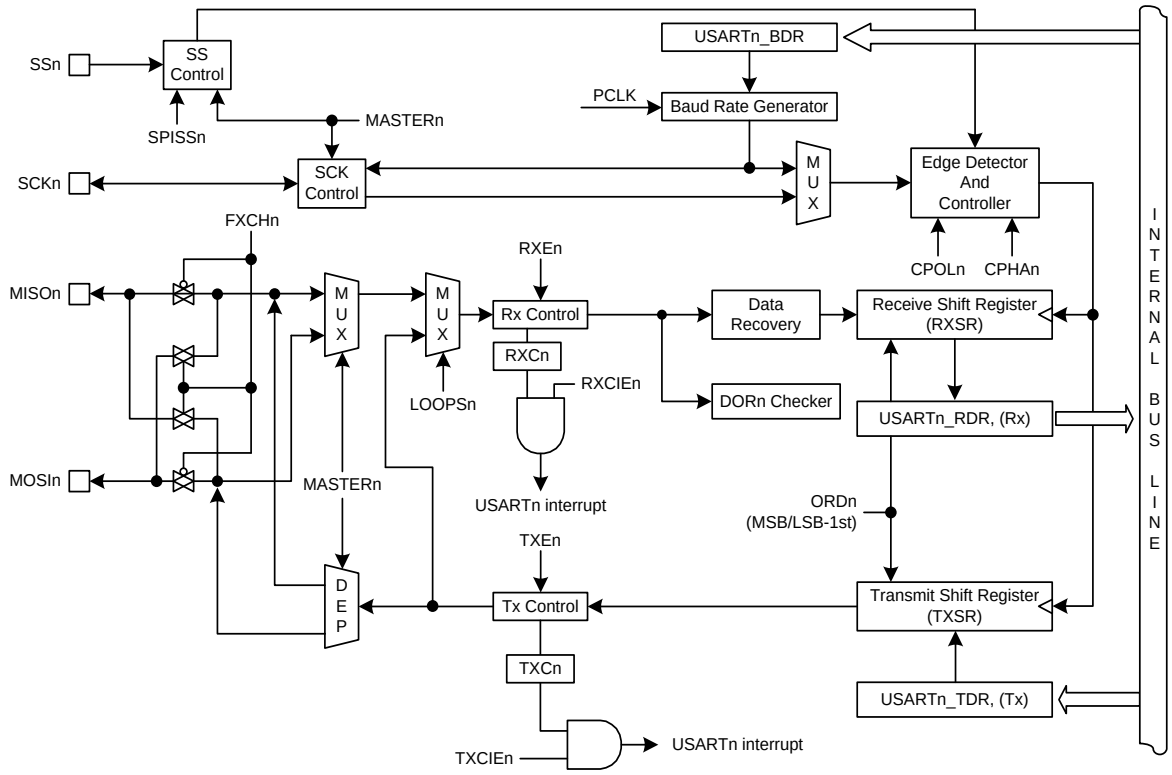


Figure 54 shows a block diagram of the SPI block.

Figure 54. SPIn Block Diagram of USART (n = 10, 11, 12, and 13)



NOTES:

1. The clock frequency of SCKn should be less than or equal to PCLK/2.
2. When DMA is used, the interval of data to be received requires at least 32 clocks of PCLK.

12.4 Pin Description for USART 10/11/12/13

Table 61. Pins and External Signals for USART 10/11/12/13

Pin Name	Type	Description
TXDn	O	UART Channel n transmit output
RXDn	I	UART Channel n receive input
SSn	I/O	SPIn Slave select input /output
SCKn	I/O	SPIn Serial clock input/output
MOSIn	I/O	SPIn Serial data (Master output, Slave input)
MISOn	I/O	SPIn Serial data (Master input, Slave output)

12.5 UART Functional Description

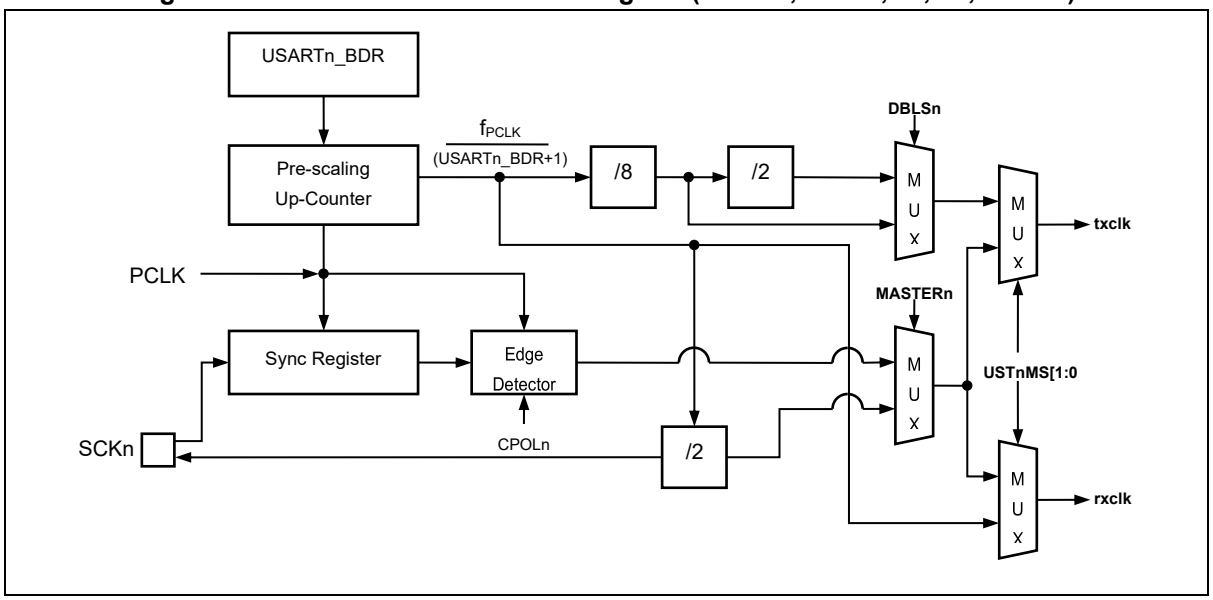
The USART comprises a clock generator, a transmitter, and a receiver. The clock generation logic includes synchronization logic for external clock input, which is used for synchronizing or SPI slave operation. Baud rate generator in the clock generation logic is for asynchronous or master (synchronous or SPI) operation.

The Transmitter consists of a write buffer, a serial shift register, a parity generator, and a control logic. Using DMA allows continuous transfer of data without any software involvement between frames.

The receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition, the receiver has a parity checker, a shift register, and control logic. The receiver supports the same frame formats as the transmitter and can detect frame errors, data overrun and parity errors. (n = 10, 11, 12, and 13)

12.5.1 USART Clock Generation

Figure 55. Clock Generation Block Diagram (USART, n = 10, 11, 12, and 13)



The clock generation logic generates the base clock for the transmitter and receiver. The USART supports four modes of clock operation, which are Normal asynchronous mode, Double speed asynchronous mode, Master synchronous mode and Slave synchronous mode.

The clock generation scheme for master SPI and slave SPI mode is the same as master synchronous and slave synchronous operation mode. The USTnMS[1:0] bits in USARTn_CR1 register selects asynchronous or synchronous operation. Asynchronous double speed mode is controlled by the DBLS bit in the USARTn_CR2 register.

The MASTER bit in USARTn_CR2 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCKn pin is active only when the USART operates in synchronous or SPI mode.

Table 62 shows the equations for calculating the baud rate (in bps).

Table 62. Equations for Calculating USART Baud Rate Register Settings (n = 10, 11, 12 and 13)

Operating Mode	Equation For Calculating Baud Rate
Asynchronous Normal Mode (DBLSn=0)	Baud Rate = PCLK/(16(USARTn_BDR+1))
Asynchronous Double Speed Mode (DBLSn=1)	Baud Rate = PCLK/(8(USARTn_BDR+1))
Synchronous or SPI Master Mode	Baud Rate = PCLK/(2(USARTn_BDR+1))

12.5.2 External Clock (SCKn)

External clock is used in the Synchronous mode or in the SPI slave mode. External clock input from the SCKn pin is sampled by the synchronization logic to remove meta-stability. The output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and the receiver.

This process describes two CPU clock period delay. The maximum frequency of the external SCKn pin is limited to 8 MHz.

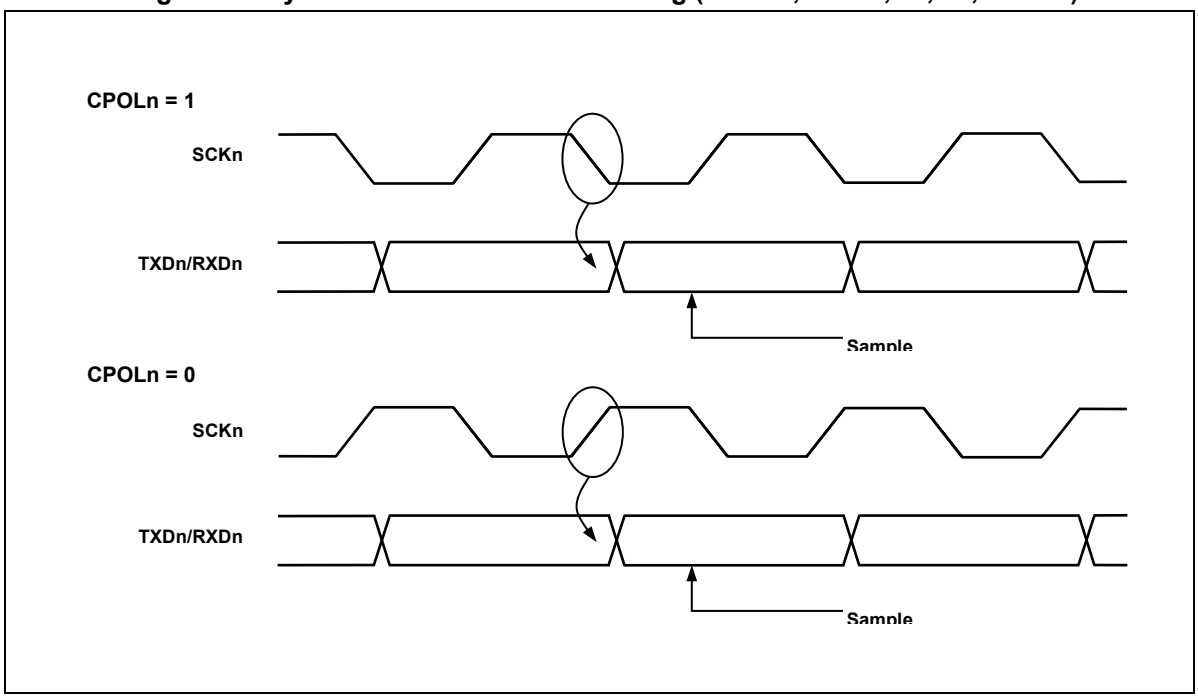
12.5.3 Synchronous Mode Operation

External clock is used in the Synchronous mode or in the SPI slave mode. When the Synchronous or the SPI mode is used, the SCKn pin will be used as either clock input (slave) or clock output (master).

Data sampling and transmission are issued on different edges of SCKn clock respectively. For example, if data input on RXDn (MISOn in SPI mode) pin is sampled on the rising edge of SCKn clock, data output on TXDn (MOSIn in SPI mode) pin is altered on the falling edge.

The CPOLn bit in USARTn_CR1 register selects which SCKn clock edge is used for data sampling and which is used for data change. As shown in Figure 56 below, when CPOLn is zero, the data will be changed at rising SCKn edge and sampled at falling SCKn edge.

Figure 56. Synchronous Mode SCKn Timing (USART, n = 10, 11, 12, and 13)



12.5.4 UART Data Format

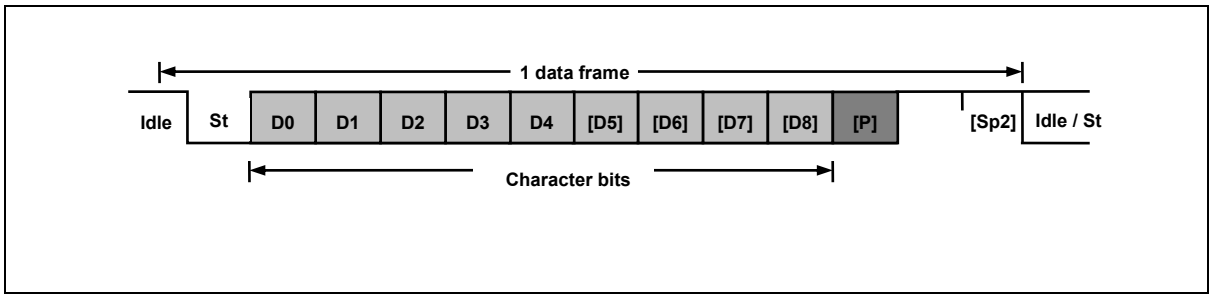
A serial frame is defined as one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection. The USART supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- No, even or odd parity bit.
- 1 or 2 stop bits.

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to nine, follow, ending with the most significant bit (MSB). If a parity function is enabled, the parity bit is inserted between the last data bit and the stop bit. A high-to-low transition on data pin is considered as a start bit.

When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle state means high state of data pin. Figure 57 shows a possible combination of the frame formats. Bits inside brackets are optional.

Figure 57. Frame Format (UART)



1 data frame consists of the following bits

- Idle: No communication on the TXDn/RXDn line
- St: Start bit (low)
- Dm: Data bits (0 to 8)
- P: Parity bit (even parity, odd parity, no parity)
- Sp: Stop bit (1 bit or 2 bits)

The frame format used by the UART is set by USTnS[2:0], USTnP[1:0] bits in the USARTn_CR1 register and USTnSB bit in the USARTn_CR2 register. The transmitter and the receiver use the same values. (n = 10, 11, 12, and 13)

12.5.5 UART Parity Bit

The parity bit is calculated by doing an exclusive-OR of all data bits. If odd parity is used, the result of the exclusive-OR is inverted. The parity bit is located between the MSB and the first stop bit of a serial frame.

- Peven = Dm-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 0
- Podd = Dm-1 ^ … ^ D3 ^ D2 ^ D1 ^ D0 ^ 1
- Peven: Parity bit using even parity
- Podd: Parity bit using odd parity
- Dm: Data bit n of the character

12.5.6 UART Transmitter

The UART transmitter is enabled by setting TXEn bit in the USARTn_CR1 register. When the Transmitter is enabled, the TXDn pin should be set to TXDn function for the serial output pin in UART mode by the GPIO registers. Baud-rate, operation mode and frame format must be set up before starting any transmission. In Synchronous operation mode, the SCKn pin is used for transmission clock, so it should be selected to do SCKn function by the GPIO registers. (n = 10, 11, 12, and 13)

12.5.6.1 UART Sending TX Data

A data transmission is initiated by loading data to the transmit data register (USARTn_TDR). The data to be written in transmit data register is moved to the shift register when the shift register is ready to send a new frame.

The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded to the new data, it will transfer one complete frame according to the settings of the control registers.

If 9-bit characters are used in the Asynchronous or the Synchronous operation mode, the 9th bit must be written to TDATA[8] bit in the USARTn_TDR register. (n = 10, 11, 12, and 13)

12.5.6.2 UART Transmitter Flag and Interrupt

The UART transmitter has two flags that indicate its state. One is USART data register empty flag (DREn) and the other is transmit completion flag (TXCn). Both flags can be used as interrupt sources.

DREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register.

When the data register empty interrupt enable (DRIEn) bit in USARTnCR1 register is set and the global interrupt is enabled, USARTn_ST status register empty interrupt is generated while DREn flag is set.

Transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out. The TXCn flag can be cleared by writing ‘1’ to TXCn bit in the USARTn_ST register.

When transmit complete interrupt enable (TXCIEn) bit in the USARTn_CR1 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXCn flag is set. (n = 10, 11, 12 and 13)

12.5.6.3 UART Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USTnP1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent. (n = 10, 11, 12, and 13)

12.5.6.4 UART Disabling Transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until the current transmission is completed. When the transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (GPIO). (n = 10, 11, 12, and 13)

12.5.7 UART Receiver

The UART receiver is enabled by setting RXEn bit in the USARTn_CR1 register. When the receiver is enabled, the RXDn pin should be set to RXDn function for the serial input pin in the UART mode by the GPIO registers. Baud-rate, operation mode and frame format must be set before serial reception. In Synchronous or SPI operation mode, the SCKn pin is used as a transfer clock input, so it should be selected to do SCKn function by the GPIO registers. (n = 10, 11, 12, and 13)

12.5.7.1 UART Receiving RX Data

When the UART is in Synchronous mode or in Asynchronous mode, the receiver starts data reception if it detects a valid start bit (LOW) on RXDn pin. Each bit after the start bit is sampled at predefined baud-rate (asynchronous) or at sampling edge of SCKn (synchronous) and shifted into the receive shift register until the first stop bit of a frame is received.

Even if there is a second stop bit in the frame, the second stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is presented in the receiver shift register and contents of the shift register are to be moved into the receive data register (USARTn_RDR). (n = 10, 11, 12, and 13)

12.5.7.2 UART Receiver Flag and Interrupt

The UART receiver has a flag that indicates the receiver's state. The receive complete (RXCn) flag indicates whether there are unread data in the receive buffer. This flag is set when there is unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When the receive complete interrupt enable (RXCIEn) bit in the USARTn_CR1 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXCn flag is set.

The UART receiver has three error flags, which are frame error (FEn), data overrun (DORn) and parity error (PEn). These error flags can be read from the USARTn_ST register.

The frame error (FEn) flag indicates the state of the first stop bit. The FEn flag is '0' when the stop bit was correctly detected as '1', while the FEn flag is '1' when the stop bit was incorrect, i.e. detected as '0'. This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DORn) flag indicates data loss due to a full receive buffer condition. DORn occurs when the receive buffer is full, and another new data is presented in the receive shift register to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To avoid data loss or to clear this flag, receive buffer must be read.

The parity error (PEn) flag indicates that the frame in the receive buffer had a parity error during reception. If parity check function is not enabled (USTnP1=0), the PEn bit is always read as '0'. (n = 10, 11, 12, and 13)

12.5.7.3 UART Parity Checker

If parity bit is enabled (USTnP1=1), the parity checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame. (n = 10, 11, 12, and 13)

12.5.7.4 UART Disabling Receiver

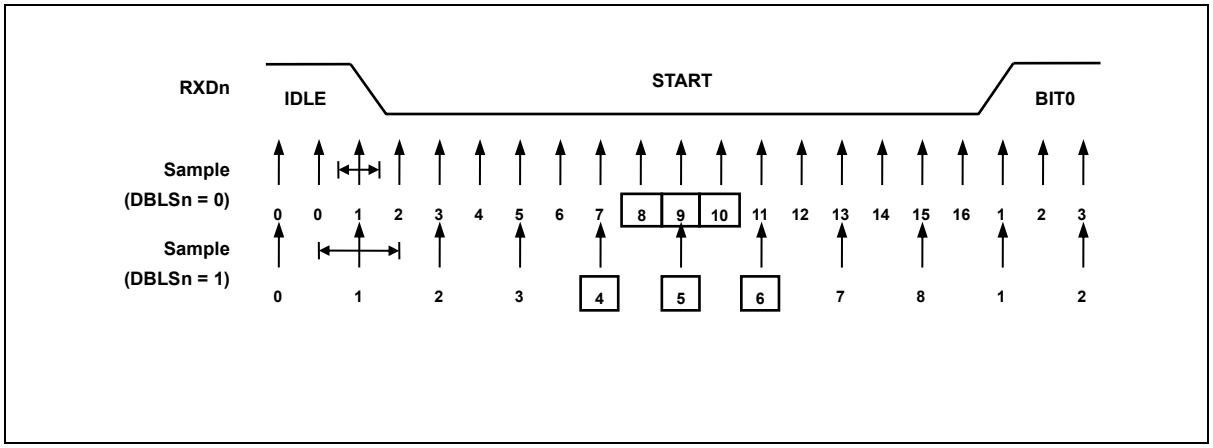
Unlike the transmitter, the receiver becomes inactive immediately after it is disabled by clearing RXEn bit. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (GPIO). (n = 10, 11, 12, and 13)

12.5.7.5 Asynchronous Data Reception

To receive an asynchronous data frame, the USART includes a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXDn pin. The data recovery logic samples and filters the incoming bits with a low pass filter and removes the noise of RXDn pin.

Figure 58 illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud-rate in normal mode and 8 times the baud-rate for double speed mode (DBLSn=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is seen using the double speed mode. (n = 10, 11, 12, and 13)

Figure 58. Asynchronous Start Bit Sampling (n = 10, 11, 12, and 13)



When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, which is the start bit condition. After detecting the high-to-low transition on RXDn line, the clock recovery logic uses samples 8, 9 and 10 for normal mode to detect whether valid start bit is received.

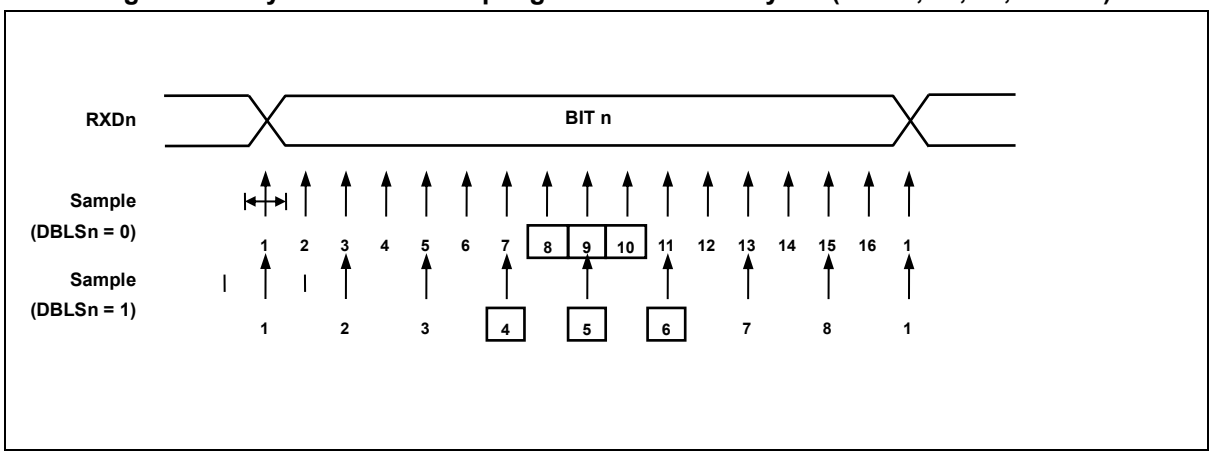
If more than 2 samples have logical low level, it is considered that a valid start bit is detected, and the internally generated clock is synchronized to the incoming data frame. Then the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost the same as clock recovery process.

The data recovery logic samples each incoming bit 16 times for normal mode and 8 times for double speed mode, and uses sample 8, 9 and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered as a logic ‘0’ and if more than 2 samples have high levels, the received bit is considered as a logic ‘1’.

The data recovery process is then repeated until a complete frame is received, including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the receiver is in idle state and waits to find the start bit. (n = 10, 11, 12, and 13)

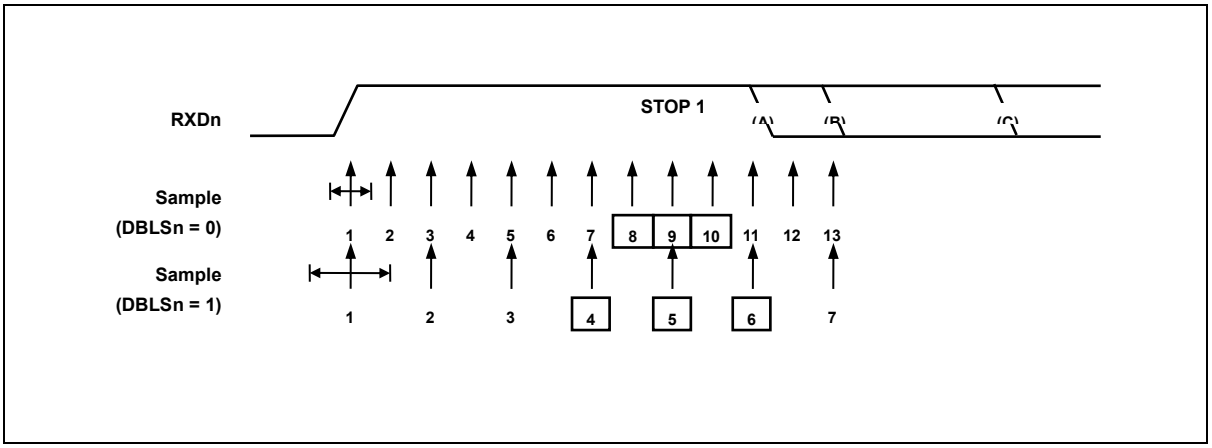
Figure 59. Asynchronous Sampling of Data and Parity Bit (n = 10, 11, 12, and 13)



The process for detecting stop bit is the same as clock and data recovery process. That is, if two or more samples of three center values have high level, correct stop bit is detected, or else a frame error (FEn) flag is set.

After deciding whether the first stop bit is valid or not, the receiver goes to idle state and monitors the RXDn line to check whether a valid high to low transition is detected (start bit detection). (n = 10, 11, 12, and 13)

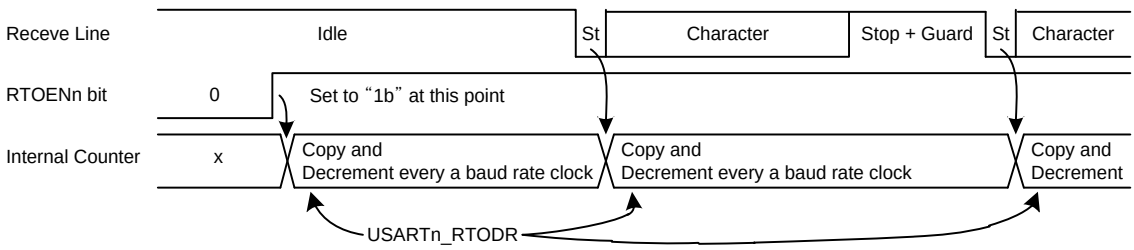
Figure 60. Stop Bit Sampling and Next Start Bit Sampling (n = 10, 11, 12, and 13)



12.5.7.6 Receive Timeout Function

The receive timeout function is used for checking the frame finish. This function is to count time with baud rate unit between the last start bit and a new start bit, and between setting the RTOENn bit of USARTn_CR3 register and a new start bit. The USARTn_RTODR register should have duration time value before using the receive time out function. (n = 10, 11, 12, and 13)

Figure 61. Receive Timeout Function (n = 10, 11, 12, and 13)



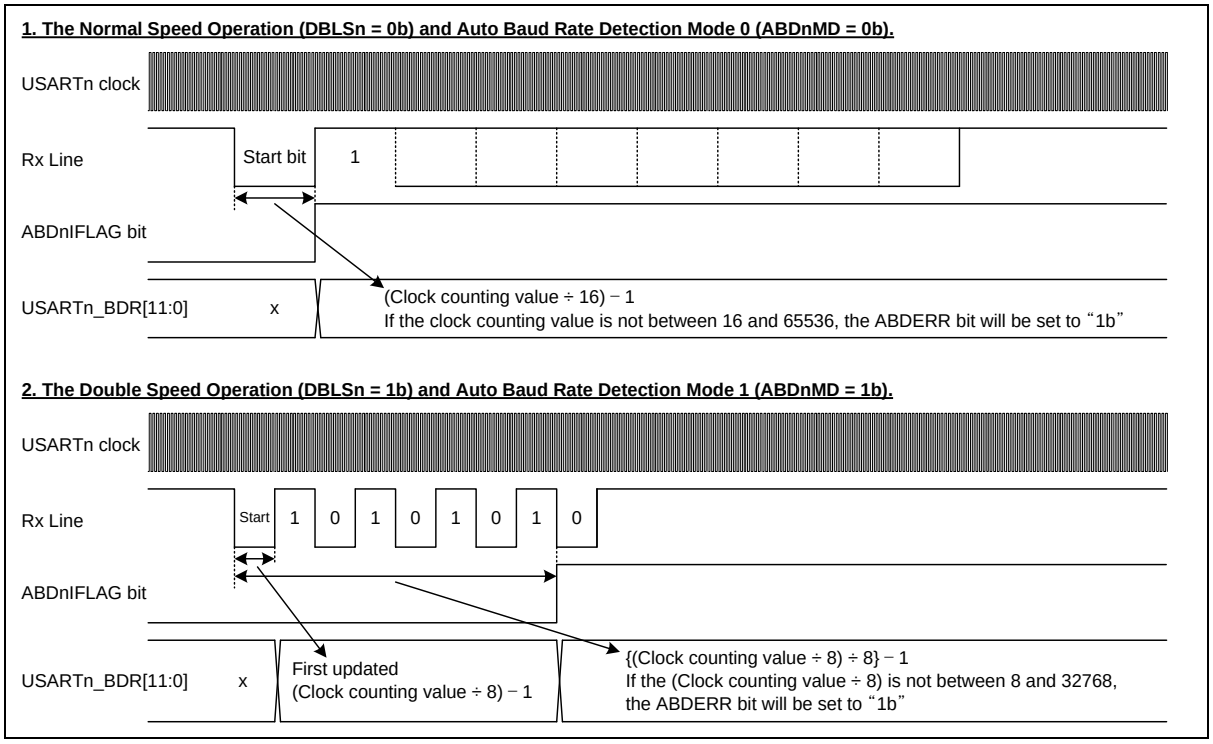
NOTES:

1. The value of USARTn_RTODR register is transferred to an internal counter when the RTOENn bit is set to '1' or when a start bit is received to.
2. If the RTOENn bit is set to '1', the internal counter counts down until underflow.
3. The RTOnIFLAG bit of USARTn_CR3 register is set to '1' on the internal counter underflow and the receive time out function is finished. So, to re-use the receive time out function, the RTOENn bit must be cleared to '0' and then set to '1' again.

12.5.7.7 UART Auto Baud Rate Detection

The auto baud rate detection is enabled by setting '1' to the ABDENn bit of the USARTn_CR3 register. The function is useful when using clock source with relatively low accuracy. There are two auto baud rate detection modes, "Start bit to measure" and "0x55 character to measure". (n = 10, 11, 12, and 13)

Figure 62. Auto Baud Rate Detection Timing Diagram (n = 10, 11, 12, and 13)



12.5.8 SPI Mode

The USART can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full Duplex, Three-wire synchronous data transfer.
- Mater and Slave Operation.
- Supports all four SPI modes of operation (mode 0, 1, 2, and 3).
- Selectable LSB first or MSB first data transfer.
- Double buffered transmit and receive.
- Programmable transmit bit rate.
- Up to 8 MHz data transfer for SPI

When the SPI mode is enabled by configuring USTnMS[1:0] as ”11”, the slave select (SSn) pin becomes active LOW input in Slave mode operation if USTnSSEN bit is set to ‘1’. The SSn function is not automatically controlled in master mode operation even if USTnSSEN bit is set to ‘1’.

Note that during SPI mode of operation, the pin RXDn is renamed as MISOn and TXDn is renamed as MOSIn for compatibility to other SPI devices. (n = 10, 11, 12, and 13)

12.5.9 SPI Clock Formats and Timing

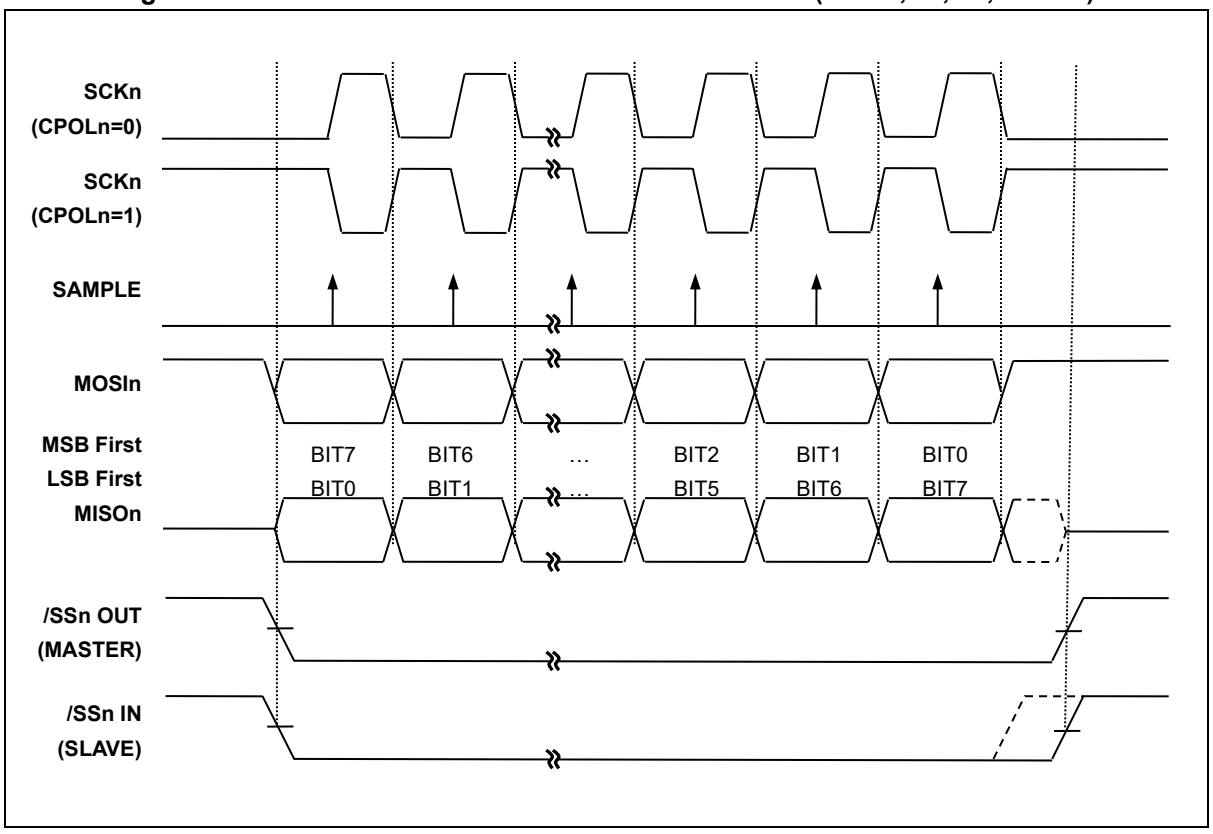
To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USART has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. CPOLn selectively inserts an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data. Note that CPHAn and CPOLn bits in USTnCR0 register have different meanings according to the USTnMS[1:0] bits, which decide the operating mode of USART.

Table 63 shows four combinations of CPOLn and CPHAn for SPI mode 0, 1, 2, and 3. (n = 10, 11, 12, and 13)

Table 63. CPOL Functionality (n = 10, 11, 12, and 13)

SPIn Mode	CPOLn	CPHAn	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

Figure 63. USART SPIn Clock Formats when CPHAn=0 (n = 10, 11, 12, and 13)

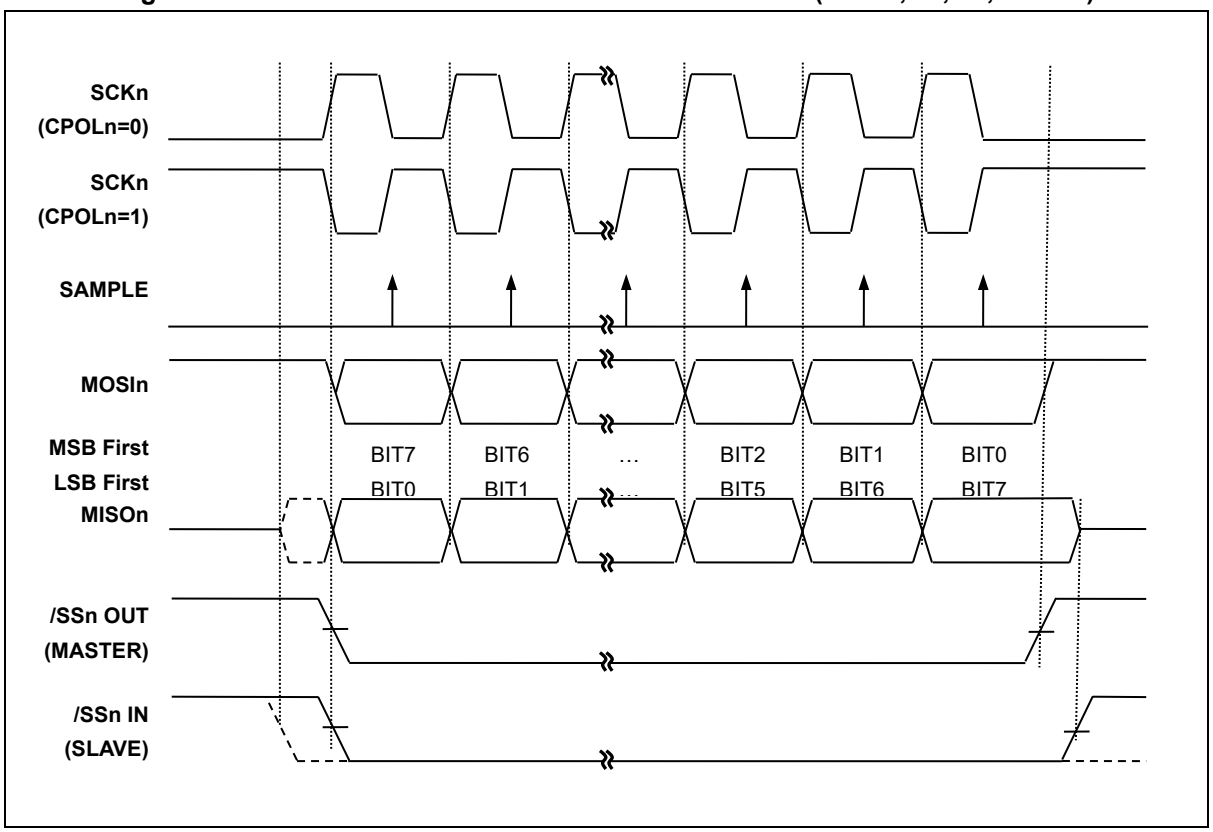


When CPHAn=0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low.

The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively.

At the second SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively. (n = 10, 11, 12, and 13)

Figure 64. USART SPIn Clock Formats when CPHAn=1 (n = 10, 11, 12, and 13)



When CPHAn=1, the slave begins to drive its MISOn output when SSn goes active low, but the data is not defined until the first SCKn edge.

The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave.

The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively.

At the third SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively.

Because the SPIn logic reuses USART resources, SPIn mode of operation is similar to that of synchronous or asynchronous operation.

A SPIn transfer is initiated by checking for the USART Data Register Empty flag (DREn=1) and then writing a byte of data to the USARTn_TDR Register. In master mode of operation, even when transmission is not enabled (TXEn=0), writing data to the USARTn_TDR register is necessary because the clock SCKn is generated from the transmitter block.

12.5.10 Local Interconnection Network (LIN) Mode

The LIN mode is selected by writing “10” to the USTnMS[1:0] bits of the USARTn_CR1 register. The LIN transmission is fixed as start bit, 8-bits data length, 1 stop bit, and no parity. So, it should be set as follows.

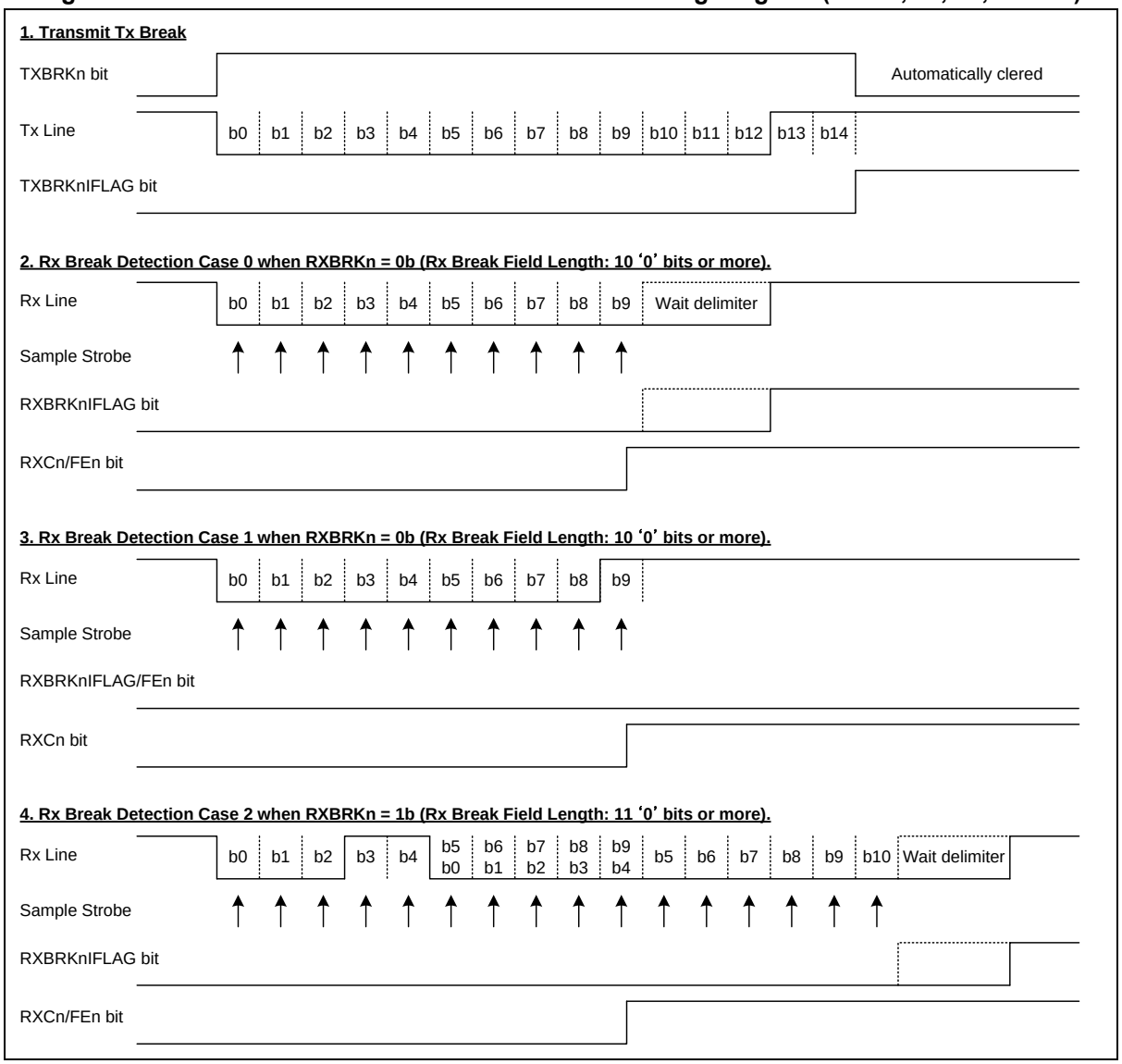
- USTnP[1:0] bits of USARTn_CR1: cleared to “00” for no parity.
- USTnS[2:0] bits of USARTn_CR1: set to “011” for 8-bit data length.
- USTnSB and LOOPSn bits USARTn_CR2: cleared to ‘0’ for 1 stop bit and normal operation.

While LIN mode is enabled, the break field detection circuit is activated, and it is independent from the UART receiver. A break field can be detected whenever it occurs during idle state or during a frame.

When the auto baud rate detection is enabled by setting ‘1’ to the ABDENn bit of USARTn_CR3 register, the Rx break field may not be detected. So, the auto baud rate detection function should be enabled after the Rx break field detection if needed.

Figure 65 shows the timing diagram for LIN break field.

Figure 65. LIN Break Field Detection and Transmit Timing Diagram (n = 10, 11, 12, and 13)



12.6 USART Registers

The base address and register map of the USART 10/11/12/13 are described in the following tables:

Table 64. Base Address of USART 10/11/12/13

Name	Base Address
USART10	0x4000_3800
USART11	0x4000_3900
USART12	0x4000_3A00
USART13	0x4000_3B00

Table 65. USARTn Register Map

Name	Offset	Type	Description	Reset Value	Reference
USARTn_CR1	0x0000	RW	USARTn control register 1	0x0000_0000	12.6.1
USARTn_CR2	0x0004	RW	USARTn control register 2	0x0000_0000	12.6.2
USARTn_CR3	0x0008	RW	USARTn control register 3	0x0000_0000	12.6.3
USARTn_ST	0x000C	RW	USARTn status register	0x0000_0080	12.6.4
USARTn_BDR	0x0010	RW	USARTn baud rate generation register	0x0000_0FFF	12.6.5
USARTn_RDR	0x0014	RO	USARTn receive data register	0x0000_0000	12.6.6
USARTn_TDR	0x0018	RW	USARTn transmit data register	0x0000_0000	12.6.7
USARTn_RTODR	0x001C	RW	USARTn receive time out data register	0x0000_00FF	12.6.8
USARTn_RCDR	0x0020	RW	USARTn receive character detection data register	0x0000_0000	12.6.9

NOTE:

1. n = 10, 11, 12, and 13.

12.6.1 USARTn_CR1: USARTn Control Register 1

The USART module should be configured properly before running.

The USARTn_CR1 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_CR1=0x4000_3800, USART11_CR1=0x4000_3900
USART12_CR1=0x4000_3A00, USART13_CR1=0x4000_3B00

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14	13 12	11 10 9	8	7	6	5	4	3	2	1	0
Reserved		USTnMS[1:0]	USTnP[1:0]	USTnS[2:0]	ORDn	CPOLn	CPHAn	DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn
-		00	00	000	0	0	0	0	0	0	0	0	0
-		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15 14	USTnMS[1:0]	USARTn Operation Mode Selection.	
		00	Asynchronous Mode. (UART)
		01	Synchronous Mode.
		10	Local Interconnection Network (LIN)
		11	SPI mode

NOTE:

1. The LIN transmission is fixed as “Start, D0, D1, D2, D3, D4, D5, D6, D7, Stop1”. So, the USTnP[1:0], LOOPSn, and USTnSB bits must be cleared to ‘0’ and the USTnS[2:0] bits should be set to “011”.

13 12	USTnP[1:0]	Selects Parity Generation and Check method. (only UART mode)	
		00	No parity.
		01	Reserved.
		10	Even parity.
		11	Odd parity.
11 9	USTnS[2:0]	Selects the length of data bit in a frame at Asynchronous or Synchronous mode.	
		000	5-bit
		001	6-bit
		010	7-bit
		011	8-bit
		111	9-bit
		Others	Reserved.
8	ORDn	Selects the first data bit to be transmitted. (only SPI mode)	
		0	LSB-first.
		1	MSB-first.
7	CPOLn	Selects the clock polarity of SCK in synchronous or SPI mode.	
		0	SCK to 0 when idle.
		1	SCK to 1 when idle.
6	CPHAn	CPOLn and this bit determine if data are sampled on the leading or the trailing edge of SCK. (only SPI mode)	
		CPOLn	CPHAn	Leading edge	Trailing edge
		0	0	Sample (Rising)	Setup (Falling)
		0	1	Setup (Rising)	Sample (Falling)
		1	0	Sample (Falling)	Setup (Rising)
		1	1	Setup (Falling)	Sample (Rising)
5	DRIEn	Transmit Data Register Empty Interrupt Enable.	

		0	Disable transmit data empty interrupt.
		1	Enable transmit data empty interrupt.
4	TXCIEn	Transmit Complete Interrupt Enable.	
		0	Disable transmit complete interrupt.
		1	Enable transmit complete interrupt.
3	RXCIEn	Receive Complete Interrupt Enable.	
		0	Disable receive complete interrupt.
		1	Enable receive complete interrupt.
2	WAKEIEn	Asynchronous Wake-up Interrupt Enable in DEEP SLEEP Mode. When the device is in DEEP SLEEP mode, if RXDn goes to low level, an interrupt can be requested to wake-up system (only UART and LIN mode). This bit should be cleared to '0' to receive Rx data.	
		0	Disable asynchronous wake-up interrupt.
		1	Enable asynchronous wake-up interrupt. (Only used for wake-up)
1	TXEn	Enables the Transmitter unit.	
		0	Transmitter is disabled.
		1	Transmitter is enabled.
0	RXEn	Enables the Receiver unit.	
		0	Receiver is disabled.
		1	Receiver is enabled.

NOTE:

1. The CPOLn and CPHAn bits should be changed while TXEn and RXEn bits are '0'.

12.6.2 USARTn_CR2: USARTn Control Register 2

The USART module should be configured properly before running.

The USARTn_CR2 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_CR2=0x4000_3804, USART11_CR2=0x4000_3904
USART12_CR2=0x4000_3A04, USART13_CR2=0x4000_3B04

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10	9	8	7	6	5	4	3	2	1 0
Reserved			USTnEN	DBLSn	MASTERn	LOOPSn	DISSCKn	USTnSSEN	FXCHn	USTnSB	Reserved
-			0	0	0	0	0	0	0	0	-
-			RW	RW	RW	RW	RW	RW	RW	RW	-

9	USTnEN	Enable USARTn block. This bit can be cleared to ‘0’ while the corresponding TXEn and RXEn bits are all ‘0b’. **NOTE**: This bit should be set to ‘1’ after setting the related registers.
	0	Disable USARTn block.
	1	Enable USARTn block.
8	DBLSn	Selects receiver sampling rate. (only asynchronous and LIN mode)
	0	Normal asynchronous operation.
	1	Double speed asynchronous operation.
7	MASTERn	Selects master or slave in SPIn or Synchronous mode and controls the direction of SCKn pin.
	0	Slave operation. (External clock for SCKn)
	1	Master operation. (Internal clock for SCKn)
6	LOOPSn	1. 1-wire Half-Duplex Communication on Asynchronous Mode.
	0	Normal operation.
	1	1-wire half-duplex communication (The TXD and RXD lines are internally connected, the RXD pin is not used, and the TXD pin is always an input when no transmitted. So, the TXD pin must be configured to open-drain with an external pull-up resistor)
	2. Loop Back for Test on SPI and Synchronous Mode	
	0	Normal operation.
	1	Loop back (The “MOSI and MISO”/”TXD and RXD” lines are internally connected and the receive input is not used).
5	DISSCKn	In synchronous mode operation, selects the waveform of SCKn output.
	0	SCKn continues to output the clock while USARTn is enabled in synchronous master mode.
	1	SCKn is active while any frame is transferring.
4	USTnSSEN	This bit controls the SSn pin operation. (only SPI mode)
	0	Disable.
	1	Enable. (The SS pin should be configured as an alternate function)
3	FXCHn	SPIn port function exchange control. (only SPI mode)
	0	No effect.
	1	Exchange MOSIn and MISOn function.
2	USTnSB	Selects the length of stop bit in Asynchronous or Synchronous mode.
	0	1-Stop bit.
	1	2-Stop bit.

12.6.3 USARTn_CR3: USARTn Control Register 3

The USART module should be configured properly before running. This register is used only for UART and LIN mode. The USARTn_CR3 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 10, 11, 12, and 13)

USART10_CR3=0x4000_3808, USART11_CR3=0x4000_3908
USART12_CR3=0x4000_3A08, USART13_CR3=0x4000_3B08

31 30 29 28 27 26 25 24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0
Reserved	RXBRKn	TXBRKn	RXBRKIEn	TXBRKIEn	RXBRKnIFLAG	TXBRKnIFLAG	Reserved	ABDENn	ABDnMD	ABDIEn	ABDnIFLAG	ABDnERR	Reserved	RCDENn	RTOENn	Reserved	RCDIEn	RTOIEn	Reserved	RCDnIFLAG	RTOnIFLAG	Reserved	BRDIVn[1:0]
-	0	0	0	0	0	0	-	0	0	0	0	0	-	0	0	-	0	0	-	0	0	-	00
-	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW	RW	-	RW	RW	-	RW	RW	-	RW

23	RXBRKn	Rx Break Field Length (Only LIN mode)
	0	10 '0' bits or more
	1	11 '0' bits or more
22	TXBRKn	Tx Break Field (Only LIN mode)
	0	No effect
	1	Transmit Tx break (13 '0' bits + 2 '1' bits, "000000000000011", automatically cleared)
21	RXBRKIEn	Rx Break Field Detection Interrupt Enable bit (Only LIN mode)
	0	Disable Rx break detection interrupt
	1	Enable Rx break detection interrupt
20	TXBRKIEn	Tx Break Field Completion Interrupt Enable bit (Only LIN mode)
	0	Disable Tx break completion interrupt
	1	Enable Tx break completion interrupt
19	RXBRKnIFLAG	Rx Break Field Detection Interrupt flag. This bit is set when LIN break is detected.
	0	No request occurred
	1	Request occurred. This bit is cleared to '0' when writing '1'.
18	TXBRKnIFLAG	Tx Break Field Completion Interrupt flag. This bit is set when Tx break field is completely transmitted.
	0	No request occurred
	1	Request occurred. This bit is cleared to '0' when writing '1'.
16	ABDENn	Auto Baud Rate Detection Enable bit.
	0	Disable auto baud rate detection
	1	Enable auto baud rate detection (This bit is automatically cleared after operation)

NOTE:

1. In the LIN mode, the Rx break field may not be detected while this bit is '1'. So, it is recommended to set this bit to '1' for auto baud rate detection after the Rx break field detection.

15	ABDnMD	Auto Baud Rate Detection Mode
	0	Mode 0, The start bit is used to measure the baud rate (The 1ST bit must be '1')
	1	Mode 1, The 0x55 character is used to measure the baud rate detection
14	ABDIEn	Auto Baud Rate Detection Interrupt Enable bit

		0	Disable auto baud rate detection interrupt
		1	Enable auto baud rate detection interrupt
13	ABDnIFLAG	Auto Baud Rate Detection Interrupt Flag. This bit is set to ‘1’ when the auto baud rate detection finishes, whether an error occurs or not.	
		0	No request occurred
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written
12	ABDnERR	Auto Baud Rate Detection Error bit. This bit is set to ‘1’ if the clock counting values are not between 16 and 65536 on the normal speed operation (DBLSn = 0) or between 8 and 32768 on the double speed operation (DBLSn = 1).	
		0	No error occurs
		1	An error occurs. This bit is cleared to ‘0’ when writing ‘1’.
		NOTE: 1. If an error occurs, the USARTn_BDR register will not be updated.	
10	RCDENn	Receive Character Detection Function Enable bit. This function is to compare the value of USARTn_RCDR register with the value just received.	
		0	Disable receive detection function.
		1	Enable receive detection function.
9	RTOENn	Receive Time Out Function Enable bit. This function is to count time with baud rate units from the leading edge of a start bit to a new start bit. The receive time out controller counts down from the value of USARTn_RTODR register every start bit and set this bit. The RTOnIFLAG bit is set to ‘1’ at the counter underflow (only asynchronous mode).	
		0	Disable receive time out function.
		1	Enable receive time out function.
7	RCDIEn	Receive Character Detection Interrupt Enable.	
		0	Disable receive character detection interrupt
		1	Enable receive character detection interrupt
6	RTOIEn	Receive Time Out Interrupt Enable.	
		0	Disable receive time out interrupt.
		1	Enable receive time out interrupt
4	RCDnIFLAG	Receive Character detection Interrupt Flag. This bit is set to ‘1’ if the value in the USARTn_RCDR register matches the value received in the non-error state of frame and parity. On match of them, the bit may be set even if data overrun occurs.	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written.
3	RTOnIFLAG	Receive Time Out Interrupt Flag	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written.
1 0	BRDIVn[1:0]	Baud Rate Clock Dividing Selection for Receive Time Out. (only asynchronous and LIN mode).	
		00	f_{BR}/1.
		01	f_{BR}/8
		10	f_{BR}/64
		11	f_{BR}/512

12.6.4 USARTn_ST: USARTn Status Register

The USARTn_ST register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_ST=0x4000_380C, USART11_ST=0x4000_390C
USART12_ST=0x4000_3A0C, USART13_ST=0x4000_3B0C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			DREn	TXCn	RXCn	WAKEn	Reserved	DORn	FEn	PEn
-			1	0	0	0	-	0	0	0
-			RO	RW	RO	RW	-	RW	RW	RW

7	DREn	Transmit Data Register Empty Interrupt Flag. The flag is set to ‘1’ when the data in the USARTn_TDR register has been transferred to the transmit shift register. This bit is cleared by a write to the USARTn_TDR register (only UART mode).
	0	Not transferred to the transmit shift register.
	1	Transferred to the transmit shift register.
6	TXCn	Transmit Complete Interrupt Flag. This flag is set to ‘1’ when the data in the transmit shift register has been shifted out and when the DREn = 1.
	0	No request occurred.
	1	The data in the transmit shift register are shifted out completely. This bit is cleared to ‘0’ when writing ‘1’.
5	RXCn	Receive Data Register Not Empty Interrupt Flag. This bit is set to ‘1’ when the data in the receive shift register has been transferred to the USARTn_RDR register. The bit is cleared by a read to the USARTn_RDR register.
	0	No request occurred.
	1	There is data in the receive data register. This bit is cleared to ‘0’ when writing ‘1’.
4	WAKEn	Asynchronous Wake-up Interrupt Flag. This flag is set when the RXD pin is detected low while the CPU is in DEEP SLEEP mode (only UART mode)
	0	No request occurred.
	1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
2	DORn	Data Overrun bit. This bit is set when the receive shift register is transferred to the USARTn_RDR register while the RXCn=1. The data of the shift register are ignored. This bit must be cleared by software to receive new data (only UART mode).
	0	No Data OverRun.
	1	Data overrun detected. This bit is cleared to ‘0’ when writing ‘1’.
1	FEn	Frame Error bit. This bit is set when the received data does not have a valid stop bit. That is, the stop bit following the last data bit is detected as ‘0’. The bit will be cleared by hardware if new data are received (only UART mode).
	0	No Frame Error.
	1	Frame error detected. This bit is cleared to ‘0’ when writing ‘1’.
0	PEn	Parity Error bit. This bit is set when the received data has a parity error on parity enable. The bit will be cleared by hardware if new data are received (only UART mode).
	0	No Parity Error.
	1	Parity error detected. This bit is cleared to ‘0’ when writing ‘1’.

12.6.5 USARTn_BDR: USARTn Baud Rate Generation Register

The USARTn_BDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_BDR=0x4000_3810, USART11_BDR=0x4000_3910
USART12_BDR=0x4000_3A10, USART13_BDR=0x4000_3B10

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			BDATA[11:0]
-			0xFFF
-			RW

11 0	BDATA[11:0]	The value in this register is used to generate an internal baud rate in UART mode or to generate SCK clock in SPI mode. The range is 0x000 to 0xFFF in asynchronous UART and SPI mode but the range is 0x002 to 0xFFF in synchronous mode.

12.6.6 USARTn_RDR: USARTn Receive Data Register

The USARTn_RDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_RDR=0x4000_3814, USART11_RDR=0x4000_3914
USART12_RDR=0x4000_3A14, USART13_RDR=0x4000_3B14

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RDATA[8:0]
-			0x000
-			RO

8 0	RDATA[8:0]	Receive Data bits. A receive shift register is moved to this register after stop bit.

NOTE:

1. When asynchronous or synchronous mode, the RDATA[8] bit is the received 9th bit.

12.6.7 USARTn_TDR: USARTn Transmit Data Register

The USARTn_TDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_TDR=0x4000_3818, USART11_TDR=0x4000_3918
USART12_TDR=0x4000_3A18, USART13_TDR=0x4000_3B18

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			TDATA[8:0]
-			0x000
-			RW

8 0	TDATA[8:0]	Transmit Data bits. This register is moved to the transmit shift register after a previous character is completely shifted out. In SPI master mode, the SCK clock is generated when data are moved to the shift register. Do not write to this transmit data register while transmitting in SPI mode.

NOTES:

1. When asynchronous or synchronous mode, the TDATA[8] bit is the 9th bit to be transmitted.
2. The data to be transmitted should be written after all control registers are set.

12.6.8 USARTn_RTODR: USARTn Receive Timeout Data Register

The USARTn_RTODR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_RTODR=0x4000_381C, USART11_RTODR=0x4000_391C
USART12_RTODR=0x4000_3A1C, USART13_RTODR=0x4000_3B1C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RTOD[7:0]
-			0xFF
-			RW

7 0	RTOD[7:0]	USARTn Receive Timeout Data. Counting number: RTOD[7:0] +1

12.6.9 USARTn_RCDR: USARTn Receive Character Detection Data Register

The USARTn_RCDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 10, 11, 12, and 13)

USART10_RCDR=0x4000_3820, USART11_RCDR=0x4000_3920
USART12_RCDR=0x4000_3A20, USART13_RCDR=0x4000_3B20

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RCDD[7:0]
-			0x00
-			RW

7 0	RCDD[7:0]	USARTn Receive Character Detection Data.

12.6.10 USART Register Map Summary

Table 66. USART Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USARTn_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USTnMS[1:0]		USTnP[1:0]		USTnS[2:0]			ORDn	CPOLn	CPHAn	DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USARTn_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USTnEN	DBLSn	MASTERn	LOOPSn	DISSCKn	USTnSSEN	FXCHn	USTnSB	Res.	Res.
	Reset value																							0	0	0	0	0	0	0	0		
0x08	USARTn_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXBRKn	TXBRKn	RXBRKIEn	TXBRKIEn	RXBRKnIFLAG	TXBRKnIFLAG	Res.	ABDENn	ABDnMD	ABDIEn	ABDnIFLAG	ABDnERR	Res.	RCDENn	RTOENn	Res.	RCDIEn	RTOIEn	Res.	RCDnIFLAG	RTOnIFLAG	Res.	BRDIVn[1:0]	
	Reset value									0	0	0	0	0	0		0	0	0	0	0		0	0		0	0		0	0		0	0
0x0C	USARTn_ST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DREn	TXCn	RXCn	WAKEn	Res.	DORn	FEn	PEn
	Reset value																									1	0	0	0		0	0	0
0x10	USARTn_BDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDATA[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x14	USARTn_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDATA[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	USARTn_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDATA[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x1C	USARTn_RTODR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTOD[7:0]							
	Reset value																									1	1	1	1	1	1	1	1
0x20	USARTn_RCDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCDD[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

13. Serial Peripheral Interface (SPI)

13.1 Introduction

SPI interface enables synchronous serial data transfer between external serial devices. It allows full-duplex communication using 4-wires (MOSIn, MISOn, SCKn, SSn).

It supports master and slave modes and selects serial clock (SCKn) polarity. In addition, for the data transmission, it selects whether to transfer LSB first or MSB first.

13.2 Main Features

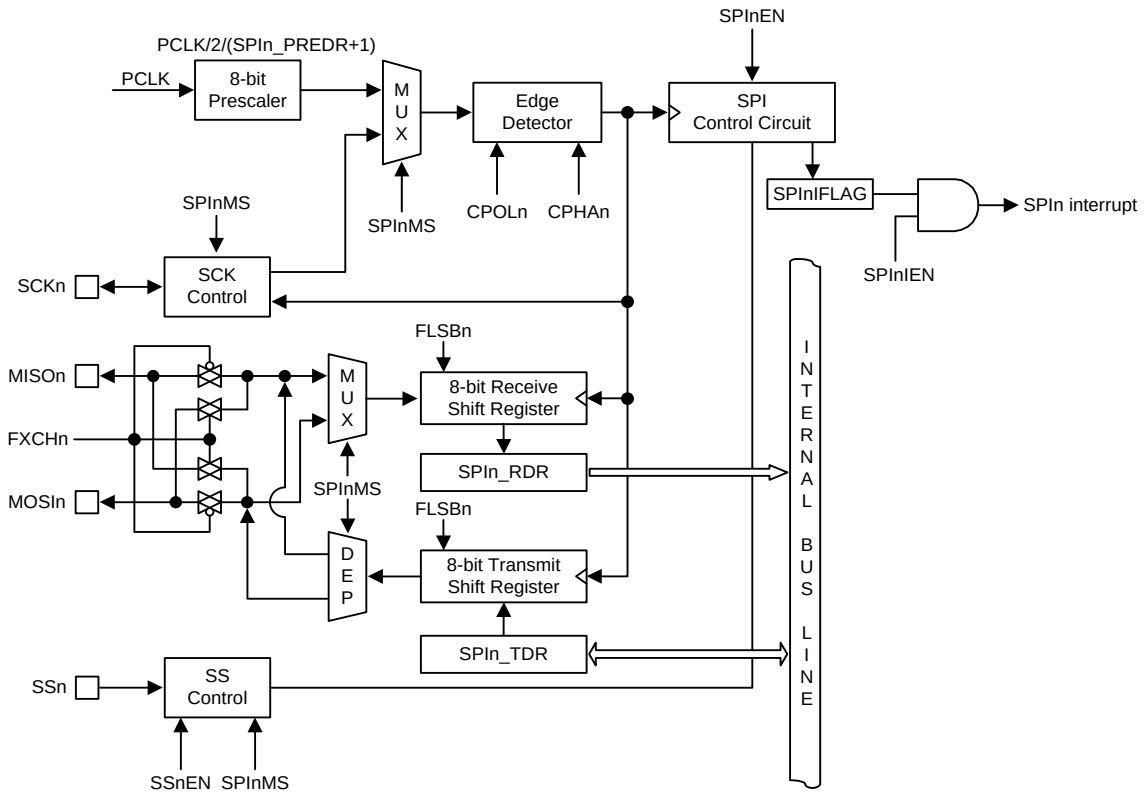
The SPI of the A31G336 features the following:

- Selectable between the master and slave operations
- Full-duplex and four-wire synchronous transfers supported
  - SS: Slave Select
  - SCK: Serial Clock
  - MOSI: Master Output, Slave Input
  - MISO: Master Input, Slave Output
- SPI clock speed and polarity adjustable
- Up to 8 MHz of SPI clock supported
- Selectable between MSB first transfer or LSB first transfer
- DMA Tx, Rx transfers

13.3 SPI 0/1 Functional Description

13.3.1 Block Diagram

Figure 66. SPIn Block Diagram (n = 0 and 1)



NOTES:

1. The clock frequency of the SCKn should be less than or equal to the PCLK/2.
2. Interrupts of the SPIn occur only when one byte transmission/reception finishes. So, the status (SPInMS and SSnHIGH) bits should be checked by software.
3. When DMA is used, at least 32 clocks of PCLK are required for the interval between data-receptions.

13.4 Pin Description for SPI 0/1

Table 67. Pins and External Signals for SPI (n = 0 and 1)

Pin Name	Type	Description
SSn	I/O	SPIn Slave select input/output
SCKn	I/O	SPIn Serial clock input/output
MOSIn	I/O	SPIn Serial data (Master output, Slave input)
MISOn	I/O	SPIn Serial data (Master input, Slave output)

13.5 Functional Description

When SPIn block is enabled (SPInEN = ’1’), the slave select (SSn) pin becomes active LOW input in slave mode operation if SSnEN bit is set to ‘1’. The SSn function is not automatically controlled in master mode operation even if SSnEN bit is set to ‘1’. (n = 0 and 1)

13.5.1 SPI Clock Formats and Timing

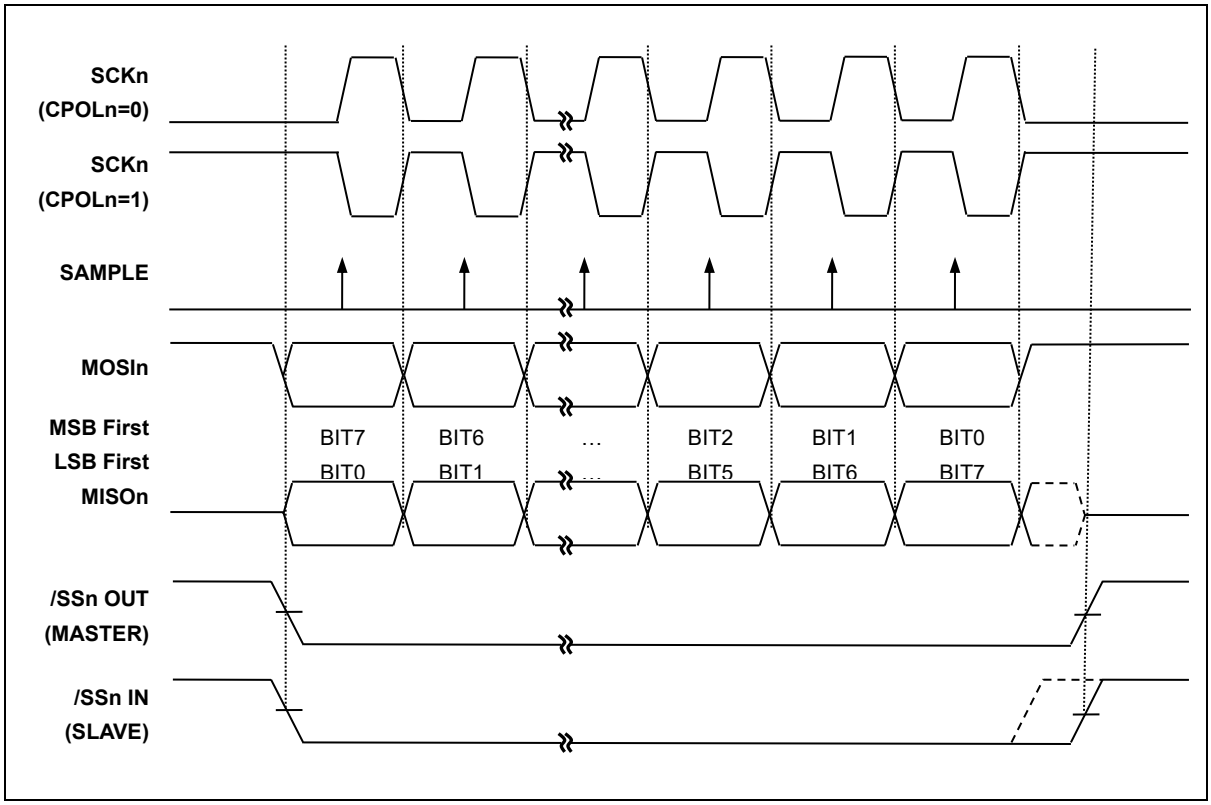
To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPIn has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. CPOLn selectively inserts an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data.

Table 68 shows the four combinations of CPOLn and CPHAn for SPIn. (n = 0 and 1)

Table 68. CPOL Functionality (n = 0 and 1)

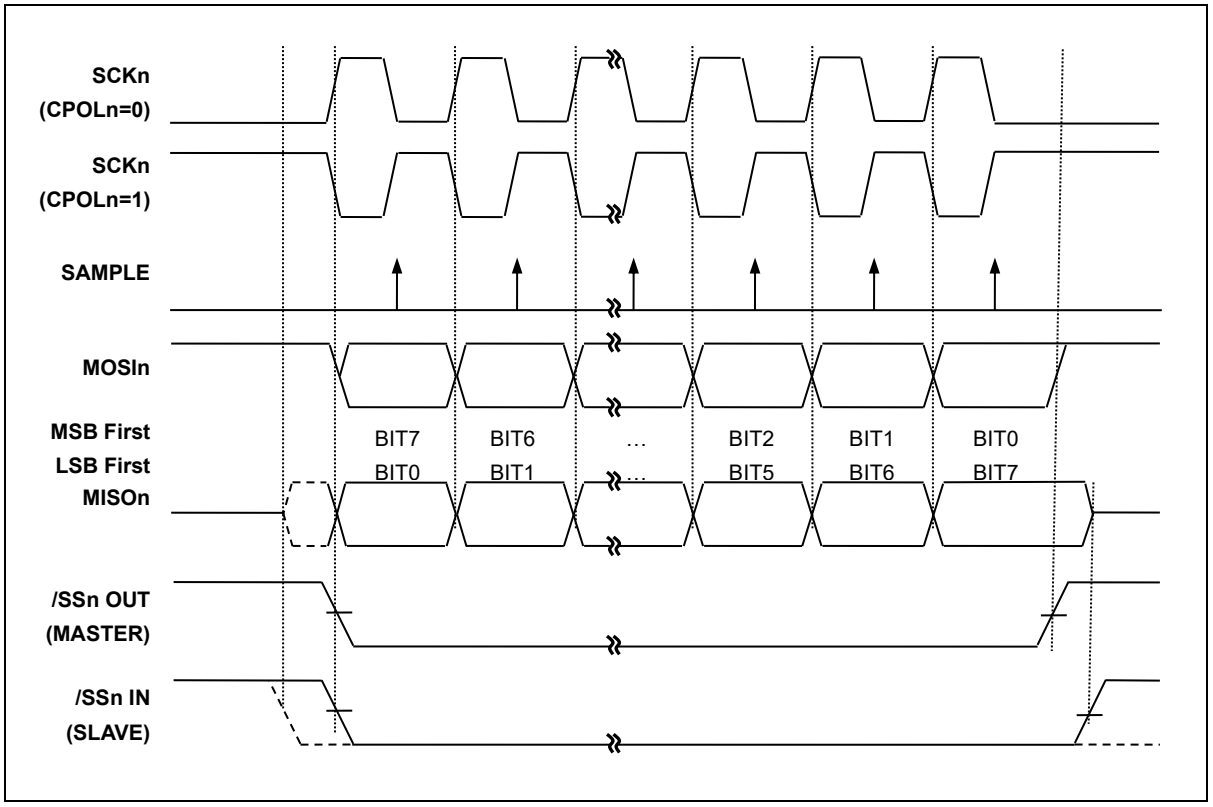
SPIn Mode	CPOLn	CPHAn	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

Figure 67. SPIn Clock Formats When CPHAn=0 (n = 0 and 1)



When CPHAn=0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low. The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the second SCKn edge, the SPIn shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively. (n = 0 and 1)

Figure 68. SPIn Clock Formats When CPHAn=1 (n = 0 and 1)



When CPHAn=1, the slave begins to drive its MISOn output when SSn goes active low, but the data is not defined until the first SCKn edge.

The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave. The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the third SCKn edge, the USART shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively.

When CPHAn=1, the slave's SSn input is not required to go to its inactive high level between transfers.

13.6 SPI 0/1 Registers

The base address and register map of the SPI 0/1 are described in the following tables:

Table 69. Base Address of SPI 0/1

Name	Base Address
SPI0	0x4000_5800
SPI1	0x4000_5880

Table 70. SPIn Register Map

Name	Offset	Type	Description	Reset Value	Reference
SPIn_CR	0x00	RW	SPIn Control Register	0x0000_0000	13.6.1
SPIn_SR	0x04	RW	SPIn Status Register	0x0000_0000	13.6.2
SPIn_RDR	0x08	RO	SPIn Receive Data Register	0x0000_0000	13.6.3
SPIn_TDR	0x0C	RW	SPIn Transmit Data Register	0x0000_0000	13.6.4
SPIn_PREDR	0x10	RW	SPIn Prescaler Data Register	0x0000_03FF	13.6.5

NOTE:

1. n = 0 and 1.

13.6.1 SPIn_CR: SPIn Control Register

SPI module should be configured properly before running.

The SPIn_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 and 1)

SPI0_CR=0x4000_5800, SPI1_CR=0x4000_5880

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			SPInEN	FLSBn	SPInMS	Reserved	SPInIEN	Reserved	CPOLn	CPHAn
-			0	0	0	-	0	-	0	0
-			RW	RW	RW	-	RW	-	RW	RW

7	SPInEN	SPIn Operation Control. **NOTE**: This bit should be set to '1' after setting the related registers.
	0	Disable SPIn operation.
	1	Enable SPIn operation.
6	FLSBn	Data Transmission sequence selection.
	0	MSB first.
	1	LSB first.
5	SPInMS	Master/Slave Selection.
	0	Slave mode.
	1	Master mode.
3	SPInIEN	SPIn Interrupt Enable.
	0	Disable SPIn interrupt.
	1	Enable SPIn interrupt.
1	CPOLn	Selects the clock polarity of SCK.
	0	SCK to 0 when idle.
	1	SCK to 1 when idle.
0	CPHAn	The CPOLn and this bit determine if data are sampled on the leading or the trailing edge of SCK.

CPOLn	CPHAn	Leading edge	Trailing edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

13.6.2 SPIn_SR: SPIn Status Register

The SPIn_SR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 and 1)

SPI0_SR=0x4000_5804, SPI1_SR=0x4000_5884

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			SPInIFLA	Reserved		SSnHIGH	Reserved		FXCHn	SSnEN
-			0	-		0	-		0	0
-			RW	-		RW	-		RW	RW

7	SPInIFLAG	SPIn Interrupt Flag.
	0	No request occurred.
	1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
4	SSnHIGH	This bit is set when the SSn pin goes high level during the pin is the corresponding function.
	0	No effect when ‘0’ is written.
	1	The SSn pin has gone from low level to high. This bit is cleared to ‘0’ when writing ‘1’.
1	FXCHn	SPIn Pin Function Exchange Control.
	0	No effect.
	1	Exchange MOSIn and MISOn function.
0	SSnEN	SSn Pin Operation Control.
	0	Disable SSn pin operation.
	1	Enable SSn pin operation. The corresponding SSn Pin should be configured to the alternate function.

13.6.3 SPIn_RDR: SPIn Receive Data Register

The SPIn_RDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 and 1)

SPI0_RDR=0x4000_5808, SPI1_RDR=0x4000_5888

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			RDATA[7:0]
-			0x00
-			R0

7 0	RDATA[7:0]	SPIn Receive Data.

13.6.4 SPIn_TDR: SPIn Transmit Data Register

The SPIn_TDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 and 1)

SPI0_TDR=0x4000_580C, SPI1_TDR=0x4000_588C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			TDATA[7:0]
-			0x00
-			RW

7 0	TDATA[7:0]	SPIn Transmit Data. When it is written a byte to this data register, the SPIn will start. **NOTE:** 1. The data to be transmitted should be written after all control registers are set.

13.6.5 SPIn_PREDR: SPIn Prescaler Data Register

The SPIn_PREDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 and 1)

SPI0_PREDR=0x4000_5810, SPI1_PREDR=0x4000_5890

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			PRED[9:0]
-			0x3FF
-			RW

9 0	PRED[9:0]	The value in this register is used to generate an SCK clock. SCKn clock: PCLK/2/(PRED[9:0] +1). The SCKn clock must be less than or equal to 16 MHz. The range is 0x00 to 0x3FF.

13.6.6 SPIn Register Map Summary

Table 71. SPIn Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPIn_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPInEN	FLSBn	SPInMS	Res.	SPInIEN	Res.	CPOLn	CPHAn
	Reset value																									0	0	0		0		0	0
0x04	SPIn_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPInIFLAG	Res.	Res.	SSnHIGH	Res.	Res.	FXCHn	SSnEN
	Reset value																									0			0			0	0
0x08	SPIn_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x0C	SPIn_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x10	SPIn_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[9:0]									
	Reset value																							1	1	1	1	1	1	1	1	1	1

NOTE:

1. n = 0 and 1.

14. I2C (Inter-Integrated Circuit)

14.1 Introduction

The I2C 0/1/2 (Inter-integrated Circuit) interface built in A31G336 satisfies the standard I2C communication protocol and is used for serial communication with internal and external devices via the I2C protocol.

Equipped with three units, it supports both master and slave modes and is capable of transmitting and receiving data in bytes by using interrupts or polling.

The I2C of the A31G336 operates in Standard mode (100 kHz), Fast mode (400 kHz) or Fast Plus mode (1 MHz) and supports General call.

It helps communicate with various peripherals that have the same bus type. To use the I2C, it is recommended to set the SCL and SDA pins to open-drain and then connect external pull-up resistors to render their output signals ‘HIGH’.

14.2 Main Features

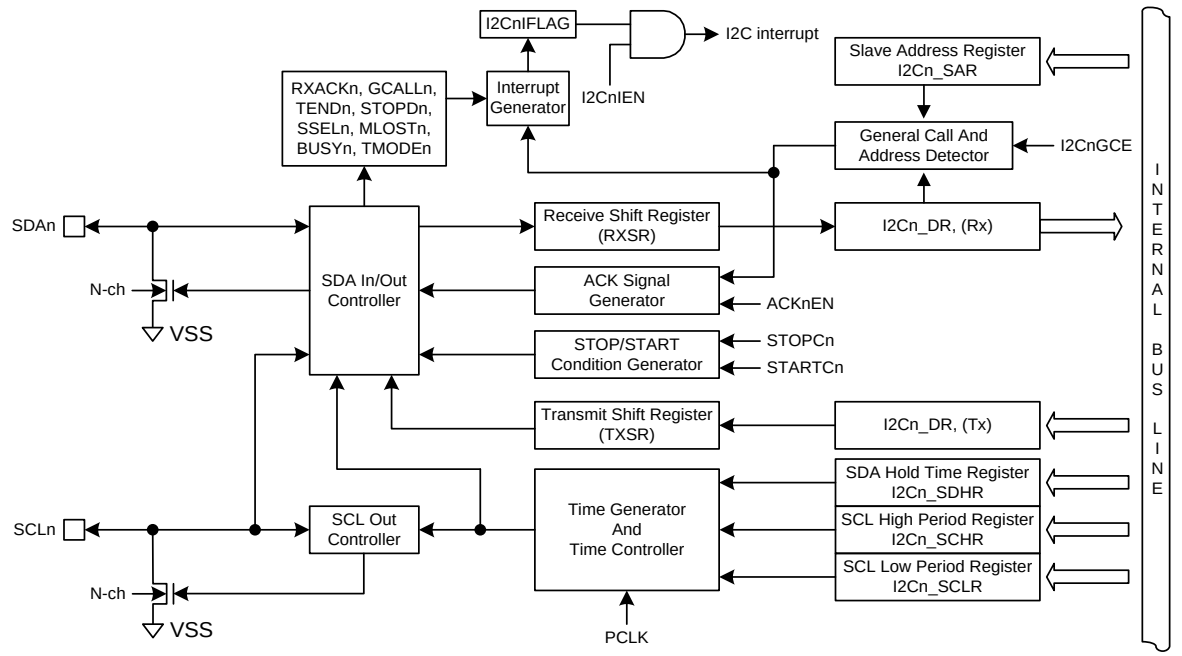
The I2C of the A31G336 0/1/2 features the following:

- Compatible with I2C bus standard
- Multi-master operation
- Up to 1 MHz data transfer read speed
- 7-bit address
- Two slave addresses supported
- Master and slave operations
- Bus busy detection

14.3 I2C 0/1/2 Block Diagram

Figure 69 shows a block diagram of the I2C module.

Figure 69. I2Cn Block Diagram (n = 0, 1 and 2)



NOTE:

1. When the corresponding port of I2C is a sub-function for SCLn/SDAn pin, the SCLn/SDAn pins are automatically set as N-channel open-drain outputs and the input latch is read when reading the pins. The corresponding pull-up resistor is determined by the control register.

14.4 Pin Description for I2C 0/1/2

Table 72. Pins and External Signals for I2Cn (n = 0, 1 and 2)

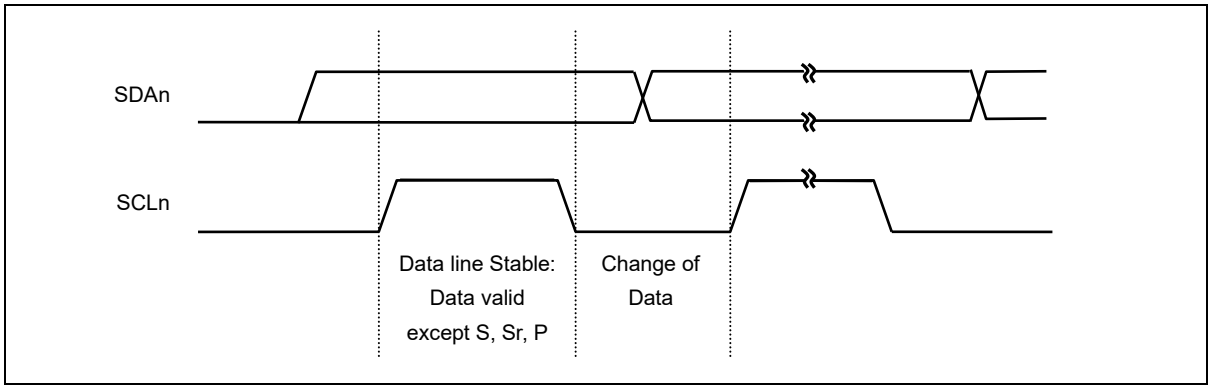
Pin Name	Type	Description
SCLn	I/O	I2C channel n Serial clock bus line (open-drain)
SDAn	I/O	I2C channel n Serial data bus line (open-drain)

14.5 I2C 0/1/2 Functional Description

14.5.1 I2C Bit Transfer

The data on the SDAn line must be stable during HIGH period of the clock, SCLn. The HIGH or LOW state of the data line can only change when the clock signal on the SCLn line is LOW. The exceptions are START(S), repeated START(Sr), and STOP(P) condition, where data line changes when clock line is high.

Figure 70. I2C Bus Bit Transfer (n = 0, 1 and 2)



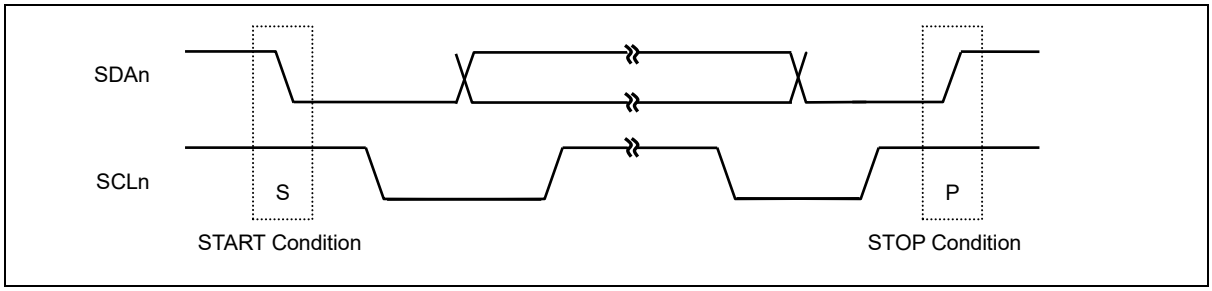
14.5.2 START/Repeated START/STOP

One master can issue a START (S) condition to detect other devices connected to the SCLn, SDAn lines that will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

- A high to low transition on the SDAn line while SCLn is high defines a START (S) condition.
- A low to high transition on the SDAn line while SCLn is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, i.e., the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays in busy mode. So, the START and repeated START conditions are functionally identical.

Figure 71. START and STOP Conditions (n = 0, 1 and 2)



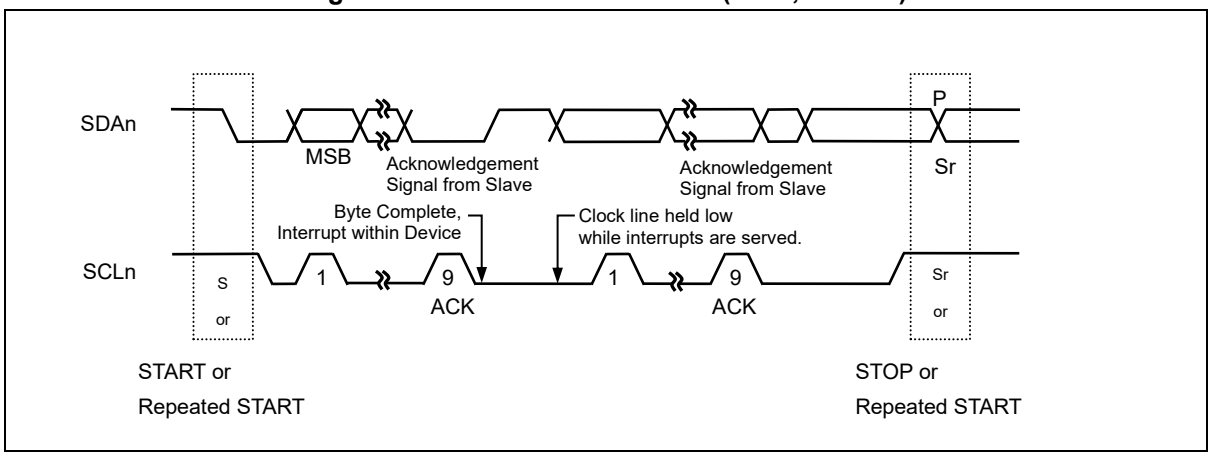
14.5.3 Data Transfer

Every byte on the SDAn line must be 8-bits long, but the number of bytes that can be transmitted per transfer is unlimited.

Each byte must be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave cannot receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCLn LOW to force the master into a wait state.

Data transfer then continues when the slave is ready for another byte of data and releases clock line SCLn.

Figure 72. I2C Bus Data Transfer (n = 0, 1 and 2)



14.5.4 Acknowledge

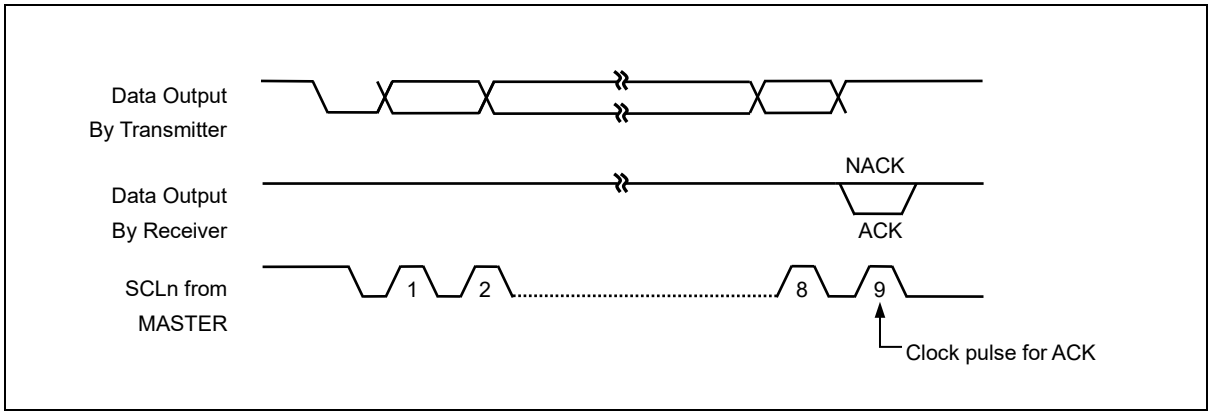
An acknowledge clock pulse is generated by the master. The transmitter releases the SDAn line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDAn line during the acknowledge clock pulse so that it remains stable at LOW during the HIGH period of this clock pulse.

When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it is performing some real time function, the data line must be left HIGH by the slave.

In addition, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDAn line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

Figure 73. I2C Bus Acknowledge (n = 0, 1 and 2)



14.5.5 Synchronization and Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCLn line. This means that a HIGH to LOW transition on the SCLn line will cause the devices concerned to start counting off their LOW period and it will hold the SCLn line in that state until the clock HIGH state is reached.

However, the LOW to HIGH transition of this clock may not change the state of the SCLn line if another clock is still within its LOW period. In this way, a synchronized SCLn clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period. A master may start a transfer only if the bus is free. Two or more masters may generate a START condition.

Arbitration takes place on the SDAn line, while the SCLn line is at the HIGH level, in such a way that a master that transmits a HIGH level, while another master that transmits a LOW level, will switch off its DATA output state because the level on the bus does not correspond to its own level.

Arbitration continues for many bits until a winning master gets the ownership of the I2C bus. Its first stage is a comparison of the address bits.

Figure 74. Clock Synchronization during Arbitration Procedure (n = 0, 1 and 2)

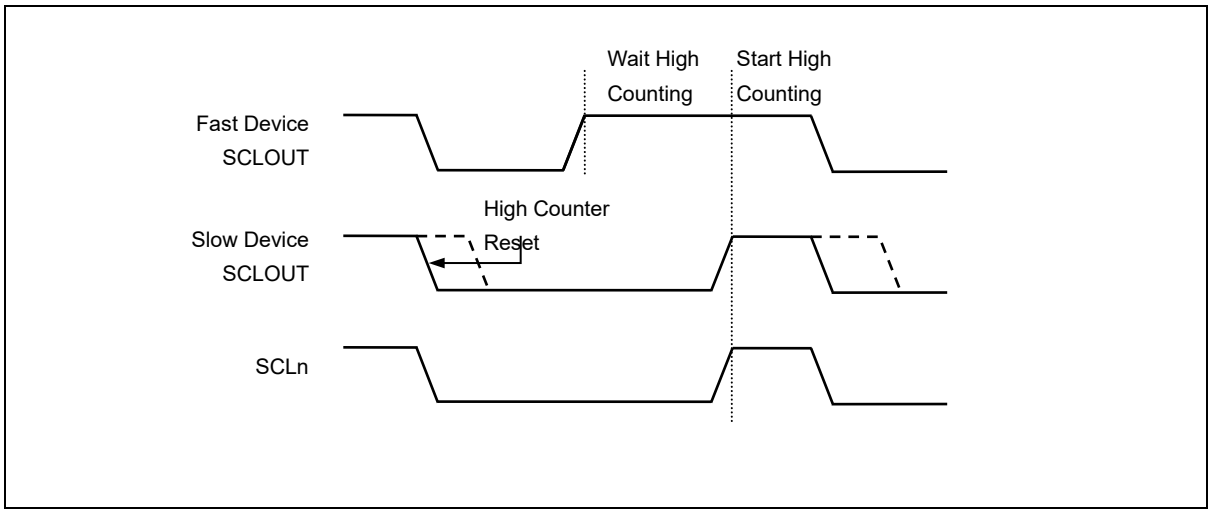
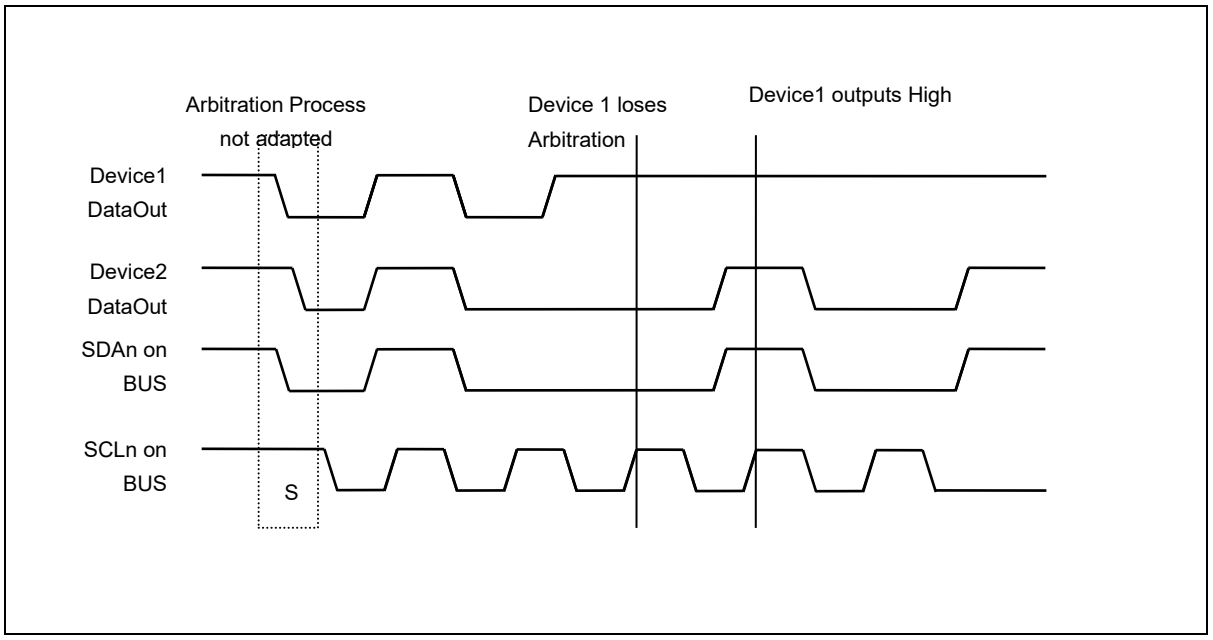


Figure 75. Arbitration Procedure Between Two Masters (n = 0, 1 and 2)



14.6 I2C Operation

The I2C is byte-oriented and interrupt-based. Interrupts are issued after all bus events except for the transmission of a START condition. Since I2C is interrupt based, the application software is free to carry on with other operations during an I2C byte transfer.

Note that when an I2C interrupt is generated, I2CnIFLAG flag in I2Cn_CR register is set, and it is cleared when all interrupt source bits in the I2Cn_ST register are cleared to '0'. When I2C interrupt occurs, the SCLn line is held at LOW until all interrupt source bits in I2Cn_ST register are cleared to '0'. When the I2CnIFLAG flag is set, the I2Cn_ST contains a value indicating the current state of the I2C bus. According to the value in I2Cn_ST, software can decide what to do next.

I2C can operate in 4 modes: master/slave, transmitter/receiver. The operating mode is configured by a winning master.

A more detailed explanation follows below. (n = 0, 1 and 2)

14.6.1 Master Transmitter

To operate I2C as a master transmitter, follow the recommended steps below.

1. Enable I2C by setting I2CnEN bit in I2Cn_CR. This provides a main clock to the peripheral.
2. Load SLA+W into the I2Cn_DR, where SLA is the address of slave device and W is the transfer direction from the viewpoint of master. For master transmitter, W is '0'. Note that I2Cn_DR is used for both address and data.
3. Configure baud rate by writing desired value to both I2Cn_SCLR and I2Cn_SCHR for the Low and High period of SCLn line.
4. Configure the I2Cn_SDHR to decide when SDAn changes value from falling edge of SCLn. If SDA should change in the middle of SCLn LOW period, load half the value of I2Cn_SCLR to the I2Cn_SDHR.
5. Set the STARTCn bit in I2Cn_CR. This transmits a START condition. Also, configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in I2Cn_DR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of receiving ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in I2Cn_ST is set, and I2C waits in idle state or can be operated as an addressed slave.

To operate as a slave when the MLOSTn bit in I2Cn_ST is set, the ACKnEN bit in I2Cn_CR must be set and the received 7-bit address must match the SLAn bits in I2Cn_SAR1/2. In this case, I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. The reason for this is to decide whether I2C should continue serial transfer or stop communication. The following steps continue, assuming that I2C does not lose mastership during the first data transfer.

I2C (Master) can choose one of the following cases regardless of receiving ACK signal from slave.

A. Master receives ACK signal from slave, so continues data transfer since slave can receive more data from master. In this case, load data to transmit to I2Cn_DR.

B. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPC bit in I2Cn_CR.

C. Master transmits repeated START condition without checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in I2Cn_CR.

After doing any of the actions above, clear all interrupt source bits in I2Cn_ST to '0' to release SCLn line. In case of A, move to step 7. In case of B, move to step 9 to handle STOP interrupt. In the case of C, move to step 6 after transmitting the data in I2Cn_DR, and if transfer direction bit is '1', go to master receiver section.

7. 1-Byte of data is transmitted. During data transfer, bus arbitration continues.

8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCLn LOW. When I2C loses bus mastership while transmitting data to arbitrate other masters, the MLOSTn bit in I2Cn_ST is set. If then, I2C waits in idle state. When the data in I2Cn_DR is transmitted completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases regardless of receiving ACK signal from slave.

   A. Master receives ACK signal from slave, so continues data transfer since slave can receive more data from master. In this case, load data to transmit to I2Cn_DR.

   B. Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in I2Cn_CR.

   C. Master transmits repeated START condition without checking ACK signal. In this case, load SLA+R/W into the I2Cn_DR and set the STARTCn bit in I2Cn_CR.

   After doing any of the actions above, clear all interrupt source bits in I2Cn_ST to '0' to release SCL line. In case of A, move to step 7. In case of B, move to step 9 to handle STOP interrupt. In case of C, move to step 6 after transmitting the data in I2CDR, and if transfer direction bit is '1', go to master receiver section.

9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write "0xff" to I2Cn_ST. After this, I2C enters an idle state.

14.6.2 Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

1. Enable I2C by setting I2CnEN bit in I2Cn_CR. This provides a main clock to the peripheral.
2. Load SLA+R into the I2Cn_DR, where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. Note that I2Cn_DR is used for both address and data.
3. Configure baud rate by writing desired value to both I2Cn_SCLR and I2Cn_SCHR for the Low and High period of SCLn line.
4. Configure the I2Cn_SDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of I2Cn_SCLR to the I2Cn_SDHR.
5. Set the STARTCn bit in I2Cn_CR. This transmits a START condition. Also, configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in I2Cn_DR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of receiving ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in I2Cn_ST is set, and I2C waits in idle state or can be operated as an addressed slave.

   To operate as a slave when the MLOSTn bit in I2Cn_ST is set, the ACKnEN bit in I2Cn_CR must be set, and the received 7-bit address must equal to the SLAn bits in I2Cn_SAR1/2. In this case, I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. The reason for this is to decide whether I2C should continue serial transfer or stop communication. The following steps continue, assuming that I2C does not lose mastership during the first data transfer.

   I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

   A. Master receives ACK signal from slave, so continues data transfer since slave can prepare and transmit more data to master. Configure ACKnEN bit in I2Cn_CR to decide whether I2C should Acknowledges the next data to be received or not.

   B. Master stops data transfer since it receives no ACK signal from slave. In this case, set the STOPCn bit in I2Cn_CR.

   C. Master transmits repeated START condition due to lack of ACK signal from slave. In this case, load SLA+R/W into the I2Cn_DR and set STARTCn bit in I2Cn_CR.

   After doing any of the actions above, clear all interrupt source bits in I2Cn_ST to '0' to release SCLn line. In case of A, move to step 7. In case of B, move to step 9 to handle STOP interrupt. In case of C, move to step 6 after transmitting the data in I2Cn_DR, and if transfer direction bit is '0', go to master transmitter section.

7. 1-Byte of data is received.
8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCLn LOW. When 1-Byte of data is received completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases according to the RXACKn flag in I2Cn_ST.

   A. Master continues receiving data from slave. To do this, set ACKnEN bit in I2Cn_CR to acknowledge the next data to be received.

   B. Master wants to terminate data transfer when it receives the next data by not generating ACK signal. This can be done by clearing ACKnEN bit in I2Cn_CR.

   C. Since no ACK signal is detected, the master terminates data transfer. In this case, set the STOPCn bit in I2Cn_CR.

   D. No ACK signal is detected, and the master transmits repeated START condition. In this case, load SLA+R/W into the I2Cn_DR and set the STARTCn bit in I2Cn_CR.

   After doing any of the actions above, clear all interrupt source bits in I2Cn_ST to ‘0’ to release SCLn line. In the case of A and B, move to step 7. In case of C, move to step 9 to handle STOP interrupt. In case of D, move to step 6 after transmitting the data in I2Cn_DR, and if transfer direction bit is ‘0’, go to master transmitter section.
9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write “0xff” value to I2Cn_ST. After this, I2C enters an idle state.

14.6.3 Slave Transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

1. If the operating clock (HCLK) of the system is slower than that of SCLn, load value 0x00 into I2Cn_SDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of HCLK where SDAH is multiple of number of HCLK coming from I2Cn_SDHR. When the hold time of SDAn is longer than the period of HCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting I2CnIEN bit and I2CnEN bit in I2Cn_CR. This provides a main clock for the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLAn bits in I2Cn_SAR1/2. If the GCALLnEN bit in I2Cn_SAR1/2 is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the address received does not match SLAn bits in I2CnSAR, I2C enters idle state, i.e., waits for another START condition. Otherwise, if the address equals to SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address matches SLAn bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs, load transmit data to I2Cn_DR and clear all interrupt source bits in I2Cn_ST to '0' to release SCLn line.
5. 1-Byte of data is transmitted.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of receiving ACK signal from master. Slave can select one of the following cases.

   A. No ACK signal is detected and I2C waits for STOP or repeated START condition.

   B. ACK signal from the master is detected. Load data to transmit into I2Cn_DR.

   After doing any of the actions above, clear all interrupt source bits in I2Cn_ST to '0' to release SCLn line. In the case of A, move to step 7 to terminate communication. In case of B, move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for the slave transmitter function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write "0xff" to I2Cn_ST. After this, I2C enters an idle state.

14.6.4 Slave Receiver

To operate I2C in slave receiver, follow the recommended steps below.

1. If the operating clock (HCLK) of the system is slower than that of SCLn, load value 0x00 into I2Cn_SDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of HCLK where SDAH is multiple of number of HCLK coming from I2Cn_SDHR. When the hold time of SDAn is longer than the period of HCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting I2CnIEN bit in I2Cn_CR. This provides a main clock for the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2CSAR. If the GCALLnEN bit in I2Cn_SAR1/2 is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not match SLAn bits in I2Cn_SAR1/2, I2C enters idle state i.e., waits for another START condition. Otherwise, if the address matches SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to SLA bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs and I2C is ready to receive data, clear all interrupt source bits in I2Cn_ST to '0' to release SCLn line.
5. 1-Byte of data is received.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of receiving ACK signal from master. Slave can select one of the following cases.

   A. No ACK signal is detected (ACKnEN=0) and I2C waits for STOP or repeated START condition.

   B. ACK signal is detected (ACKnEN=1) and I2C can continue to receive data from master.

   After doing any of the actions above, clear all interrupt source bits in I2Cn_ST to '0' to release SCLn line. In the case of A, move to step 7 to terminate communication. In case of B, move to step 5. In either case, a repeated START condition can be detected. For that case, move to step 4.
7. This is the final step for the slave receiver function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear I2Cn_ST, write "0xff" to I2Cn_ST. After this, I2C enters an idle state.

14.7 I2C Registers

The base address and register map of the I2C module are described in the following tables:

Table 73. Base Address of I2Cn Interface

Name	Base Address
I2C0	0x4000_4800
I2C1	0x4000_4900
I2C2	0x4000_4A00

Table 74. I2C Register Map

Name	Offset	Type	Description	Reset Value	Reference
I2Cn_CR	0x00	RW	I2Cn Control Register	0x0000_0000	14.7.1
I2Cn_ST	0x04	RW	I2Cn Status Register	0x0000_0000	14.7.2
I2Cn_SAR1	0x08	RW	I2Cn Slave Address Register 1	0x0000_0000	14.7.3
I2Cn_SAR2	0x0C	RW	I2Cn Slave Address Register 2	0x0000_0000	14.7.4
I2Cn_DR	0x10	RW	I2Cn Data Register	0x0000_0000	14.7.5
I2Cn_SDHR	0x14	RW	I2Cn SDA Hold Time Register	0x0000_0001	14.7.6
I2Cn_SCLR	0x18	RW	I2Cn SCL Low Period Register	0x0000_003F	14.7.7
I2Cn_SCHR	0x1C	RW	I2Cn SCL High Period Register	0x0000_003F	14.7.8

NOTE:

1. n = 0, 1, and 2

14.7.1 I2Cn_CR: I2Cn Control Register

The register can be set to configure I2C operation mode activate I2C transactions.

The I2Cn_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_CR=0x4000_4800, I2C1_CR=0x4000_4900, I2C2_CR=0x4000_4A00

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			I2CnEN	TXDLYENBn	I2CnIEN	I2CnIFLAG	ACKnEN	IMASTERn	STOPCn	STARTCn
-			0	0	0	0	0	0	0	0
.			RW	RW	RW	RO	RW	RO	RW	RW

7	I2CnEN	Activate I2Cn Block.
		0 Disable I2Cn block.
		1 Enable I2Cn block.
6	TXDLYENBn	I2Cn_SDHR Register Control.
		0 Enable I2Cn_SDHR register.
		1 Disable I2Cn_SDHR register.
5	I2CnIEN	I2Cn Interrupt Enable.
		0 Disable I2Cn interrupt.
		1 Enable I2Cn interrupt.
4	I2CnIFLAG	I2Cn Interrupt Flag. This bit is cleared when all interrupt source bits in the I2Cn_ST register are cleared to '0'.
		0 No request occurred.
		1 Request occurred.
3	ACKnEN	Controls ACK signal generation at ninth SCL period.
		0 No ACK signal is generated. (SDA = 1)
		1 ACK signal is generated. (SDA = 0)
		NOTE: 1. ACK signal is output (SDA = 0) for the following three cases. A. When received the address packet is equal to SLAn[6:0] bits in I2Cn_SAR1/I2Cn_SAR2 register. B. When received, the address packet is equal to value 0x00 with GCALLn enabled. C. When I2Cn operates as a receiver (master or slave)
2	IMASTERn	Represents Operation Mode of I2Cn. This bit is cleared to '0' on STOP condition.
		0 I2Cn is in slave mode.
		1 I2Cn is in master mode.
1	STOPCn	STOP Condition Generation When I2Cn is master.
		0 No effect.
		1 Generate STOP condition.
0	STARTCn	START Condition Generation When I2Cn is master.
		0 No effect.
		1 Generate START or Repeated START condition.

14.7.2 I2Cn_ST: I2Cn Status Register

The I2Cn_ST register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_ST=0x4000_4804, I2C1_ST=0x4000_4904, I2C2_ST=0x4000_4A04

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RW	RW	RO	RW

7	GCALLn	This bit has different meaning depending on whether I2C is master or slave. When I2C is a master, this bit represents whether it received AACK (address ACK) from slave.
	0	No AACK is received. (Master mode)
	1	AACK is received (Master mode). It may be set to '1' after address transmission.
	When I2C is a slave, this bit is used to indicate general call.	
	0	General call address is not detected. (Slave mode)
	1	General call address is detected. (Slave mode)
6	TENDn	This bit is set when 1-byte of data is transferred completely.
	0	1 byte of data is not completely transferred.
	1	1 byte of data is completely transferred.
5	STOPDn	This bit is set when a STOP condition is detected.
	0	A STOP condition is not detected.
	1	A STOP condition is detected.
	NOTE: A STOP condition is not detected on unaddressed slaves.	
4	SSELn	This bit is set when I2C is addressed by another master.
	0	I2C is not selected as a slave.
	1	I2C is addressed by another master and acts as a slave.
3	MLOSTn	This bit represents the result of bus arbitration in master mode.
	0	I2C maintains bus mastership.
	1	I2C has lost bus mastership during arbitration process.
2	BUSYn	This bit reflects bus status.
	0	I2C bus is idle, so a master can issue a START condition.
	1	I2C bus is busy.
1	TMODEn	This bit is used to indicate whether I2C is transmitter or receiver.
	0	I2C is a receiver.
	1	I2C is a transmitter.
0	RXACKn	This bit shows the state of ACK signal.
	0	No ACK is received.
	1	ACK is received at ninth SCL period.

NOTES:

1. The GCALLn, TENDn, STOPDn, SSELn, and MLOSTn bits can be source of interrupt.
2. When an I2C interrupt occurs except for DEEP SLEEP mode, the SCL line is held low. To release SCL, Clear to '0' all interrupt source bits in I2Cn_ST register.
3. The GCALLn, TENDn, STOPDn, SSELn, MLOSTn, and RXACKn bits are cleared when '1' is written to the corresponding bit.

14.7.3 I2Cn_SAR1: I2Cn Slave Address Register 1

The I2Cn_SAR1 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_SAR1=0x4000_4808, I2C1_SAR1=0x4000_4908, I2C2_SAR1=0x4000_4A08

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1	0
Reserved			SLAn[6:0]	GCALLnEN
-			0	0
-			RW	RW

7 1	SLAn[6:0]	These bits configure the slave address 1 in slave mode.
0	GCALLnEN	This bit decides whether I2Cn allows general call address 1 or not in I2Cn slave mode.
	0	Ignore general call address 1.
	1	Allow general call address 1.

14.7.4 I2Cn_SAR2: I2Cn Slave Address Register 2

The I2Cn_SAR2 register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_SAR2=0x4000_480C, I2C1_SAR2=0x4000_490C, I2C2_SAR2=0x4000_4A0C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1	0
Reserved			SLAn[6:0]	GCALLnEN
-			0	0
-			RW	RW

7 1	SLAn[6:0]	These bits configure the slave address 2 in slave mode.
0	GCALLnEN	This bit decides whether I2Cn allows general call address 2 or not in I2Cn slave mode.
	0	Ignore general call address 2.
	1	Allow general call address 2.

14.7.5 I2Cn_DR: I2Cn Data Register

The I2Cn_DR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_DR=0x4000_4810, I2C1_DR=0x4000_4910, I2C2_DR=0x4000_4A10

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			DATA[7:0]
-			0
-			RW

7 0	DATA[7:0]	The I2Cn_DR Transmit buffer and Receive buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the I2Cn_DR register. Reading the I2Cn_DR register returns the contents of the Receive Buffer.

14.7.6 I2Cn_SDHR: I2Cn SDA Hold Time Register

The I2Cn_SDHR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_SDHR=0x4000_4814, I2C1_SDHR=0x4000_4914, I2C2_SDHR=0x4000_4A14

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			HLDT[11:0]
-			0x001
-			RW

11 0	HLDT[11:0]	This register is used to control SDA output timing from the falling edge of SCL. Note that SDA is changed after tPCLK X (I2Cn_SDHR+2). In master mode, load half the value of I2Cn_SCLR to this register to make SDA switch in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after tPCLK X (I2Cn_SDHR+2) in master mode. So, to ensure proper operation in slave mode, the value tPCLK X (I2Cn_SDHR + 3) must be smaller than the period of SCL.

14.7.7 I2Cn_SCLR: I2Cn SCL Low Period Register

The I2Cn_SCLR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_SCLR=0x4000_4818, I2C1_SCLR=0x4000_4918, I2C2_SCLR=0x4000_4A18

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			SCLL[11:0]
-			0x03F
-			RW

11 0	SCLL[11:0]	This register defines the low period of SCL in master mode. The base clock is PCLK and the period is calculated by the formula: tPCLK × (4 × I2Cn_SCLR + 3) where tPCLK is the period of PCLK.

14.7.8 I2Cn_SCHR: I2Cn SCL High Period Register

The I2Cn_SCHR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.
(n = 0, 1, and 2)

I2C0_SCHR=0x4000_481C, I2C1_SCHR=0x4000_491C, I2C2_SCHR=0x4000_4A1C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			SCLH[11:0]
-			0x03F
-			RW

11 0	SCLH[11:0]	This register defines the high period of SCL in master mode. The base clock is PCLK and the period is calculated by the formula: tPCLK × (4 × I2Cn_SCHR + 3) where tPCLK is the period of PCLK.

14.7.9 I2C Register Map Summary

Table 75. I2C Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	I2Cn_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2CnEN	TXDLYENBn	I2CnIEN	I2CnIFLAG	ACKnEN	IMASTERn	STOPCn	STARTCn
	Reset value																									0	0	0	0	0	0	0	0
0x04	I2Cn_ST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
	Reset value																									0	0	0	0	0	0	0	0
0x08	I2Cn_SAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLAn[6:0]							GCALLnEN
	Reset value																									0	0	0	0	0	0	0	0
0x0C	I2Cn_SAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLAn[6:0]							GCALLnEN
	Reset value																									0	0	0	0	0	0	0	0
0x10	I2Cn_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x14	I2Cn_SDHR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HLDT[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	1
0x18	I2Cn_SCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCLL[11:0]											
	Reset value																					0	0	1	1	1	1	1	1	1	1	1	1
0x1C	I2Cn_SCHR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCLH[11:0]											
	Reset value																					0	0	1	1	1	1	1	1	1	1	1	1

15. High-Speed 12-bit ADC

15.1 Introduction

ADC (Analog-to-Digital Converter) of A31G336 allows conversion of an analog input signal to a corresponding 12-bit digital value. Its A/D module has sixteen analog inputs as shown in Figure 76. The output of the multiplexer is the input into the converter, which generates the result through successive approximation.

The A/D module has seven registers such as a control register (ADC_CR), a data register (ADC_DR), a prescaler data register (ADC_PREDR), an oversampling control register (ADC_OVSCR), an interrupt enables and status register (ADC_IESR), a sampling time register (ADC_SAMR), and a channel selection register (ADC_CHSELR). The A/D module supports single, sequential, and continuous conversion modes.

15.2 Main Features

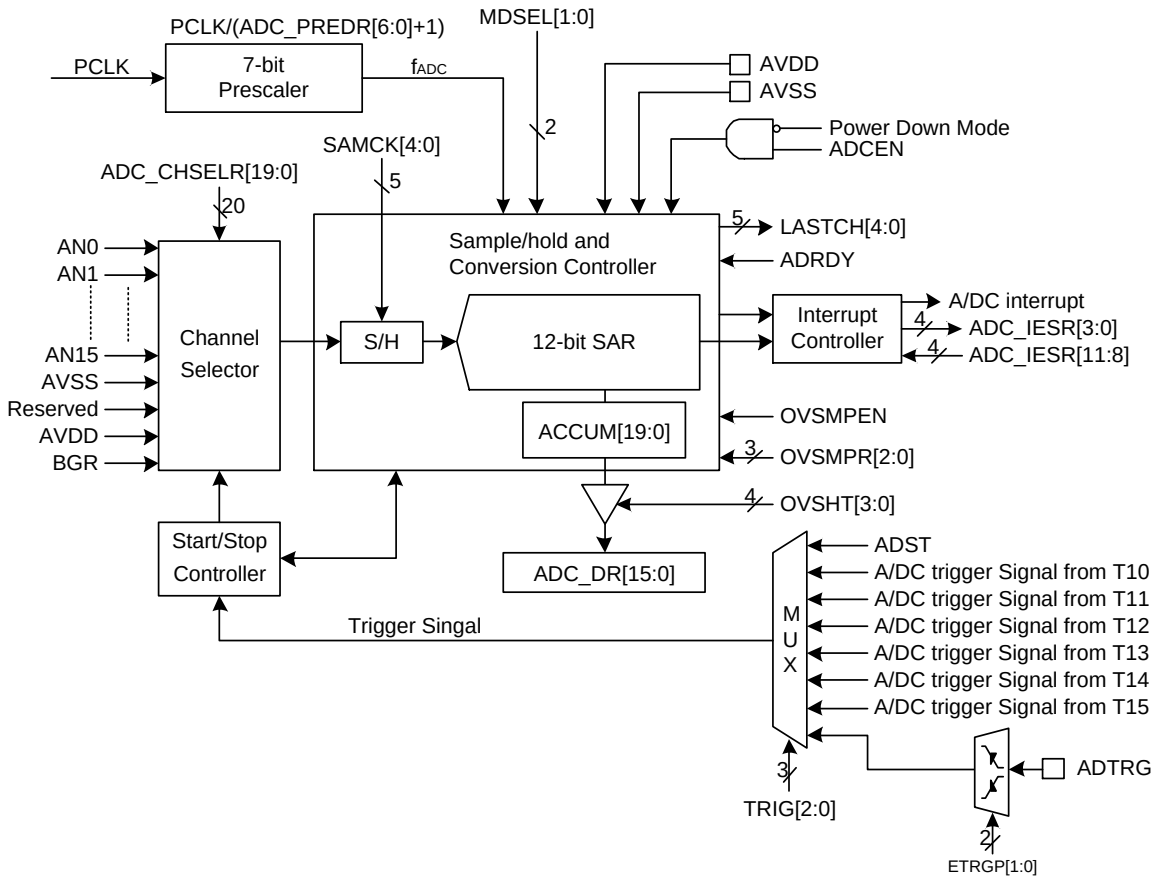
Main features of the ADC module are listed below:

- 16-channel of analog inputs
- Software (ADST), Timer trigger (T10/11/12/13/14/15 ADC trigger signal), and external trigger support
- Conversion time: Up to 1us with 14 clocks + at least 4 sample/hold clocks
- 7-bit Prescaler and 16-bit data registers
- Up to 256 over sampling
- Single, sequential, and continuous conversion mode

15.3 ADC Block Diagram

Figure 76 shows a block diagram of the 12-bit ADC.

Figure 76. 12-Bit ADC Block Diagram



NOTE:

1. The external trigger input, ADTRG, signal should be slower than a system clock frequency.

15.4 Pin Description for ADC

Pin Name	Type	Description
AN0	A	ADC Input 0
AN1	A	ADC Input 1
AN2	A	ADC Input 2
AN3	A	ADC Input 3
AN4	A	ADC Input 4
AN5	A	ADC Input 5
AN6	A	ADC Input 6
AN7	A	ADC Input 7
AN8	A	ADC Input 8
AN9	A	ADC Input 9
AN10	A	ADC Input 10
AN11	A	ADC Input 11
AN12	A	ADC Input 12
AN13	A	ADC Input 13
AN14	A	ADC Input 14
AN15	A	ADC Input 15
AVSS	AP	Analog GND
AVDD	AP	Analog Power

NOTE:

1. A: Analog, AP: Analog Power.

15.5 ADC Functional Description

15.5.1 ADC Enable/Disable Control

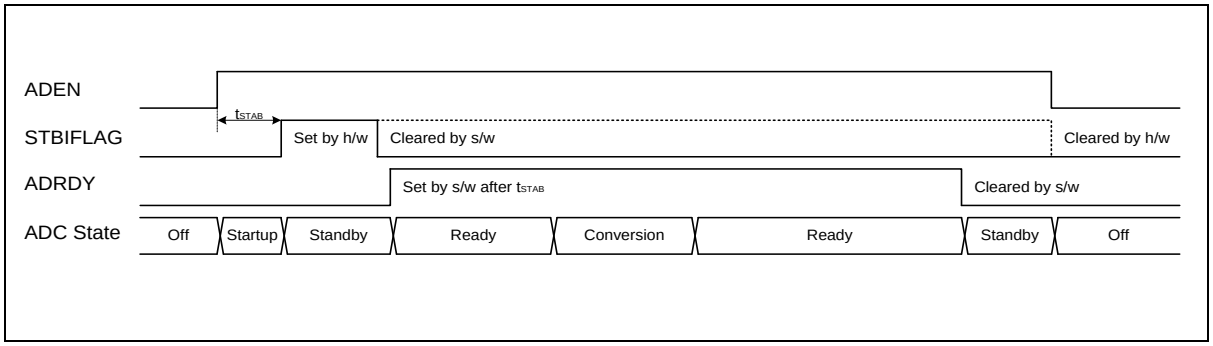
A/D converter needs a stabilization time of about 16/fADC, tSTAB, before it starts converting. The following procedure is required for an ADC conversion.

1. Set the ADCEN bit of ADC_CR register to ‘1’ for enabling ADC module operation.
2. Wait until the STBIFLAG bit of ADC_IESR register is set to ‘1’. The bit is set after the ADC stabilization time.
3. Set the ADRDY bit of ADC_CR register to ‘1’ for converting.

The following procedure is required to disable the ADC module.

1. Clear the ADRDY bit of ADC_CR register to ‘0’ for conversion stop.
2. Clear the ADCEN bit of ADC_CR register to ‘0’ for disabling ADC module operation.

Figure 77. 12-Bit ADC Enable/Disable Timing Chart



15.5.2 Channel Selection

The ADC has 16 input channels from GPIO pins and three internal channels. It is possible to convert a single channel or to scan a sequence of channels. The channels to be converted should be programmed in the ADC_CHSELR register. The conversion order is always from AN0 to AN15.

15.5.3 ADC Conversion Timing

Conversion clock of the ADC is the sum of sampling and converting. The sampling clock is equal to the ADC_SAMR register + 2 and converting clock is always 14 clocks. The ADC clock should be set appropriately by the ADC_PREDR register according to the AVDD voltage. In addition, the ADC_SAMR register must be set carefully for accurate conversion.

If the ADC_SAMR register has value of “0x2”, the sampling clock is four clocks. Since the converting clock is always 14 clocks, the conversion clock of the ADC is calculated as shown in the following:

Conversion clock = (ADC_SAMR + 2) + 14 [clocks]

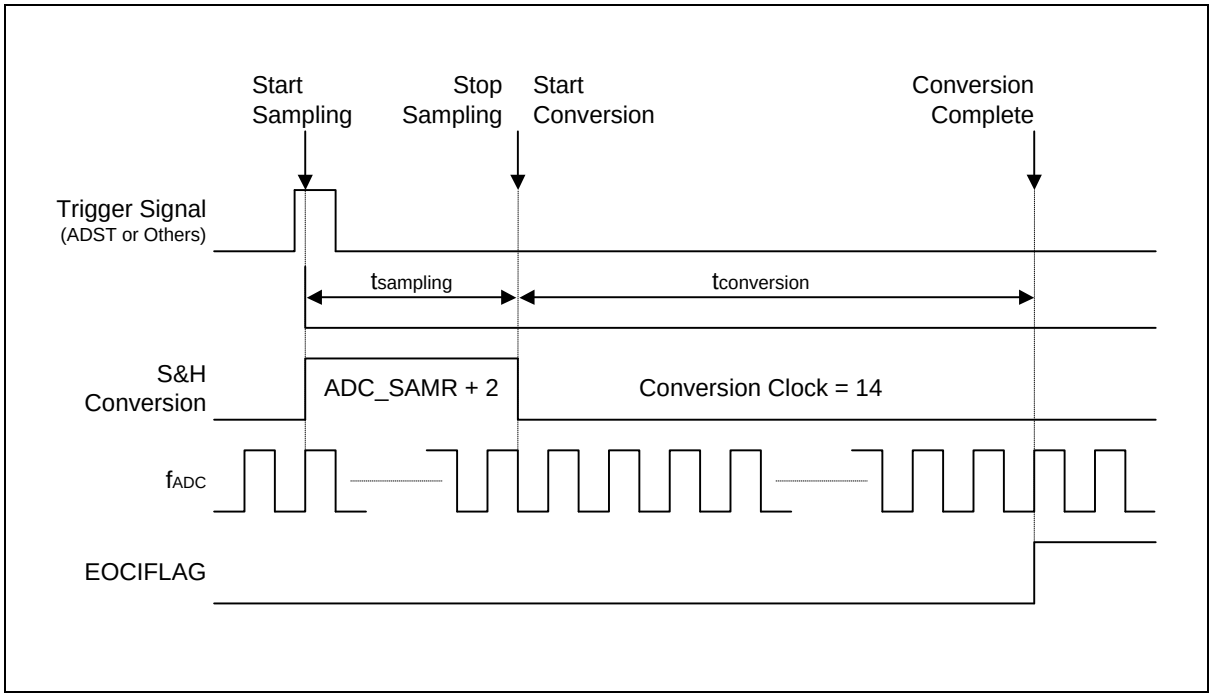
Table 76. ADC Frequency Set according to AVDD

AVDD Range	Max. fADC	ADC_PREDR (Ex: PCLK = 32 MHz)
3.0 V < AVDD	Up to 16 MHz	0x1 or more
2.7 V < AVDD	Up to 8 MHz	0x3 or more

NOTE:

1. On low or high temperature, set the ADC frequency lower than the above table.

Figure 78. High-Speed ADC Conversion Timing Chart



15.5.4 ADC Conversion Mode

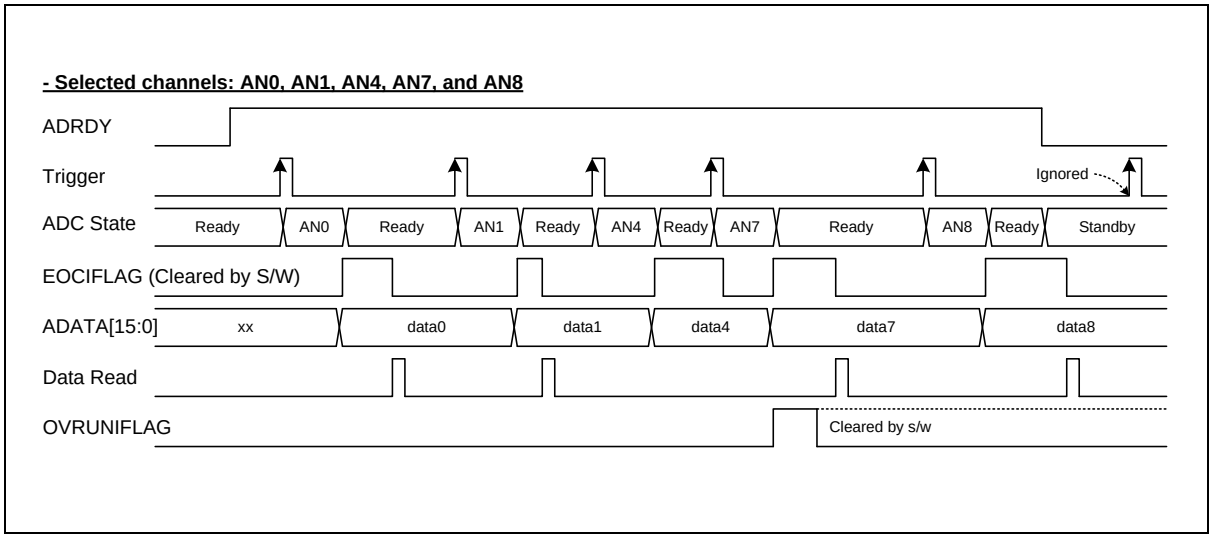
There are three modes for ADC such as Single conversion mode, Sequential conversion mode and Continuous conversion mode. A mode is selected by the MDSEL[1:0] bits of ADC_CR register.

15.5.4.1 Single Conversion Mode

The ADC converts one of the selected channels in order every trigger signal during single conversion mode. Analog input signal is selected by ADC_CHSELR register.

The end of conversion interrupt flag, the EOCIFLAG bit of ADC_IESR register, is set to ‘1’ as soon as a new conversion data result is available. The EOCIFLAG bit is cleared by software by writing ‘1’ to it. An ADC data overrun interrupt flag is set to ‘1’ if a trigger finishes a new conversion while the previous conversion data is not read.

Figure 79. Example of Single Conversion Mode

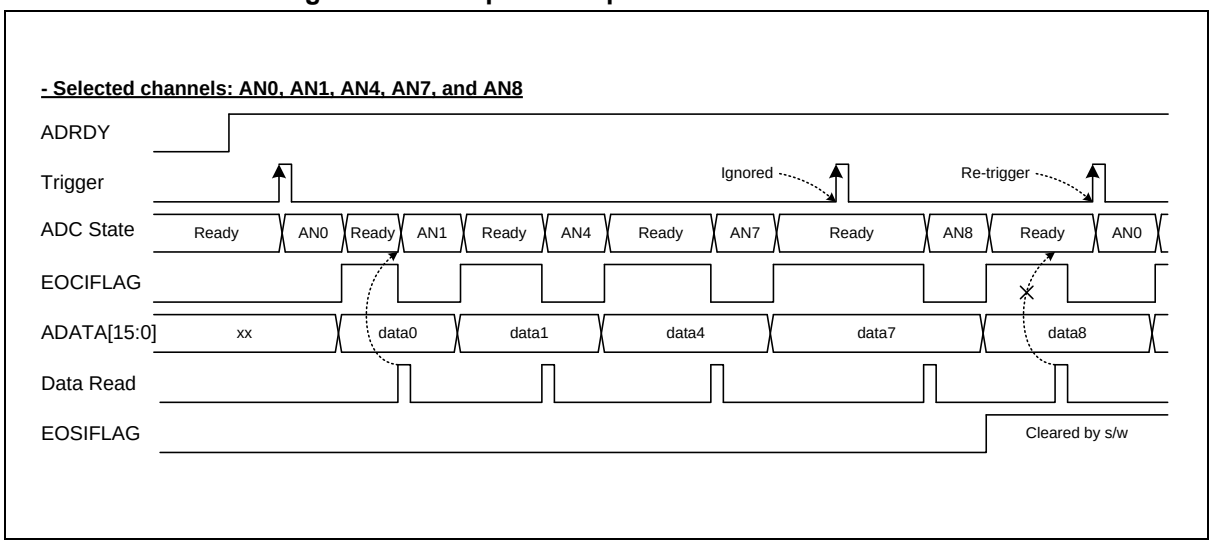


15.5.4.2 Sequential Conversion Mode

The ADC converts all selected channels in order by a trigger signal during Sequential conversion mode. All trigger signals are ignored during a sequence procedure. The next conversion starts immediately after the data is read.

The conversion sequence is terminated after all selected channels are converted. The end of sequence interrupt flag, the EOSIFLAG bit of ADC_IESR register, is set to '1' as soon as the last data result of a sequence is available. The EOSIFLAG bit is cleared by software by writing '1' to it.

Figure 80. Example of Sequential Conversion Mode

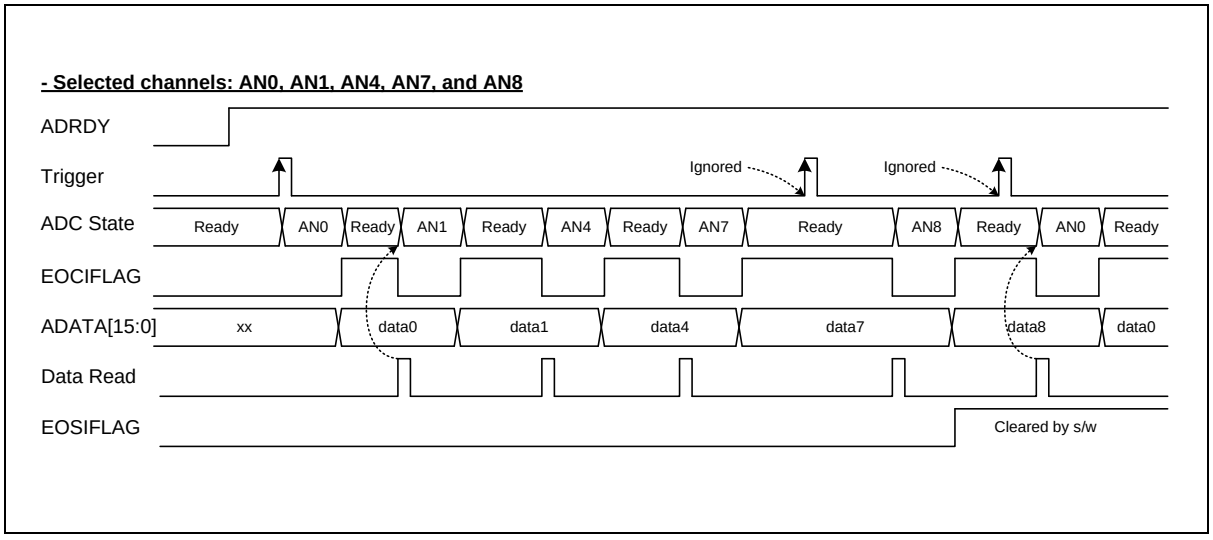


15.5.4.3 Continuous Conversion Mode

The ADC repeatedly converts all selected channels in order by a trigger signal during Continuous conversion mode. All trigger signals are ignored on the Continuous conversion mode. The next conversion starts immediately after data read as in the sequential conversion mode.

The end of sequence interrupt flag, the EOSIFLAG bit of ADC_IESR register, is also set to ‘1’ as soon as the last data result of a sequence is available, but the next conversion sequence is continued until a termination by software. The continuous conversion can be terminated by writing ‘0’ to the ADRDY bit of ADC_CR register.

Figure 81. Example of Continuous Conversion Mode



15.5.5 ADC Oversampling

The ADC has oversampling function by hardware for averaging, SNR improvement, and filtering. The function can handle multiple conversions and average them into a single data width, up to 16-bit. The oversampling ratio is configured by the OVSMPR[2:0] bits of ADC_OVSCR register with enabling the oversampling. The range is from x2 to x256. The ADC block has 20-bit accumulator for all sums of sampling data (256 x 12-bit: 20-bit). The average result consists of a right bit shift up to 8-bit. The right bit shift is selected by the OVSHT[3:0] bits of ADC_OVSCR register.

$$\text{Average Result} = \sum_{n=1}^{n=2^{OVSMPR[2:0]+1}} Data_n \text{ >> OVSHT[3:0]}$$

The upper bits of the average result are truncated with only the 16 least significant bits before being transferred into the ADC_DR register.

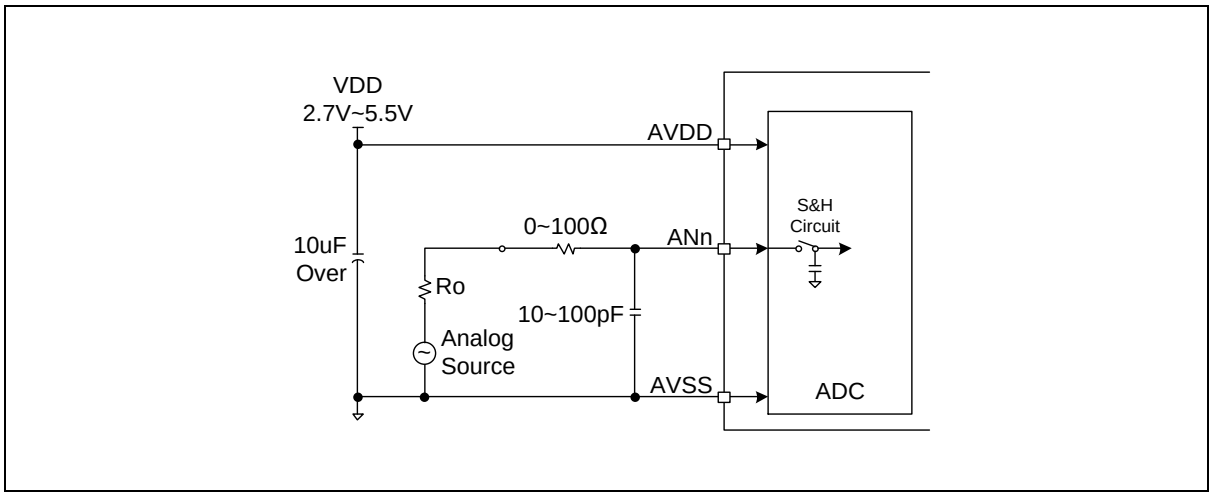
Table 77. ADC Result Data

OVSMPEN	OVSHT[3:0]	ADC_DR Register															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Don't care	0	0	0	0	ADATA[11:0]											
1	0	ACCUM[15:0] → ADATA[15:0]															
1	1	ACCUM[16:1] → ADATA[15:0]															
1	2	ACCUM[17:2] → ADATA[15:0]															
1	3	ACCUM[18:3] → ADATA[15:0]															
1	4	ACCUM[19:4] → ADATA[15:0]															
1	5	0	ACCUM[19:5] → ADATA[14:0]														
1	6	0	0	ACCUM[19:6] → ADATA[13:0]													
1	7	0	0	0	ACCUM[19:7] → ADATA[12:0]												
1	8	0	0	0	0	ACCUM[19:8] → ADATA[11:0]											

15.5.6 ADC Recommend Circuit

An output resistor (Ro) of analog source increases the capacitor charging time of the circuit. It may degrade the accuracy of ADC. The charging time depends on the resistor and capacitor of an input circuit. So, the sampling time should be adjusted appropriately by the ADC sampling time register (ADC_SAMR). The interval time of conversion should also be adjusted for accuracy.

Figure 82. Recommend Circuit for ADC Input



15.6 ADC Registers

The base address and register map of the ADC are described in the following:

Table 78. Base Address of ADC

Name	Base Address
ADC	0x4000_3000

Table 79. ADC Register Map

Name	Offset	Type	Description	Reset Value	Reference
ADC_CR	0x0000	RW	A/D Converter Control Register	0x0000_0000	15.6.1
ADC_OVSCR	0x0004	RW	A/D Converter Oversampling Control Register	0x0000_0000	15.6.2
ADC_IESR	0x0008	RW	A/D Converter Interrupt Enable and Status Register	0x0000_0000	15.6.3
ADC_DR	0x000C	RO	A/D Converter Data Register	Unknown	15.6.4
ADC_PREDR	0x0010	RW	A/D Converter Prescaler Data Register	0x0000_0000	15.6.5
ADC_SAMR	0x0014	RW	A/D Converter Sampling Time Register	0x0000_0000	15.6.6
ADC_CHSELR	0x0018	RW	A/D Converter Channel Selection Register	0x0000_0000	15.6.7

15.6.1 ADC_CR: A/D Converter Control Register

A/D Converter module should be configured properly before running.

ADC_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_CR=0x4000_3000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12 11 10	9 8	7	6	5 4 3	2	1 0
Reserved		ADCEN	Reserved	TRIG[2:0]	ETRGP[1:0]	ADRDY	Reserved	MDSEL[1:0]	Reserved	ADST
-		0	-	0	0	0	-	0	-	0
-		RW	-	RW	RW	RW	-	RW	-	RW

15	ADCEN	ADC Module Enable. (The ADC is automatically disabled at power down mode)	
		0	Disable ADC module operation.
		1	Enable ADC module operation.
12 10	TRIG[2:0]	ADC Trigger Signal Selection.	
		000	ADST.
		001	A/DC trigger signal from timer 10
		010	A/DC trigger signal from timer 11
		011	A/DC trigger signal from timer 12
		100	A/DC trigger signal from timer 13
		101	A/DC trigger signal from timer 14
		110	A/DC trigger signal from timer 15
		111	External ADC trigger input (ADTRG)
9 8	ETRGP[1:0]	ADC External Trigger Input Polarity Selection.	
		00	Disable ADC external trigger function
		01	Trigger on falling edge
		10	Trigger on rising edge
		11	Trigger on both falling and rising edge
7	ADRDY	ADC Conversion Ready.	
		0	Stop subsequent steps.
		1	Ready to convert.
4 3	MDSEL	ADC Conversion Mode Selection.	
		00	Single conversion mode.
		01	Sequential conversion mode
		10	Continuous conversion mode
		11	Reserved.
0	ADST	ADC Conversion software Start. This bit is automatically cleared to ‘0’ after operation.	
		0	No effect.
		1	Software Trigger signal generation for conversion start.

15.6.2 ADC_OVSCR: A/D Converter Oversampling Control Register

ADC_OVSCR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_OVSCR=0x4000_3004

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14 13 12 11 10 9 8	7 6 5	4	3 2 1 0
Reserved		OVSMPEN	Reserved	OVSMPR[2:0]	Reserved	OVSHT[3:0]
-		0	-	0	-	0
-		RW	-	RW	-	RW

15	OVSMPEN	Oversampling Enable.	
		0	Disable oversampling.
		1	Enable oversampling.
7 5	OVSMPR[2:0]	Oversampling Ratio Selection.	
		Oversampling ratio: $2^{OVSMPR[2:0]+1}$, Ex) On OVSMPR[2:0] = 010, 2^{2+1} = x8	
3 0	OVSHT[3:0]	Oversampling Data Shift.	
		0000	No shift
		0001	Shift right 1-bit
		0010	Shift right 2-bit
		0011	Shift right 3-bit
		0100	Shift right 4-bit
		0101	Shift right 5-bit
		0110	Shift right 6-bit
		0111	Shift right 7-bit
		1000	Shift right 8-bit
		Others	reserved

15.6.3 ADC_IESR: A/D Converter Interrupt Enable and Status Register

ADC_IESR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_IESR=0x4000_3008

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12	11	10	9	8	7 6 5 4	3	2	1	0
Reserved			STBIEN	OVRUNIEN	EOCIEN	EOSIEN	Reserved	STBIFLAG	OVRUNIFLAG	EOCIFLAG	EOSIFLAG
-			0	0	0	0	-	0	0	0	0
-			RW	RW	RW	RW	-	RW	RW	RW	RW

11	STBIEN	ADC Stabilization Interrupt Enable.	
		0	Disable stabilization interrupt.
		1	Enable stabilization interrupt.
10	OVRUNIEN	ADC Data Overrun Interrupt Enable.	
		0	Disable overrun interrupt.
		1	Enable overrun interrupt.
9	EOCIEN	ADC End of Conversion Interrupt Enable.	
		0	Disable end of conversion interrupt
		1	Enable end of conversion interrupt
8	EOSIEN	ADC End of Sequence Interrupt Enable.	
		0	Disable end of sequence interrupt.
		1	Enable end of sequence interrupt.
3	STBIFLAG	ADC Stabilization Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written.
		NOTE: 1. This bit will be set to ‘1’ after about 16/fADC time when the ADC module is enabled by ADCEN bit. So, the ADC conversion should start after reviewing whether this bit is ‘1’.	
2	OVRUNIFLAG	ADC Data Overrun Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written.
1	EOCIFLAG	ADC End of Conversion Interrupt Flag.	
		00	No request occurred.
		01	Request occurred. This bit is cleared to ‘0’ when ‘1’ is written or the result data are read by software or DMA.
0	EOSIFLAG	ADC End of Sequence Interrupt Flag.	
		0	No request occurred.
		1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.

15.6.4 ADC_DR: A/D Converter Data Register

ADC_DR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_DR=0x4000_300C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	LASTCH[4:0]	Reserved	ADATA[15:0]
-	Unknown	-	Unknown
-	RO	-	RO

28 24	LASTCH [4:0]	A/DC Last Conversion Channel Number The LASTCH[4:0] indicates the last converted channel number.
15 0	ADATA [15:0]	A/D Converter Result Data. On OVSMPEN = 0 - ADATA[15:12] = 0x0 and ADATA[11:0] = 12-bit data converted On OVSMPEN = 1 - OVSHT[3:0] = 0: ACCUM[15:0] → ADATA[15:0] - OVSHT[3:0] = 1: ACCUM[16:1] → ADATA[15:0] - OVSHT[3:0] = 2: ACCUM[17:2] → ADATA[15:0] - OVSHT[3:0] = 3: ACCUM[18:3] → ADATA[15:0] - OVSHT[3:0] = 4: ACCUM[19:4] → ADATA[15:0] - OVSHT[3:0] = 5: ACCUM[19:5] → ADATA[15] = 0x0 and ADATA[14:0] - OVSHT[3:0] = 6: ACCUM[19:6] → ADATA[15:14] = 0x0 and ADATA[13:0] - OVSHT[3:0] = 7: ACCUM[19:7] → ADATA[15:13] = 0x0 and ADATA[12:0] - OVSHT[3:0] = 8: ACCUM[19:8] → ADATA[15:12] = 0x0 and ADATA[11:0]

NOTE:

1. After waking up in power down mode, this data register is uncertain.

15.6.5 ADC_PREDR: A/D Converter Prescaler Data Register

ADC_PREDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_PREDR=0x4000_3010

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			PRED[6:0]
-			0
-			RW

6 0	PRED[6:0]	A/D Converter Prescaler Data bits. The prescaler sets the A/D conversion clock. The frequency of A/D converter should be less than or equal to 16 MHz. The range is 0x0 to 0x7F. fADC = PCLK/(PRED[6:0] +1) Max. fADC: 16 MHz If AVDD > 3.0V Max. fADC: 8 MHz If AVDD > 2.7V

15.6.6 ADC_SAMR: A/D Converter Sampling Time Register

ADC_SAMR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_SAMR=0x4000_3014

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			SAMCK[4:0]
-			0
-			RW

15	SAMCK[4:0]	Sampling Cycles for Sample/Hold Circuit. The range is 0x2 to 0x1F. Sampling cycles: SAMCK[4:0] + 2 during the values are 0x2 to 0x1E Sampling circuit is always on during the values are 0x1F Conversion cycles: Sampling cycles + 14 cycles.

15.6.7 ADC_CHSELR: A/D Converter Channel Selection Register

ADC_CHSELR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

ADC_CHSELR=0x4000_3018

31 30 29 28 27 26 25 24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	AN23(Res)	AN22(Res)	AN21(Res)	AN20(Res)	AN19(BGR)	AN18(AVDD)	Reserved	AN16(AVSS)	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
-	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	RW	RW	RW	RW	RW	RW	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

x	ANx	Channel for manufacturer use. x: 20 to 23 These bits must keep “0000”	
x	ANx	A/D Converter Channel Selection, x: 0 to 19	
		0	ANx is not selected for conversion
		1	ANx is selected for conversion

NOTE:

1. This register should not be written on going conversion.

15.6.8 ADC Register Map Summary

Table 80. ADC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADCEN	Res.	Res.	TRIG[2:0]			ETRGP[1:0]		ADRDY	Res.	Res.	MDSEL[1:0]		Res.	Res.	ADST
	Reset value																	0			0	0	0	0	0	0			0	0			0
0x04	ADC_OVSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVSMPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVSMPR[2:0]			Res.	OVSHT[3:0]			
	Reset value																	0								0	0	0		0	0	0	0
0x08	ADC_IESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.					STBIEN	OVRUNIEN	EOCIEN	EOSIEN	Res.	Res.	Res.	Res.	STBIFLAG	OVRUNIFLAG	EOCIFLAG	EOSIFLAG
	Reset value																					0	0	0	0					0	0	0	0
0x0C	ADC_DR	Res.	Res.	Res.	LASTCH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADATA[15:0]															
	Reset value				x	x	x	x	x									x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x10	ADC_PREDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRED[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x14	ADC_SAMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAMCK[4:0]				
	Reset value																												0	0	0	0	0
0x18	ADC_CHSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AN23(Res)	AN22(Res)	AN21(Res)	AN20(Res)	AN19(BGR)	AN18(AVDD)	Res.	AN16(AVSS)	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
	Reset value									0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16. Cyclic Redundancy Check (CRC) and Checksum

16.1 Introduction

A CRC (Cyclic Redundancy Check) generator is used to obtain 16-bit CRC code of Flash ROM and any data stream.

Among other applications, the CRC-based techniques are used to verify data transmission or storage integrity. In the scope of functional safety standards, they offer means of verifying Flash memory's integrity.

The CRC generator helps with computing the signature of the software during runtime, comparing with a reference signature.

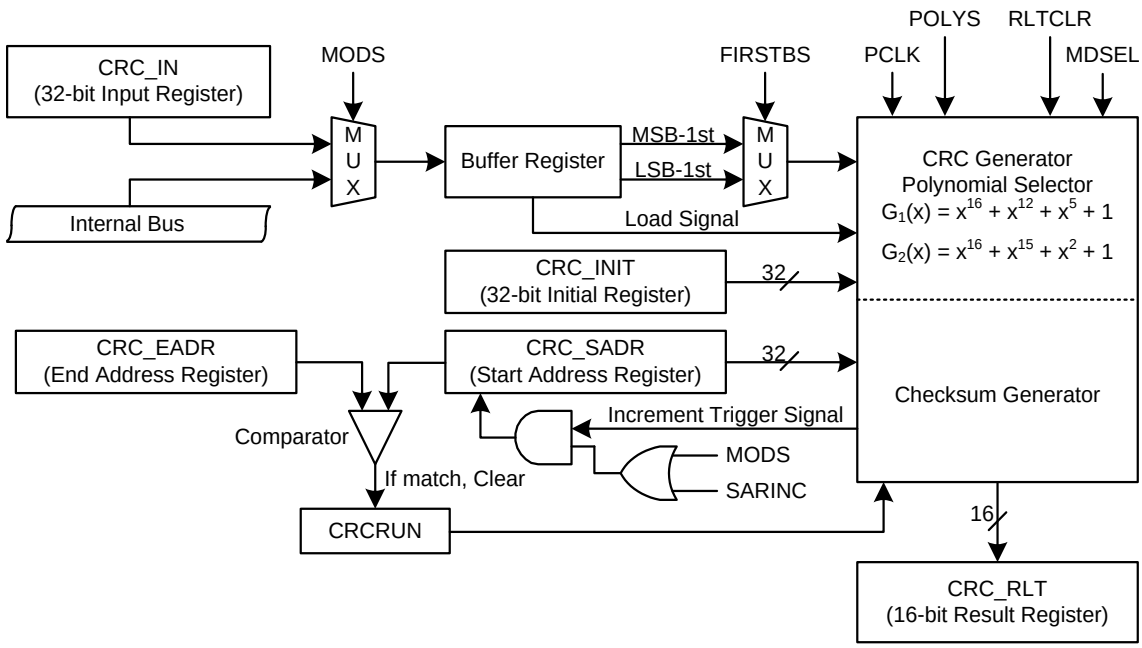
16.2 Main Features

The CRC generator of the A31G336 features the following:

- Auto CRC and user CRC mode
- CRC-CCITT (G1(x) = $x^{16} + x^{12} + x^5 + 1$) supported
- CRC-16 (G2(x) = $x^{16} + x^{15} + x^2 + 1$) supported
- CRC and checksum mode
- CRC/checksum start address auto increment (User mode only)

16.3 CRC and Checksum Block Diagram

Figure 83. CRC and Checksum Timer Block Diagram



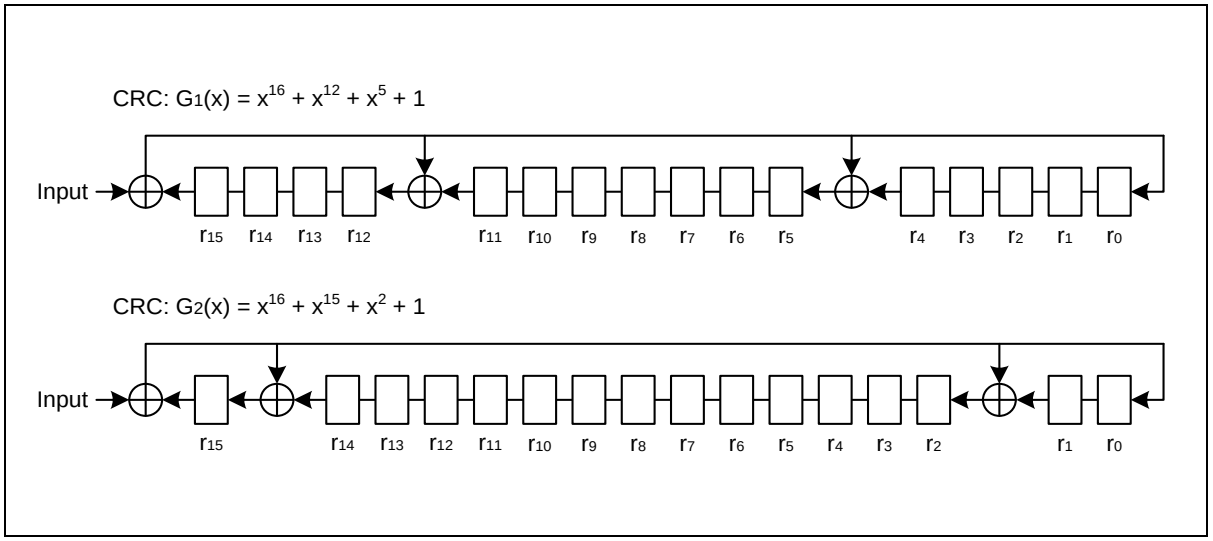
NOTES:

1. The operation is finished after calculation from CRC_SADR to the CRC_EADR on auto mode or SARINC=1.
2. The CRC_SADR and CRC_EADR are the same value after completion.
3. The end address (CRC_EADR) must be at least four greater than the start address (CRC_SADR).
4. The end address must be at least 0x7C from the start address on “Auto mode”.
5. The CRC_SADR/CRC_EADR on the “Auto mode” should have any value in Flash memory area (0x10000000 to 0x1003FFFF).
6. The CPU will be held on the “Auto Mode” if the CPU is in the Flash memory and the global interrupt should be disabled by software.
7. The HCLK frequency should be less than or equal to 16 MHz on the “Auto mode” by CRC_CR.MODS bit.
8. On auto mode run, the CRC_CR register can’t be written and the CRC_RLT register can’t be read.

16.4 CRC and Checksum Functional Description

16.4.1 CRC Polynomial Structure

Figure 84. CRC Polynomial Structure



16.4.2 The CRC Operation Procedure in Auto CRC/Checksum Mode

1. CRC/Checksum Clock Enable
2. Set CRC start address register. (CRC_SADR)
3. Set CRC end address register. (CRC_EADR)
4. Set CRC initial data register. (CRC_INIT)
5. Global interrupt Disable.
6. Select CRC(HCLK) Clock. (HCLK should be less than or equal to 16 MHz during CRC/Checksum auto mode)
7. Select Auto CRC/Checksum Mode and CRC.
8. CRC operation starts. (CRCRUN = 1)
9. Read the CRC result.
10. Global interrupt Enable.

16.4.3 The CRC Operation Procedure in User CRC/Checksum Mode

1. CRC/Checksum Clock Enable
2. Set CRC start address register. (CRC_SADR)
3. Set CRC end address register. (CRC_EADR)
4. Set CRC initial data register. (CRC_INIT)
5. Select User CRC/Checksum Mode and CRC
6. CRC operation starts. (CRCRUN = 1)
7. Input CRC Data at CRC_IN.
8. Check CRC is finished on Start Address Auto Increment or Compare Start address and End address to check CRC end point.
9. Repeat 8 and 9 until CRC end point.
10. CRC Stop and read CRC result.

16.5 CRC and Checksum Registers

The base address and register map of the CRC and checksum block are described in the following tables.

Table 81. Base Address of CRC and Checksum

Name	Base Address
CRC	0x3000_1000

Table 82. CRC and Checksum Register Map

Name	Offset	Type	Description	Reset Value	Reference
CRC_CR	0x0000	RW	CRC/Checksum Control Register	0x0000_0000	16.5.1
CRC_IN	0x0004	RW	CRC/Checksum Input Data Register	0x0000_0000	16.5.2
CRC_RLT	0x0008	RO	CRC/Checksum Result Data Register	0x0000_FFFF	16.5.3
CRC_INIT	0x000C	RW	CRC/Checksum Initial Data Register	0x0000_0000	16.5.4
CRC_SADR	0x0010	RW	CRC/Checksum Start Address Register	0x1000_0000	16.5.5
CRC_EADR	0x0014	RW	CRC/Checksum End Address Register	0x1003_FFFC	16.5.6

16.5.1 CRC_CR: CRC Control Register

The CRC_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_CR=0x3000_1000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			MODS	RLTCLR	MDSEL	POLYS	SARINC	Reserved	FIRSTBS	CRCRUN
-			0	0	0	0	0	-	0	0
-			RW	RW	RW	RW	RW	-	RW	RW

7	MODS	User/Auto Mode Selection.
	0	User mode. (Calculate every data written to the CRC_IN register)
	1	Auto mode. (Calculate till CRC_SADR == CRC_EADR)
6	RLTCLR	CRC/Checksum Result Data Register (CRC_RLT) Initialization.
	0	No effect.
	1	Initialize the CRC_RLT register with the value of CRC_INIT (This bit is automatically cleared to '0' after operation)
5	MDSEL	CRC/Checksum Selection.
	0	Select CRC.
	1	Select checksum.
4	POLYS	Polynomial Selection. (CRC only)
	0	Select CRC-CCITT ($G_1(x) = x^{16} + x^{12} + x^5 + 1$)
	1	Select CRC-16 ($G_2(x) = x^{16} + x^{15} + x^2 + 1$)
3	SARINC	CRC/Checksum Start Address Auto Increment Control. (User mode only)
	0	No effect.
	1	The CRC/Checksum start address register is incremented by '4' every data written to the CRC_IN register.
1	FIRSTBS	First Shifted-in Selection. (CRC only)
	0	MSB-1st.
	1	LSB-1st.
0	CRCRUN	CRC/Checksum Start Control and Busy.
	0	Not busy. The CRC operation can be finished by writing '0' to this bit while running.
	1	Start CRC operation. This bit is automatically cleared to '0' when the value of CRC_SADR register reaches the value of CRC_EADR register.

NOTE:

1. The 5 "NOP instruction" should be executed immediately after this bit becomes '1'.

NOTES:

1. The CRC_RLT register and the CRC/Checksum block should be initialized by writing '1' to the RLTCLR bit before a new CRC/Checksum calculation.
2. The CRCRUN bit should be set to '1' last time after setting appropriate values to the registers.
3. On the user mode, it will be calculated every writing data to the CRC_IN register during CRCRUN==1.
4. On the user mode with SARINC==0, the block is finished by writing '0' to the CRCRUN

bit.

5. It is prohibited to write any data to the CRC_IN register during CRCRUN==0.
6. The checksum is calculated by byte unit.
   Ex) On 0x34A7E991, CRC_RLT = 0x34 + 0xA7 + 0xE9 + 0x91.
7. The 5 “NOP Instruction” should follow immediately after CRCRUN bit is set to ‘1’.

16.5.2 CRC_IN: CRC Input Data Register

The CRC_IN register is 32-bit size.

CRC_IN=0x3000_1004

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
INDATA[31:0]			
0x00000000			
RW			

31 0	INDATA[31:0]	CRC Input Data.

NOTE:

1. The CRC_IN register should be written in 1-word (32-bit).

16.5.3 CRC_RLT: CRC Result Data Register

The CRC_RLT register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_RLT=0x3000_1008

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		RLTDATA[15:0]	
-		0xFFFF	
-		RO	

15 0	RLTDATA[15:0]	CRC Result Data.

16.5.4 CRC_INIT: CRC Initial Data Register

The CRC_INIT register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_INIT=0x3000_100C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		INIDATA[15:0]	
-		0x0000	
-		RW	

15 0	INIDATA[15:0]	CRC Initial Data.

16.5.5 CRC_SADR: CRC Start Address Register

The CRC_SADR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_SADR=0x3000_1010

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
SADR[31:2]				Don't care
0x10000000				00
RW				-

31 2	SADR[31:2]	CRC Start Address.
1 0	-	Don't care.

NOTE:

1. The LSB 7-bit of the start address should be “0x00” on “Auto mode”

16.5.6 CRC_EADR: CRC End Address Register

The CRC_EADR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

CRC_EADR=0x3000_1014

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
EADR[31:2]				Don't care
0x1003FFFC				00
RW				-

31 2	EADR[31:2]	CRC End Address.
1 0	-	Don't care.

NOTE:

1. The LSB-7bits of the end address should be "0x7C" on "Auto mode".

16.5.7 CRC Register Map Summary

Table 83. CRC Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODS	RLTCLR	MDSEL	POLYS	SARINC	Res.	FIRSTBS	CRCRUN
	Reset value																									0	0	0	0	0		0	0
0x04	CRC_IN	INDATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	CRC_RLT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLTDATA[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	CRC_INIT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INIDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	CRC_SADR	SADR[31:2]																														Don't care	
	Reset value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	CRC_EADR	EADR[31:2]																														Don't care	
	Reset value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

17. Comparator 0/1

17.1 Introduction

The A31G336 series includes two comparator modules. Each comparator module has two registers such as a control register (CMP_CR), and a status register (CMP_SR). The comparator module has an internal reference circuit too.

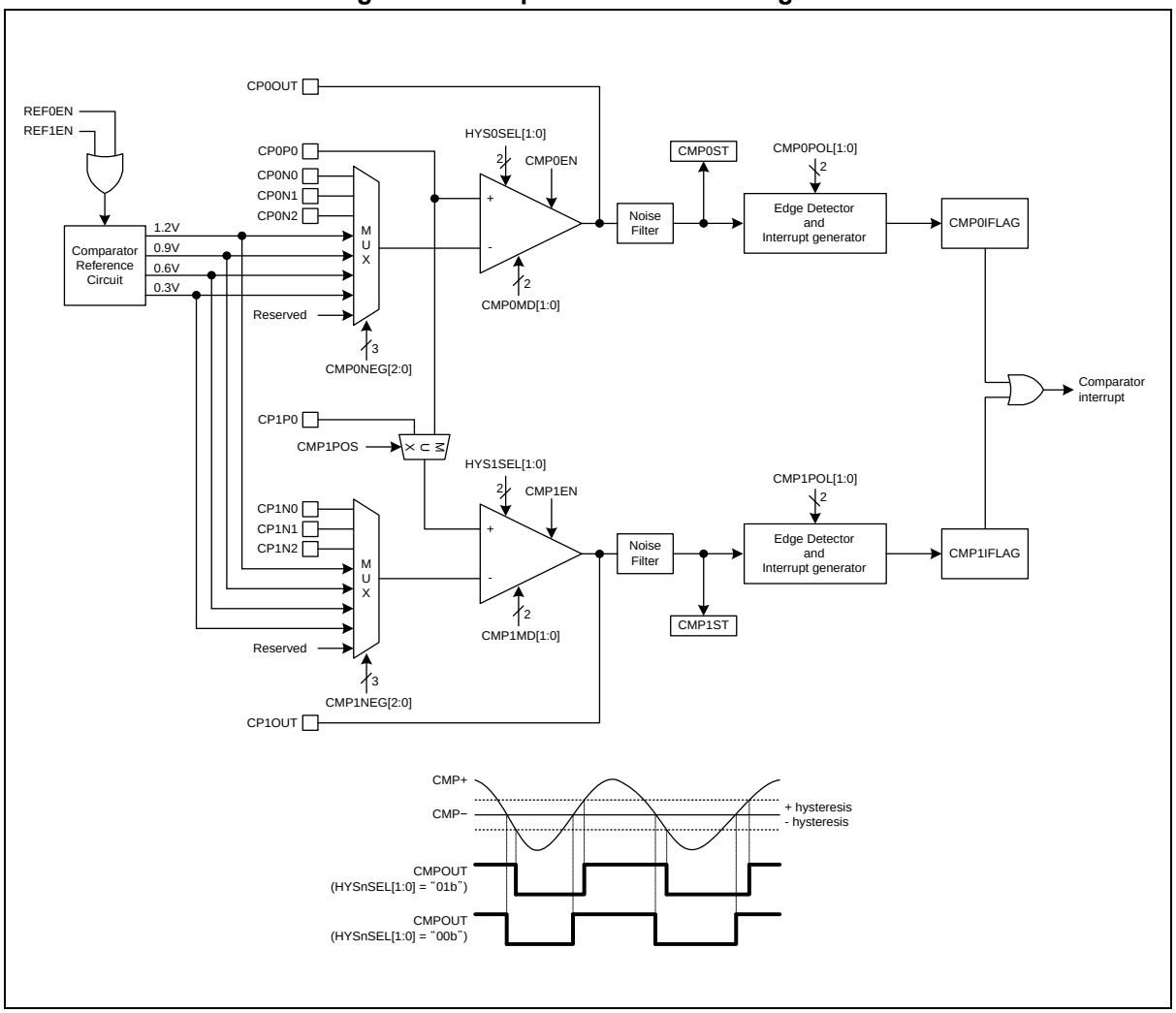
17.2 Main Features

The Comparator of the A31G336 features the following:

- External analog inputs
- Hysteresis function
- Low and fast speed selectable
- Wake-up possible from DEEP SLEEP mode

17.3 Comparator 0/1 Block Diagram

Figure 85. Comparator 0/1 Block Diagram



17.4 Pin description for Comparator 0/1

Table 84. Pins and External Signals for Comparator 0/1

Pin Name	Type	Description
CP0P0	A	Comparator 0 positive input
CP0N0	A	Comparator 0 negative input
CP0N1	A	Comparator 0 negative input
CP0N2	A	Comparator 0 negative input
CP0OUT	A	Comparator 0 output
CP1P0	A	Comparator 1 positive input
CP1N0	A	Comparator 1 negative input
CP1N1	A	Comparator 1 negative input
CP1N2	A	Comparator 1 negative input
CP1OUT	A	Comparator 1 output

17.5 Comparator 0/1 Registers

The base address and register map of the Comparator 0/1 are described in the following tables.

Table 85. Base Address of Comparator 0/1

Name	Base Address
CMP0	0x4000_5600
CMP1	0x4000_5680

Table 86. Comparator n Register Map (n = 0 and 1)

Name	Offset	Type	Description	Reset Value	Reference
CMPn_CR	0x0000	RW	Comparator n Control Register	0x0000_0000	17.5.1
CMPn_SR	0x0004	RW	Comparator n Status Register	0x0000_0000	17.5.2

17.5.1 CMPn_CR: Comparator n Control Register

The CMPn_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access. (n = 0 and 1)

CMP1_CR=0x4000_5600, CMP1_CR=0x4000_5680

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	REFnEN / Reserved / HYSnSEL[1:0] / Reserved	CMPnEN / CMPnNEG[2:0] / Reserved / CMPnPOS / Reserved	CMPnMD[1:0] / CMPnPOL[1:0] / Reserved / NFCKn[2:0]
-	0 - 0 -	0 0 - 0 -	0 0 - 0
-	RW - RW -	RW RW - RW -	RW RW - RW

25	REFnEN	Comparator n Internal Reference Enable.
	0	Disable internal reference
	1	Enable internal reference

NOTE:

1. The comparator reference turns on when at least one of REF0EN and REF1EN is 1.

19 18	HYSnSEL[1:0]	Comparator n Hysteresis Selection.
	00	Disable hysteresis
	01	Enable low hysteresis
	10	Enable medium hysteresis
	11	Enable high hysteresis
15	CMPnEN	Comparator n Enable (The comparator n isn't automatically disabled at power down mode).
	0	Disable comparator n operation
	1	Enable comparator n operation
14 12	CMPnNEG[2:0]	Comparator n Negative Input Selection.
	000	Select external CPnN0 pin
	001	Select external CPnN1 pin
	010	Select external CPnN2 pin
	011	Select internal reference 1.2V
	100	Select internal reference 0.9V
	101	Select internal reference 0.6V
	110	Select internal reference 0.3V
	111	Reserved
9	CMPnPOS	Comparator n Positive Input Selection
	0	Select external CPnP0 pin
	1	Select CP0P0 signal on comparator 1 (Reserved on comparator 0)
7 6	CMPnMD[1:0]	Comparator n Mode Selection
	00	Ultra low-power mode
	01	Low-power mode
	10	Medium power mode
	11	High-speed mode
5 4	CMPnPOL[1:0]	Comparator n Interrupt Polarity Selection.
	00	No interrupt at any edge

		01	Interrupt on falling edge
		10	Interrupt on rising edge
		11	Interrupt on both of falling and rising edge
20	NFCKn[2:0]	Comparator n Noise Filter Sampling Clock Selection.	
		000	PCLK/1
		001	PCLK/2
		010	PCLK/4
		011	PCLK/8
		100	PCLK/16
		101	PCLK/32
		110	PCLK/64
		111	Reserved

NOTES:

1. If a level is not detected three or more times in a row at the sampling clock, the signal is eliminated as noise.
2. A pulse level should be input for the duration of 3 clocks or more to be actually detected as a valid edge.
3. The comparator noise filter is automatically disabled at deep sleep mode and recovered after deep sleep mode release.

17.5.2 CMPn_SR: Comparator n Status Register

The CMPn_SR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

(n = 0 and 1)

CMP0_SR=0x4000_5604, CMP1_SR=0x4000_5684

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5	4	3 2 1	0
Reserved				CMPnIFLAG	Reserved	CMPnST
-				0	-	0
-				RW	-	RO

4	CMPnIFLAG	Comparator n Interrupt Flag	
		0	No request occurred
		1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
0	CMPnST	Comparator n Output Status	
		0	Comparator n output is low
		1	Comparator n output is high

17.5.3 CMP Register Map Summary

Table 87. CMP Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CMPn_CR	Res.	Res.	Res.	Res.	Res.	Res.	REFnEN	Res.	Res.	Res.	Res.	Res.	HYSnSEL[1:0]		Res.	Res.	CMPnEN	CMPnNEG[2:0]			Res.	Res.	CMPnPOS	Res.	CMPnMD[1:0]		CMPnPOL[1:0]		Res.	NFCKn[2:0]		
	Reset value							0						0	0			0	0	0	0			0		0	0	0	0		0	0	0
0x04	CMPn_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMPnIFLAG	Res.	Res.	Res.	CMPnST
	Reset value																												0				0

18. Temperature Sensor (TS)

18.1 Introduction

The Temperature Sensor (TS) is a ring-oscillator type and can be used to measure the junction temperature of the device. The nominal frequency at 30°C is about 0.9 MHz and it varies from 0.6 MHz to 1.25 MHz as the temperature changes from -40°C to +85°C.

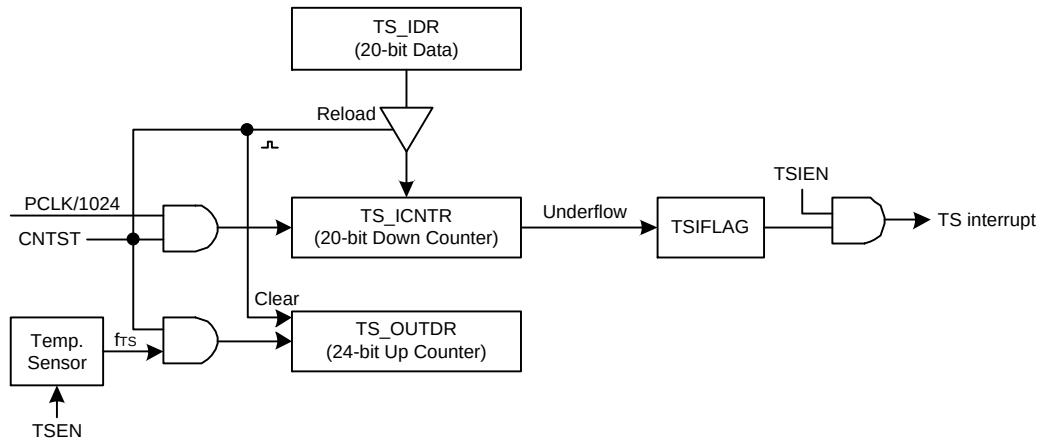
18.2 Main Features

The TS of the A31G336 features the following:

- -40°C to +85°C wide range of operating temperature
- A down counter at 20-bit intervals to count the frequency of the TS
- A 24-bit data register to store the count value of the temperature sensor frequency

18.3 TS Block Diagram

Figure 86. TS Block Diagram



NOTE:

1. The CNTST bit will be cleared to '0' after the underflow of TS_ICNTR.

18.4 TS Functional Description

18.4.1 Ring-Oscillator of Temperature Sensor

The ring-oscillator of the TS needs a maximum stabilization time of about 500 μsec. Its frequency ranges from about 0.6 MHz to 1.25 MHz depending on temperature. The frequency variation is about 3.2 kHz per Celsius degree and its frequency is directly proportional to temperature.

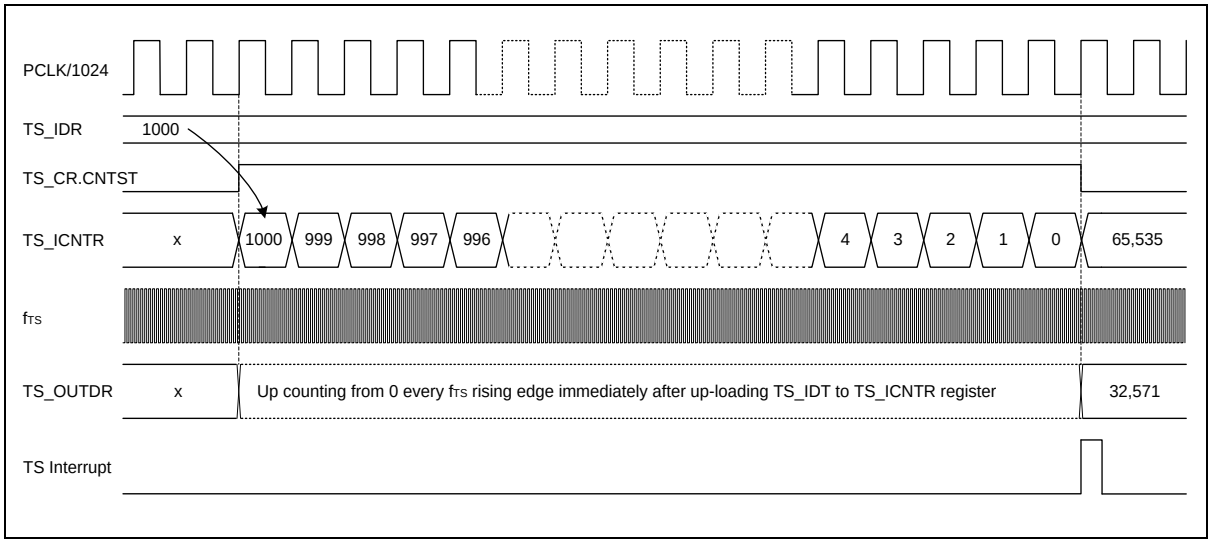
18.4.2 Frequency Counting

The frequency of the TS can be measured indirectly by using the system clock. The TS block provides 20-bit interval data and 24-bit output data registers to count the number of temperature sensor clocks during a specific period.

The 16-bit interval data register, TS_IDR, sets the measurement period.

Figure 87 shows temperature sensor timing diagram.

Figure 87. Temperature Sensor Timing Diagram



18.4.3 Temperature Calculation

The Configure Option Page 0 provides the frequency values of the TS measured by the manufacturer at 30°C and 85°C, respectively.

Table 88 shows the corresponding registers in the Configure Option Page 0.

Table 88. TS Output Frequency Registers in Configure Option Page 0

Name	Address	Type	Description
TS_FREQ_T30	0x1FFF2048	RO	Temperature Sensor Output Frequency acquired at 30°C [Hz]
TS_FREQ_T85	0x1FFF204C	RO	Temperature Sensor Output Frequency acquired at 85°C [Hz] (Commercial grade)

The temperature can be calculated using the formula below:

$$Temperature = \frac{F(T) - F(30)}{\Delta F} + 30\ [℃]$$

Where

$$\Delta F = \frac{F(T2) - F(T1)}{T2 - T1}$$

T1 = 30°C, T2 = 85°C (Commercial grade)

F(T1) [kHz] is the temperature sensor output frequency acquired at 30°C.

F(T2) [kHz] is the temperature sensor output frequency acquired at 85°C (commercial grade).

F(T) [kHz] is the temperature sensor output frequency acquired at an arbitrary temperature.

Example)

- TS_FREQ_T30 of Configure Option Page 0 = 840,000 (840 kHz)
- TS_FREQ_T85 of Configure Option Page 0= 1,030,000 (1,030 kHz)
- If the acquired frequency at arbitrary temperature is about 660,000 (660 kHz)
- ΔF = (1,030 – 840) / (85 – 30) = 190/55 = 3.4 [kHz]
- Measured temperature = (660.000 – 840)/3.4 + 30 = -52.9 +30 ≈ -22.9°C
- Therefore, the result is about -22.9°C.

18.5 TS Registers

The base address and register map of the temperature sensor are described in the following tables.

Table 89. Base Address of TS

Name	Base Address
TS	0x4000_5F80

Table 90. TS Register Map

Name	Offset	Type	Description	Reset Value	Reference
TS_CR	0x0000	RW	Temperature Sensor Control Register	0x0000_0000	18.5.1
TS_IDR	0x0004	RW	Temperature Sensor Interval Data Register	0x000F_FFFF	18.5.2
TS_ICNTR	0x0008	RO	Temperature Sensor Interval Counter Register	0x000F_FFFF	18.5.3
TS_OUTDR	0x000C	RO	Temperature Sensor Output Data Register	0x0000_0000	18.5.4

18.5.1 TS_CR: Temperature Sensor Control Register

The TS_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TS_CR=0x4000_5F80

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			TSIEN	Reserved		TSIFLAG	Reserved		CNTST	TSEN
-			0	-		0	-		0	0
-			RW	-		RW	-		RW	RW

7	TSIEN	Temperature Sensor Interrupt Enable	
		0	Disable interrupt
		1	Enable interrupt
4	TSIFLAG	Temperature Sensor Interrupt flag	
		0	No request occurred
		1	Request occurred. This bit is cleared to ‘0’ when writing ‘1’.
1	CNTST	Counting Start	
		0	Stop counting
		1	Counting start after “reload data to TS_ICNTR and clear TS_OUTDR”
		NOTE: 1. This bit will be cleared to ‘0’ after the underflow of TS_ICNTR register.	
0	TSEN	Temperature Sensor Enable	
		0	Disable temperature sensor
		1	Enable temperature sensor

18.5.2 TS_IDR: Temperature Sensor Interval Data Register

The TS_IDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TS_IDR=0x4000_5F84

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		IDATA[19:0]	
-		0xFFFFF	
-		RW	

19 0	IDATA[19:0]	Temperature Sensor Interval Data

18.5.3 TS_ICNTR: Temperature Sensor Interval Counter Register

The TS_ICNTR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TS_ICNTR=0x4000_5F88

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		ICNT[19:0]	
-		0xFFFFF	
-		RO	

19 0	ICNT[19:0]	Temperature Sensor Interval Counter

18.5.4 TS_OUTDR: Temperature Sensor Control Register

The TS_OUTDR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

TS_OUTDR=0x4000_5F8C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved	OUTDATA[23:0]		
-	0		
-	RO		

23 0	OUTDATA	Temperature Sensor Output Data bits

18.5.5 TS Register Map Summary

Table 91. TS Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TS_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSIEN	Res.	Res.	TSIFLAG	Res.	Res.	CNTST	TSEN
	Reset value																									0			0			0	0
0x04	TS_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDATA[19:0]																			
	Reset value													1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	TS_ICNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ICNT[19:0]																			
	Reset value													1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	TS_OUTDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTDATA[23:0]																							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19. High-Speed Internal OSC Clock Management Unit (HSICMU)

19.1 Introduction

The High-Speed Internal OSC Clock Management Unit (HSICMU) integrated into the A31G336 is designed for the auto-trimming of the Internal OSC clock used for Xtal-less USB.

The following are characteristics of the High-Speed Internal OSC Clock Management Unit (HSICMU) built in the A31G336. It is used to auto-trimming of Internal OSC clock for Xtal-less USB.

The HSICMU is ideally suited to provide a precise clock for the USB peripheral. In such a case, the synchronization signal can be derived from the start-of-frame (SOF) packet signalization on the USB bus, which is sent by a USB host at 1 ms intervals.

The synchronization signal can also be derived from the LSE oscillator output, or it can be generated by user software.

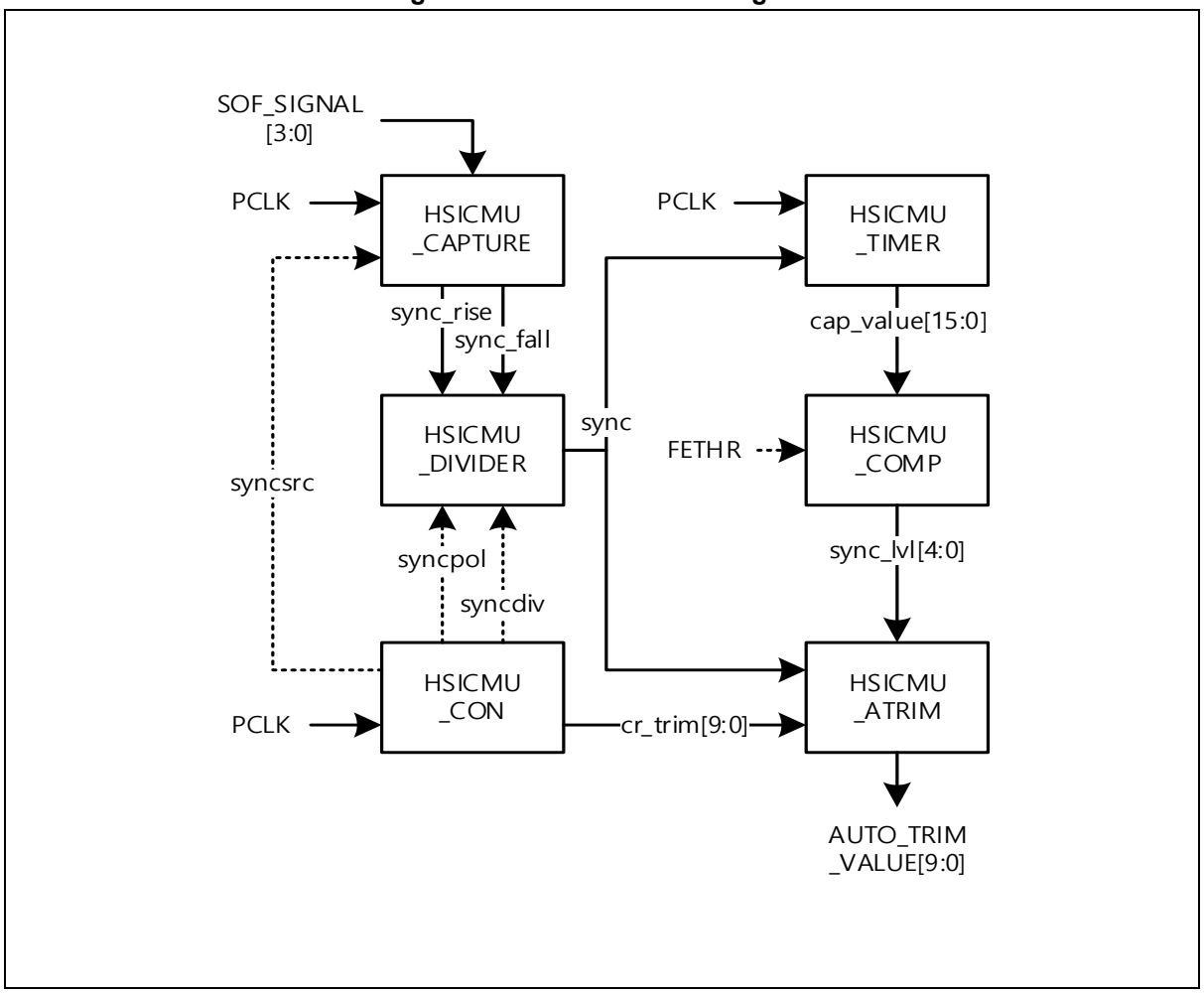
19.2 Main Features

The HSICMU generator of the A31G336 features the following:

- Selectable synchronization source with programmable prescaler and polarity:
  - USB SOF (Start of Frame) sent received toggle
  - LSE
  - External pin (PD5)
- Possibility to generate synchronization pulses by software
- Automatic oscillator trimming capability with no need for CPU action
- 16-bit frequency error counter with automatic error value capture and reload
- Programmable limit for automatic frequency error value evaluation and status reporting

19.3 HSICMU Block Diagram

Figure 88. HSICMU Block Diagram



19.4 HSICMU Functional Description

19.4.1 Synchronization Input

The HSICMU synchronization (SYNC) source, selectable through the HSICMU_CFG register, can be the signal from the LSE clock or the USB SOF signal.

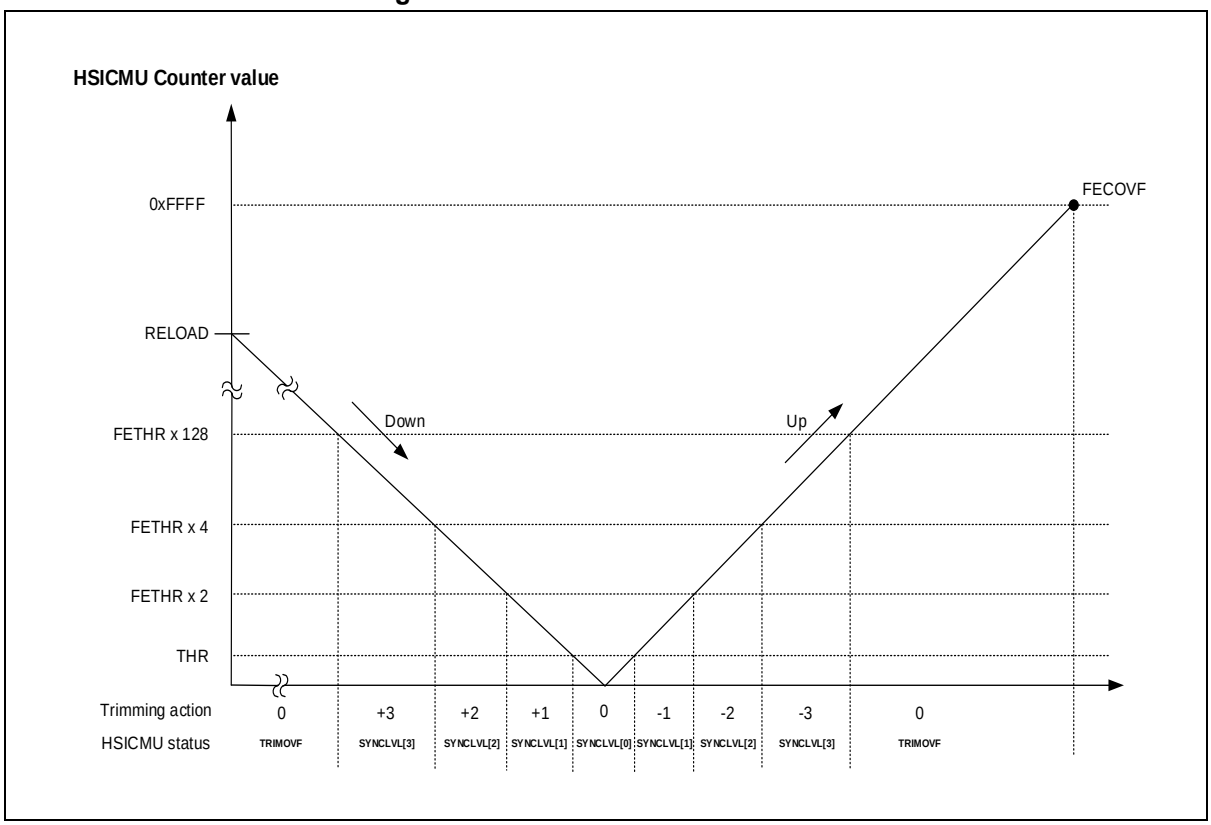
For more information on the HSICMU synchronization source configuration, refer to 19.5.2.

19.4.2 Frequency Error Measurement

The frequency error counter is a 16-bit down/up counter which is reloaded with the RELOAD value on each SYNC event. It starts counting down till it reaches zero value. Then it starts counting to the 0xFFFF where it eventually stops (if no SYNC event is received) and generates a FECOVF event.

When the SYNC event is detected, the actual value of the frequency error counter and its counting direction are stored in the CAPTURE (frequency error capture) field and in the FECDIR (frequency error direction) bit of the HSICMU_ISR register. When the SYNC event is detected during the down counting phase (before reaching the zero value), it means that the actual frequency is lower than the target (and so, that the TRIM value must be incremented), while when it is detected during the up counting phase it means that the actual frequency is higher (and that the TRIM value must be decremented).

Figure 89. HSICMU Counter Behavior



19.4.3 Frequency Error Evaluation and Automatic Trimming

The measured frequency error is evaluated by comparing its value with a set of limits:

- SYNCLVL[0] limit, given directly in the FETHR field of the HSICMU_THR register
- SYNCLVL[1] limit, defined as 2 × FETHR value
- SYNCLVL[2] limit, defined as 4 × FETHR value
- SYNCLVL[3] limit, defined as 128 × FETHR value

The result of this comparison is used to generate the status indication and to control the automatic trimming which is enabled by setting the AUTOTRIM_EN bit in the HSICMU_CR register:

- When the frequency error is below the SYNCLVL[0] limit, it means that the actual trimming value in the TRIM field is the optimal one, hence no trimming action is needed.
  - SYNCLVL[0] status indicated (TRIM value not changed in AUTOTRIM mode)
- When the frequency error is above or equal to the SYNCLVL[0] limit but below the SYNCLVL[1] limit, it means that some trimming action is necessary but that adjustment by one trimming step is enough to reach the optimal TRIM value.
  - SYNCLVL[1] status indicated (TRIM value adjusted by one trimming step in AUTOTRIM mode)
- When the frequency error is above or equal to the SYNCLVL[1] limit but below the SYNCLVL[2] limit, it means that a stronger trimming action is necessary, and there is a risk that the optimal TRIM value is not reached for the next period.
  - SYNCLVL[2] status indicated (TRIM value adjusted by two trimming steps in AUTOTRIM mode)
- When the frequency error is above or equal to the SYNCLVL[2] limit but below the SYNCLVL[3] limit, it means that a stronger trimming action is necessary, and there is a risk that the optimal TRIM value is not reached for the next period.
  - SYNCLVL[3] status indicated (TRIM value adjusted by three trimming steps in AUTOTRIM mode)
- When the frequency error is above or equal to the SYNCLVL[3] limit, it means that the frequency is out of the trimming range. This can also happen when the SYNC input is not clean or when some SYNC pulse is missing (for example when one USB SOF is corrupted).
  - SYNCLVL[4] status indicated (TRIM value not changed in AUTOTRIM mode)

NOTE:

1. If the actual value of the ATRIM field is so close to its limits that the automatic trimming would force it to overflow or underflow, then the TRIM value is set just to the limit and the TRIMOVF status is indicated
   In AUTOTRIM mode (AUTOTRIM_EN bit set in the HSICMU_CR register), the ATRIM field of HSICMU_ATRIM is adjusted by hardware and is read-only.

19.4.4 HSICMU Initialization and Configuration

The RELOAD value must be selected according to the ratio between the target frequency and the frequency of the synchronization source after prescaling. It is then decreased by one to reach the expected synchronization on the zero value. The formula is the following:

$$\mathrm{RELOAD} = \left(\frac{f_{\mathrm{TARGET}}}{f_{\mathrm{SYNC}}}\right) - 1$$

The reset value of the RELOAD field corresponds to a target frequency of 32 MHz and a synchronization signal frequency of 1 kHz (SOF signal from USB).

The selection of the FETHR value is closely coupled with the HSI oscillator characteristics and its typical trimming step size. The optimal value corresponds to half of the trimming step size, expressed as several HSI oscillator clock ticks. The following formula can be used:

$$\mathrm{FETHR} = \left(\frac{f_{\mathrm{TARGET}}}{f_{\mathrm{SYNC}}}\right) \times \mathrm{HSI\ TRIM\ STEP}[\%]/100\%/\ 2$$

The result must always be rounded up to the nearest integer value to obtain the best trimming response. If frequent trimming actions are not needed in the application, the hysteresis can be increased by slightly increasing the FETHR value.

The reset value of the FELIM field corresponds to (f_{TARGET} / f_{SYNC}) = 32,000 and to a typical trimming step size of 0.2%.

There is no hardware protection from a wrong configuration of the RELOAD and FETHR fields which can lead to an erratic trimming response. The expected operational mode requires proper setup of the RELOAD value (according to the synchronization source frequency), which is also greater than 128 × FETHR value (SYNCLVL[3] limit).

19.5 HSICMU Registers

The base address and register map of the HSICMU block are described in the following tables.

Table 92. Base Address of HSICMU

Name	Base Address
HSICMU	0x4000_6000

Table 93. HSICMU Register Map

Name	Offset	Type	Description	Reset Value	Reference
HSICMU_CR	0x0000	RW	HSICMU Control Register	0x0000_1000	19.5.1
HSICMU_CFG	0x0004	RW	HSICMU Configuration Register	0x0000_FFFF	19.5.2
HSICMU_ISR	0x0008	RO	HSICMU ISR Register	0x0000_0000	19.5.3
HSICMU_ICR	0x000C	WO	HSICMU ICR Register	0x0000_0000	19.5.4
HSICMU_TRIM	0x0010	RW	HSICMU TRIM Register	0x0000_0000	19.5.5
HSICMU_ATRIM	0x0014	RO	HSICMU ATRIM Register	0x0000_0000	19.5.6
HSICMU_CAP	0x0018	RO	HSICMU Capture Register	0x0000_0000	19.5.7
HSICMU_THR	0x001C	RW	HSICMU Frequency Error Threshold Register	0x0000_0000	19.5.8

19.5.1 HSICMU_CR: HSICMU Control Register

The HSICMU_CR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_CR=0x4000_6000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14	13	12	11	10	9	8	7 6 5 4 3 2 1 0
Reserved			SET_TRIMVALU	SYNC_START	SET_TIMER	SW_SYNC	AUTOTRIM_EN	FERRCNT_EN	Reserved
-			0	1	0	0	0	0	-
-			WO	RW	WO	WO	RW	RW	-

13	SET_TRIMVALUE	Trim value is set to HSICMU_TRIM in HSICMU_ATRIM block.	
		0	Disable
		1	Enable
12	SYNC_START	Start synchronized timer by SOF signals	
		0	Disable
		1	Enable
11	SET_TIMER	The value in the reload register is written to the HSICMU_TIMER counter	
		0	Disable
		1	Enable
10	SW_SYNC	Generate Software SYNC event	
		0	Disable
		1	Enable
9	AUTOTRIM_EN	Auto trim enable	
		0	Disable
		1	Enable
8	HSICMU_EN	HSICMU Peripheral block Enable	
		0	Disable
		1	Enable

19.5.2 HSICMU_CFG: HSICMU Configure Register

The HSICMU_CFG register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_CFG=0x4000_6004

31 30	29 28	27	26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
SYNCPOL[1:0]	SYNCSRC[1:0]	AC_EN	SYNCDIV[2:0]	Reserved	RELOAD[15:0]	
0	0	0	0	-	0xFFFF	
RW	RW	RW	RW	.	RW	

31 30	SESYNCPOL[1:0]	Sync Signal Polarity Selection.	
		00	Both edge
		01	Rising edge
		10	Falling edge
		11	Reserved
29 28	SYNCSRC[1:0]	Sync Signal Source Selection	
		00	Capture Data Update when SOF is successfully received
		01	Capture Data Update when SOF is successfully received
		10	LSE clock
		11	CMUSYNC signal of PD5
27	AC_EN	Auto Clear HSICMU_ISR.SYNC_FLAG when read HSICMU_ISR.	
		0	Disable
		1	Enable
26 24	SYNCDIV[2:0]	Sync divider	
		000	Sync not divided (default)
		001	Sync divided by 2
		010	Sync divided by 4
		011	Sync divided by 8
		100	Sync divided by 16
		101	Sync divided by 32
		110	Sync divided by 64
		111	Sync divided by 128
15 0	RELOAD[15:0]	HSICMU_TIMER counter is set to RELOAD Value Set expected timer value. (Sync period / PCLK period)	

19.5.3 HSICMU_ISR: HSICMU Interrupt and Status Register

The HSICMU_ISR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_ISR=0x4000_6008

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11 10 9 8	7 6 5	4	3	2	1	0
Reserved		FECDIR	SYNC_FLAG	TRIMOVF	FECOVF	Reserved		SYNCLVL[4]	SYNCLVL[3]	SYNCLVL[2]	SYNCLVL[1]	SYNCLVL[0]
-		0	0	0	0	-		0	0	0	0	0
-		RO	RO	RO	RO	-		RO	RO	RO	RO	RO

15	FECDIR	Frequency Error Counter Direction.
		0 Counter in decreasing direction
		1 Counter in increasing direction
14	SYNC_FLAG	Sync Signal Flag
		0 None
		1 Updated HSICMU_CAP register with input sync signal
13	TRIMOVF	Trim Value Overflow
		0 None
		1 The trim value is out of limits when AUTOTRIMEN = 1
12	FECOVF	Frequency Error Overflow
		0 None
		1 The Counter value changed in an increasing direction and the counter value increased to the end (0xFFFF)
4 0	SYNCLVL[4:0]	Sync Error Rate level
		SYNCLVL[0]: HSICMU_CAP ≤ FETHR(Pass)
		SYNCLVL[1]: FETHR < HSICMU_CAP < FETHR × 2
		SYNCLVL[2]: FETHR × 2 < HSICMU_CAP ≤ FETHR × 4
		SYNCLVL[3]: FETHR × 4 < HSICMU_CAP ≤ FETHR × 128
		SYNCLVL[4]: FETHR × 128 < HSICMU_CAP

19.5.4 HSICMU_ICR: HSICMU Interrupt and Status Clear Register

The HSICMU_ICR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_ICR=0x4000_600C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12	11 10 9 8	7 6 5	4	3	2	1	0
Reserved			SYNC_FLAG	TRIMOVF	FECOVF	Reserved		SYNCLVL[4]	SYNCLVL[3]	SYNCLVL[2]	SYNCLVL[1]	SYNCLVL[0]
-			0	0	0	-		0	0	0	0	0
-			WO	WO	WO	-		WO	WO	WO	WO	WO

14	SYNC_FLAG	Sync Signal Flag clear
		0 None
		1 Clear when set to ‘1’
13	TRIMOVF	Trim value overflow flag clear
		0 None
		1 Clear when set to ‘1’
12	FECOVF	Frequency Error Overflow Flag Clear bit after HSICMU_ CR.HSICMU_EN set to ‘0’.
		0 None
		1 Clear when set to ‘1’
4 0	SYNCLVL[4:0]	Sync Error Rate Level Flag Clear bits after HSICMU_CR.SYNC_START set to 0.
		SYNCLVL[0]: Clear when set to ‘1’
		SYNCLVL[1]: Clear when set to ‘1’
		SYNCLVL[2]: Clear when set to ‘1’
		SYNCLVL[3]: Clear when set to ‘1’
		SYNCLVL[4]: Clear when set to ‘1’

19.5.5 HSICMU_TRIM: HSICMU Control TRIM Register

The HSICMU_TRIM register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_TRIM=0x4000_6010

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			TRIM[10:0]
-			0x000
-			RW

10 0	TRIM[10:0]	Trim value register for trimming

19.5.6 HSICMU_ATRIM: HSICMU Control Auto-TRIM Register

The HSICMU_ATRIM register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_ATRIM=0x4000_6014

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			ATRIM[10:0]
-			0x000
-			RO

10 0	ATRIM[10:0]	Current trim value during auto trimming

19.5.7 HSICMU_CAP: HSICMU Capture Value Register

The HSICMU_CAP register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_CAP=0x4000_6018

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		CAPTURE[15:0]	
-		0x0000	
-		RO	

15 0	CAPTURE[15:0]	Capture value of the frequency error rate counter

19.5.8 HSICMU_THR: HSICMU Frequency Error Rate Threshold Register

The HSICMU_THR register is 32-bit size. This register is available at 32-/ 16-/ 8-bit access.

HSICMU_TH=0x4000_601C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved		FETHER [15:0]	
-		0x0000	
-		RW	

15 0	FETHR[15:0]	Frequency error rate threshold

19.5.9 HSICMU Register Map Summary

Table 94. HSICMU Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	HSICMU_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SET_TRIMVALUE	SYNC_START	SET_TIMER	SW_SYNC	AUTOTRIM_EN	FERRCNT_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																			0	1	0	0	0	0								
0x04	HSICMU_CFG	SYNCPOL[1:0]		SYNCSRC[1:0]		AC_EN	SYNCDIV[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RELOAD[15:0]															
	Reset value	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	HSICMU_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FECDIR	SYNC_FLAG	TRIMOVF	FECOVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCLVL[4]	SYNCLVL[3]	SYNCLVL[2]	SYNCLVL[1]	SYNCLVL[0]
	Reset value																	0	0	0	0								0	0	0	0	0
0x0C	HSICMU_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNC_FLAG	TRIMOVF	FECOVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCLVL[4]	SYNCLVL[3]	SYNCLVL[2]	SYNCLVL[1]	SYNCLVL[0]
	Reset value																		0	0	0								0	0	0	0	0
0x10	HSICMU_TRIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x14	HSICMU_ATRIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATRIM[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x18	HSICMU_CAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAPTURE[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	HSICMU_THR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FETHR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20. USB Full Speed Device Interface

20.1 Introduction

The USB block of the A31G336 series controls an interface between a full-speed USB bus and the AHB bus.

There are 5 endpoints in this controller. Each of the endpoint can be configured as IN or OUT endpoint. It is USB 2.0 full-speed device specification and supports control/bulk/interrupt transfer types. USB suspend/resume are supported, which allows the device clocks to stop for low-power consumption. Please refer to *Universal Serial Bus Specification Revision 1.1* for details.

20.2 Main Features

The USB generator of the A31G336 features the following:

- USB 2.0 Full-Speed specification
- Provides 1 interrupt vector with 3 different interrupt events (WAKEUP, USB and BUS)
- Supports Control/Bulk/Interrupt transfer type.
- Supports suspend function when no bus activity existing for 3 ms
- Provides remote wake-up capability.
- Support Only Full-Speed (FS, 12-Mbps)
- Configurable number of endpoints from 1 to 5
- USB DMA two channels, 64 bytes buffer size.

20.3 USB Implementation

Table 95 describes the USB implementation in the devices.

Table 95. A31G336 USB Implementation

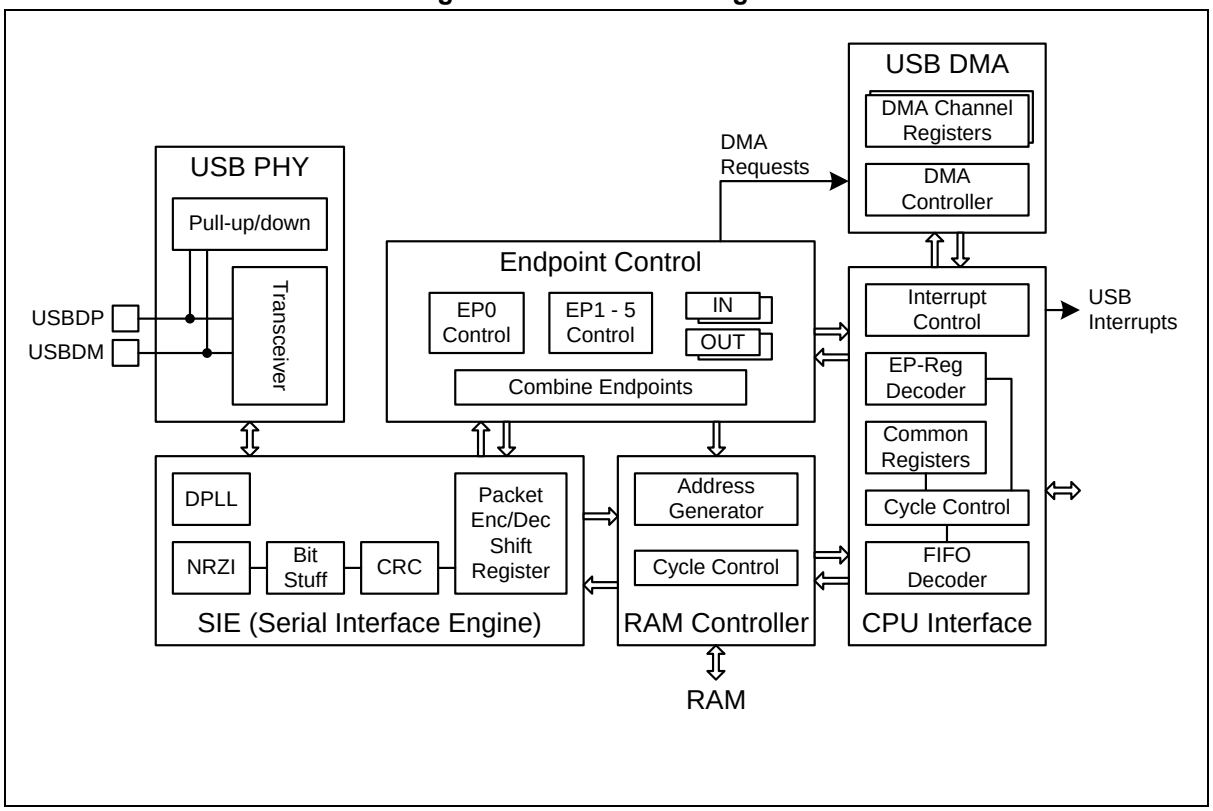
USB Features	USB Implementation
Number of endpoints excluding endpoints 0	5
Size of dedicated packet buffer memory SRAM	1024 bytes
Dedicated packet buffer memory SRAM access scheme	4 x 8 bits / word
Battery Charging Detection (BCD) support	X
Embedded pull-up/down resistor on USB_DP/DN line	O

NOTE:

1. The maximum size of the FIFO is fixed at 64 / 128 bytes or less.

20.4 USB Block Diagram

Figure 90. USB Block Diagram



The USB peripheral provides a USB-compliant connection between the host computer and the function implemented by the microcontroller. Data transfer between the host computer and the system memory occurs through a dedicated packet buffer memory accessed directly by the USB peripheral. This dedicated memory size is up to 128 bytes, and up to 10 mono-directional or 5 bidirectional endpoints can be used.

The USB peripheral interfaces with the USB host, detecting token packets, handling data transmission/reception, and processing handshake packets as required by the USB standard. Transaction formatting is performed by the hardware, including CRC generation and checking.

20.5 USB Functional Description

20.5.1 USB Pins and Signals

The external pins assigned for USB are described in Table 96.

Table 96. Pins Assignment of USB: External Pins

Pin Name	Type	Description	Supported Packages				
			A31G336 (64-LQFP)	A31G336 (48-LQFP)	A31G336 (48-QFN)	A31G336 (40-QFN)	A31G336 (32-QFN)
USBDP	I/O	USB data line plus	O	O	O	O	O
USBDM	I/O	USB data line minus	O	O	O	O	O

20.5.2 USB Description of USB Block

The USB peripheral implements all the features related to USB interfacing, which include the following blocks:

- USB Physical Interface (USB PHY): This block maintains the electrical interface to an external USB host. It contains the differential analog transceiver itself, controllable embedded pull-up / pull-down resistor (connected to USB_DP / USB_DM line), multiplexed on same USB_DP and USB_DM lines.
- Serial Interface Engine (SIE): The SIE handles NRZI encoding/decoding, bit stuffing/unstuffing, and CRC generation/checking. It generates a 12 MHz USB clock from the 48 MHz input clock to the core and synchronizes this clock to the incoming data stream when receiving data from the USB. It generates headers for packets to be transmitted and decodes the headers of received packets. This unit also generates signals according to USB peripheral events, such as Start of Frame (SOF), USB_Reset, Data errors etc. and to Endpoint related events like end of transmission or correct reception of a packet; these signals are then used to generate interrupts.
- Endpoint Controllers: Two controller state machines are used: one for control transfers over Endpoint 0, and one for Bulk, Interrupt transactions over Endpoints 1 to 5. Each endpoint has an associated register containing the endpoint type and its current status. For mono-directional/single-buffer endpoints, a single register can be used to implement two distinct endpoints. The number of endpoints is 5, allowing up to 10 mono-directional/single-buffer in any combination. Also, it contains information about the status of the whole USB peripheral and used to force some USB events, such as resume and power-down.
- SRAM Controllers: The SRAM controller provides an interface to a single block of synchronous single-port RAM which is used to buffer packets between the CPU and USB. It takes the FIFO pointers from the endpoint controllers, converts them to address pointers within the SRAM block and generates the SRAM access control signals.
- CPU Interface: The CPU Interface allows access to the control/status registers and the FIFOs for each endpoint. It also generates an interrupt to the CPU when a packet has been successfully transmitted or received, or when the core enters Suspend mode or resumes from Suspend mode.
- USB DMA: The USB DMA provides two channels to allow FIFOs for IN Endpoints 1 to 5 and OUT Endpoints 1 to 5 to be accessed by dedicated DMA controller. The DMA request lines are individually enabled by a bit in the appropriate INCSR2 control register or OUTCSR2 control register. These DMA Modes are primarily designed to be used where large blocks of data are transferred to a Bulk endpoint. The USB protocol requires such packets to be split into a series of packets of maximum packet size = 8, 16, 32 or 64 bytes. The last packet in the series will be less than the maximum packet size (or a null packet if the total size of the transfer is an exact multiple of the maximum packet size) and the receiver may use this ‘short’ packet to signal the end of the transfer. DMA Mode 1 can be used to avoid the overhead of having to interrupt the processor after each individual packet – instead just interrupting the processor after the transfer has completed. If the endpoint is configured for isochronous transfers, DMA Mode 0 should always be selected. However, the use of isochronous transfers is greatly limited because the buffer size in each FIFO is small. (FIFO buffers are supported up to 128 bytes and do not meet to 1023 bytes.)

20.6 Programming Considerations

In the following sections describes the main tasks required of the application software to obtain USB compliant behavior, to ease application software development.

20.6.1 Select Clock Source

For USB communication, the 48 MHz clock output from the PLL is used. The clock source for the PLL can use either the HSE or the HSI. If the HSE is used as the PLL source clock, the PLL output clock can be used for USB peripheral without calibration. However, if HSI is used, the HSI output clock must be calibrated to 32 MHz using the HSICMU before using it as the PLL source clock.

The HSICMU is ideally suited to provide a precise clock for the USB peripheral. In such case, the synchronization signal can be derived from the start-of-frame (SOF) packet signalization on the USB bus, which is sent by a USB host at 1 ms intervals. The synchronization signal can also be derived from the LSE oscillator output or from an external pin.

20.6.2 Endpoint Initialization

The first step to initialize an endpoint is to write appropriate values to the registers so that the USB peripheral finds the data to be transmitted already available and the data to be received can be buffered.

Typically, unidirectional endpoints need to initialize only bits and registers related to the supported direction. Once the transmission and/or reception are enabled, setting register should not be modified by the application software, as the hardware can change their value on the fly. When the data transfer operation is completed, notified by an interrupt event (if enabled), they can be accessed again to re-enable a new operation.

20.6.3 Endpoint Control

There are 5 endpoints in this controller. Each of the endpoints can be configured as Control, Bulk, Interrupt or Isochronous transfer type. All the operations including Control, Bulk, Interrupt and Isochronous transfer are implemented in this block. It is also used to manage data sequential synchronization, endpoint state control, current endpoint start address, current transaction status, and data buffer status in each endpoint.

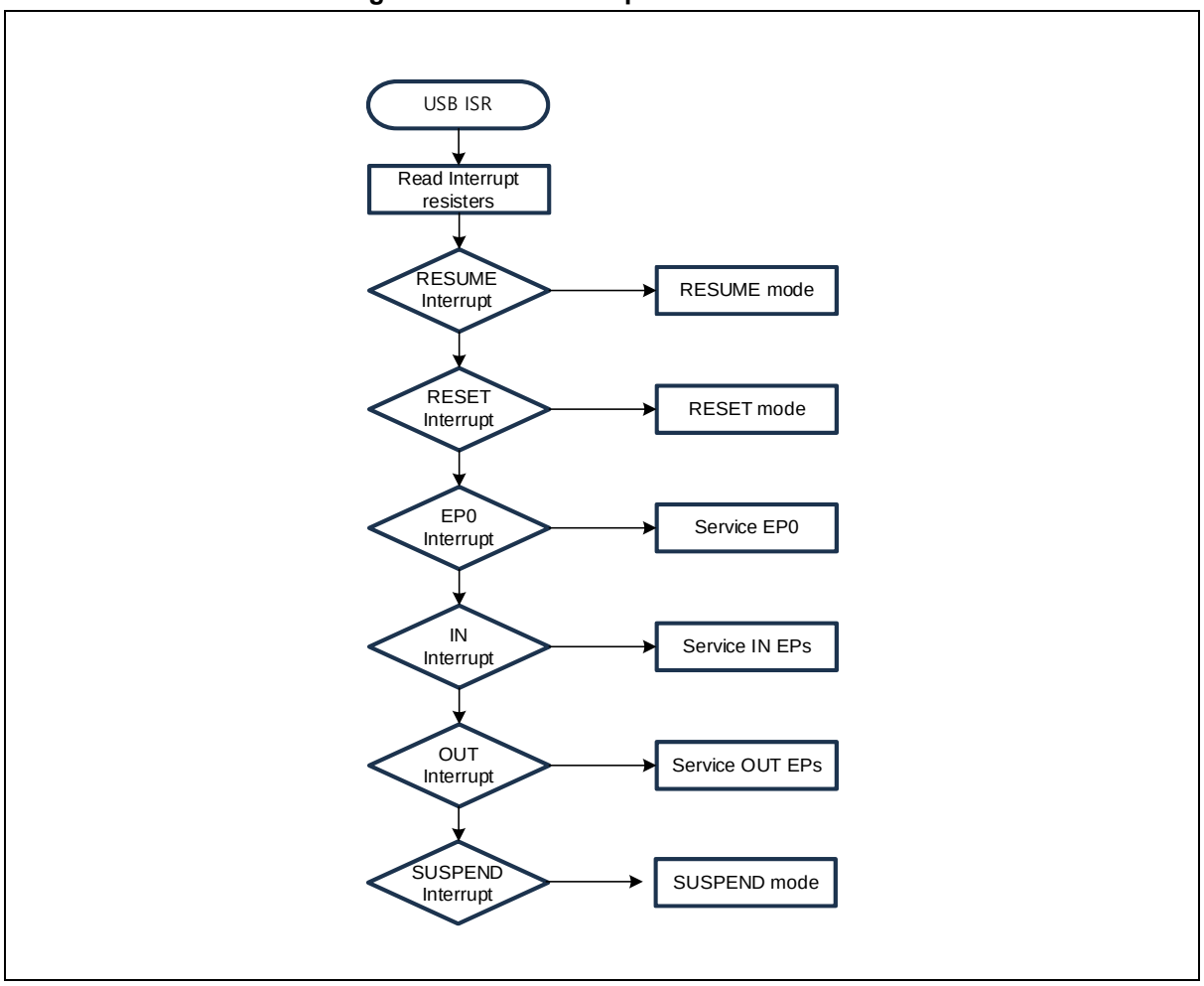
20.6.4 USB Interrupt Handling

This USB provides 1 interrupt vector with USB and BUS interrupt events. The USB event notifies users of some USB requests, such as IN ACK, OUT ACK., and the BUS event notifies users of some bus events, such as suspend and, resume. The related bits must be set in the interrupt enable register of USB Device Controller to enable USB interrupts.

USB interrupt is used to notify users of any USB event on the bus, and users can read INTRIN1 and INTROUT1 to take necessary responses.

Same as USB interrupt, BUS interrupt notifies users of some bus events, like SOF, USB reset, suspend, and resume. A user can read INTRUSB to acknowledge bus events.

Figure 91. USB Interrupt Service Routine



20.6.5 Power Saving

If set to support Suspend mode, the USB automatically turns off the PHY transceiver to save power while the chip enters Suspend mode. To further reduce current consumption in the application, sleep / deep sleep mode can be used, and the clock source must be changed to either LSE or LSI to ensure that the USB Peripheral operates to a minimum. When an application wakes up in sleep mode or deep sleep mode by an interrupt, the clock source must be restored to 48 MHz PLL output to resume USB communication.

20.6.6 Buffer Control

The USB Specification defines an endpoint as a source or sink of data. Since USB is a serial bus, a device endpoint is a FIFO which sequentially empties or fills with USB data bytes. A31G336 USB can uniquely address 5 endpoints, IN1 through IN5 and OUT1 through OUT5.

There is 960 bytes of SRAM in the controller and the 5 endpoints share this buffer. The seven 64-byte buffers are designated EP0, EP1IN/OUT, EP2IN/OUT and EP3IN/OUT and the four 128-byte buffers are designated EP4IN/OUT and EP5IN/OUT.

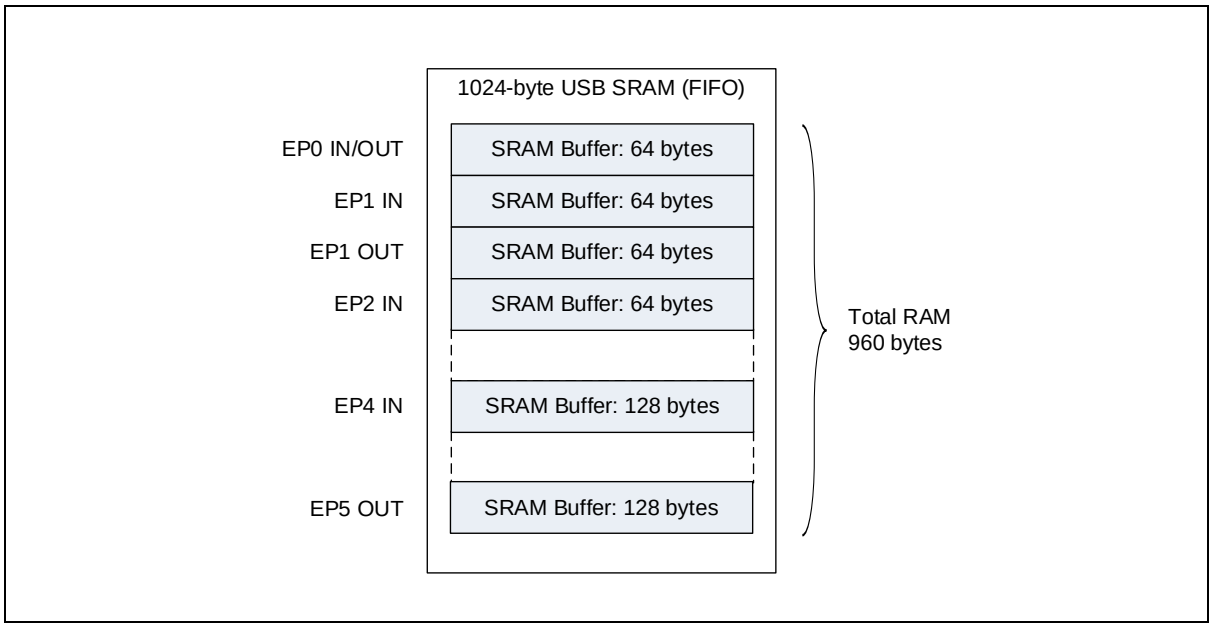
EP0 is the default CONTROL endpoint, a bidirectional endpoint that uses a single 64-byte buffer for both IN and OUT data. Firmware reads or fills the EP0 buffer when the data stage of a CONTROL transfer is required.[NOTE1]

Other EPn[NOTE2] IN/OUT use separate 64-byte or 128-byte buffers. Firmware can configure these endpoints as BULK or INTERRUPT. Also, EP4 and EP5 have 128-byte buffers, they can be used as double buffers to improve speed when using BULK transfer. But the buffer size is predetermined and cannot be changed by the user to suit bandwidth requirements. FIFO is divided into read and write, so the written value cannot be read again. To check the written value, users need to send data to USB.

NOTES:

1. The eight SETUP bytes in a CONTROL transfer do not appear in the 64-byte EP0 endpoint buffer. Instead, to simplify programming, the A31G336 USB automatically stores eight SETUP bytes in a separate buffer.
2. n = 1 to 5

Figure 92. USB Buffer Control

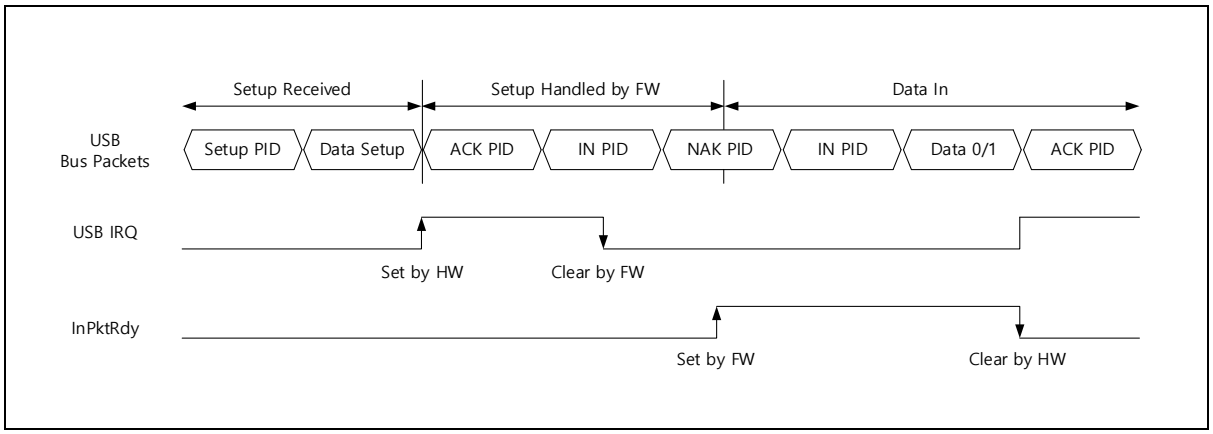


20.6.7 Handling Transactions with USB Device Peripheral

Users can use interrupt to monitor the USB transactions. When transactions occur, USB_IRQHandler will be set by hardware and send an interrupt request to CPU (if related interrupt enabled). The following is the control flow with interrupt enabled.

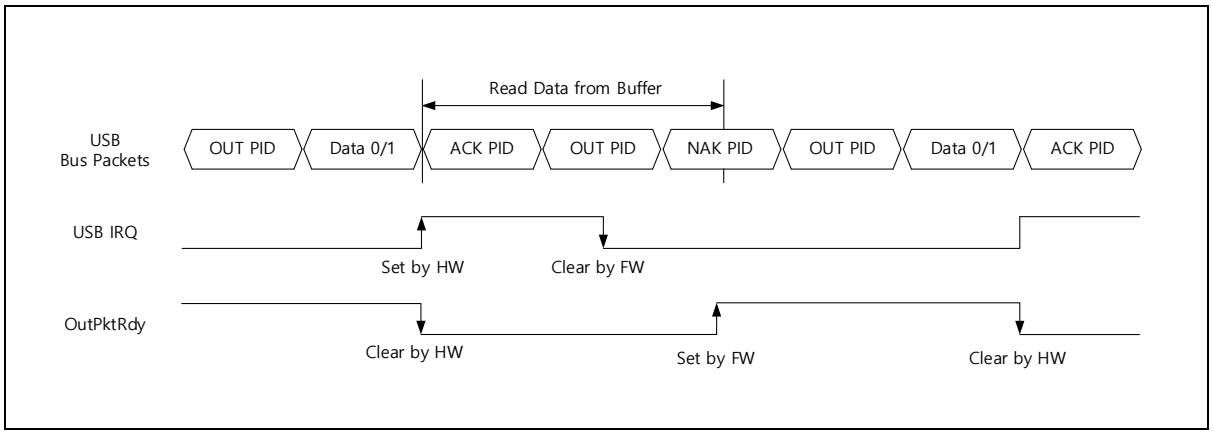
When USB host has requested data from a device controller, users need to prepare related data in the specified endpoint buffer in advance. After buffering the required data, users need to write the actual data length in the specified INMAXP register. Once this register is written, the internal signal “INPKTRDY” will be asserted, and the buffering data will be transmitted immediately after receiving associated IN token from Host. Note that after transferring the specified data, the signal “INPKTRDY” will be de-asserted automatically by hardware.

Figure 93. Setup Transaction Followed by Data IN Transaction



Alternatively, when USB host wants to transmit data to the OUT endpoint in the device controller, hardware will buffer these data to the specified endpoint buffer. After this transaction is completed, hardware will record the data length in specified OUTMAXP register and de-assert the internal signal “OUTPKTRDY”. This will avoid hardware accepting the next transaction until user moves out the current data in the related endpoint buffer. Once users have processed this transaction, the specified OUTMAXP register needs to be written by firmware to assert the signal “OUTPKTRDY” again to accept the next transaction.

Figure 94. Data Out Transfer



20.7 USB Registers

The following tables describe the base address and register map for the USB.

Table 97. Base Address of USB

Name	Base Address
USB	0x3000_2000

Table 98. USB Register Map

Name	Offset	Type	Description	Reset Value	Reference.
USB_FADDR	0x0000	RW	Function Address register	0x0000_0000	20.7.1
USB_POWER	0x0004	RW	Power management register	0x0000_0000	20.7.2
USB_INTRIN1	0x0008	RO	Interrupt Flag Register for Endpoint 0 plus IN Endpoints 1 to 5	0x0000_0000	20.7.3
USB_INTROUT1	0x0010	RO	Interrupt Flag Register for OUT Endpoints 1 to 5	0x0000_0000	20.7.4
USB_INTRUSB	0x0018	RO	Interrupt Flag Register for Common USB Interrupts	0x0000_0000	20.7.5
USB_INTRIN1E	0x001C	RW	Interrupt Enable Register for Endpoint 0 plus IN Endpoints 1 to 5	0x0000_003F	20.7.6
USB_INTROUT1E	0x0024	RW	Interrupt Enable Register for OUT Endpoints 1 to 5	0x0000_003E	20.7.7
USB_INTRUSBE	0x002C	RW	Interrupt Enable Register for Common USB Interrupts	0x0000_0006	20.7.8
USB_FRAME1	0x0030	RO	Frame number bit 0 to 7	0x0000_0000	20.7.9
USB_FRAME2	0x0034	RO	Frame number bit 8 to 10	0x0000_0000	20.7.10
USB_INDEX	0x0038	RW	Index Register for Selecting the Endpoint Status and Control Registers	0x0000_0000	20.7.11
USB_PUPDCR	0x0040	RW	DP/DN Pins Pull-up/Down Resistor Control Register	0x0000_0000	20.7.12
USB_PHYCR	0x0044	RW	PHY IP SUSPEND mode control register	0x0000_0001	20.7.13
USB_INMAXP	0x0050	RW	Maximum Packet Size for IN Endpoint. (Index Register set to select Endpoints 1 – 5 only)	0x0000_0000	20.7.14
USB_CSR0	0x0054	RW	Control Status Register for Endpoint 0. (Index Register set to select Endpoint 0)	0x0000_0000	20.7.15

Table 98. USB Register Map (continued)

Name	Offset	Type	Description	Reset Value	Reference.
USB_INCSR1	0x0054	RW	Control Status Register 1 for IN Endpoint. (Index Register set to select Endpoints 1 – 5)	0x0000_0000	20.7.16
USB_INCSR2	0x0058	RW	Control Status Register 2 for IN Endpoint. (Index Register set to select Endpoints 1 – 5 only)	0x0000_0000	20.7.17
USB_OUTMAXP	0x005C	RW	Maximum Packet Size for OUT Endpoint. (Index Register set to select Endpoints 1 – 5 only)	0x0000_0000	20.7.18
USB_OUTCSR1	0x0060	RW	Control Status Register 1 for OUT Endpoint. (Index Register set to select Endpoints 1 – 5 only)	0x0000_0000	20.7.19
USB_OUTCSR2	0x0064	RW	Control Status Register 2 for OUT Endpoint. (Index Register set to select Endpoints 1 – 5 only)	0x0000_0000	20.7.20
USB_COUNT0	0x0068	RO	Number of Received Bytes in Endpoint 0 FIFO. (Index Register set to select Endpoint 0)	0x0000_0000	20.7.21
USB_OUTCOUNT1	0x0068	RO	Number of Bytes in OUT Endpoint FIFO (Lower Byte). (Index register set to select Endpoints 1 – 5 only)	0x0000_0000	20.7.22
USB_OUTCOUNT2	0x006C	RO	Number of Bytes in OUT Endpoint FIFO (Upper Byte). (Index Register set to select Endpoints 1 – 5 only)	0x0000_0000	20.7.23
USB_FIFO0	0x0080 to 0x008C	RW	FIFO for EP0	0x0000_0000	20.7.24
USB_FIFO1	0x0090 to 0x009C	RW	FIFO for EP1	0x0000_0000	20.7.24
USB_FIFO2	0x00A0 to 0x00AC	RW	FIFO for EP2	0x0000_0000	20.7.24
USB_FIFO3	0x00B0 to 0x00BC	RW	FIFO for EP3	0x0000_0000	20.7.24
USB_FIFO4	0x00C0 to 0x00CC	RW	FIFO for EP4	0x0000_0000	20.7.24
USB_FIFO5	0x00D0 to 0x00DC	RW	FIFO for EP5	0x0000_0000	20.7.24
USB_DMAINTR	0x0200	RO	USB DMA Interrupt Pending Register	0x0000_0000	20.7.25
USB_DMA1CNTL	0x0204	RW	USB DMA Channel 1 Control Register	0x0000_0000	20.7.26
USB_DMA1ADDR	0x0208	RW	USB DMA Channel 1 Memory Address Register	0x0000_0000	20.7.27
USB_DMA1COUNT	0x020C	RW	USB DMA Channel 1 Transfer Byte Count Register	0x0000_0000	20.7.28
USB_DMA2CNTL	0x0214	RW	USB DMA Channel 2 Control Register	0x0000_0000	20.7.29
USB_DMA2ADDR	0x0218	RW	USB DMA Channel 2 Memory Address Register	0x0000_0000	20.7.30
USB_DMA2COUNT	0x021C	RW	USB DMA Channel 2 Transfer Byte Count Register	0x0000_0000	20.7.31

20.7.1 USB_FADDR: Function Address Register

FADR should be written with the function's 7-bit address. It is then used for decoding the function address in subsequent token packets.

USB_FADDR=0x3000_2000

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4 3 2 1 0
Reserved			UPDATE	FADDR[6:0]
-			0	0
-			RO	RW

7	UPDATE	Set when FADR is written. Cleared when the new address takes effect (at the end of the current transfer).	
		0	Clear by USB when end of the current transfer.
		1	Updated Function address. (FADDR)
6 0	FADDR[6:0]	The function address	

20.7.2 USB_POWER: Power Management Register

PWR is used for controlling Suspend and Resume signaling.

USB_POWER=0x3000_2004

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4	3	2	1	0
Reserved			ISOUPDATE	Reserved	RESET	RESUME	SUSPENDMD	SUSPENDEN
-			0	-	0	0	0	0
-			RW	-	RO	RW	RO	RW

7	ISOUPDATE	When set by the CPU, this Device will wait for an SOF token from the time INPKTRDY is set before sending the packet. If an IN token is received before an SOF token, then a zero-length data packet will be sent. This bit is only used by endpoints performing Isochronous transfers.	
		0	Disables.
		1	Wait SOF token in isochronous (ISO) mode
3	RESET	set while reset signaling is present on the bus	
		0	Cleared by USB
		1	Detect Reset status on the bus
2	RESUME	Set by the CPU to generate resume signaling when the function is in suspend mode.	
		0	clear this bit by CPU after 10 ms (a maximum of 15 ms) to end Resume signaling.
		1	Generate Resume signal in suspend mode
1	SUSPENDMD	Set by the USB when suspend mode is entered.	
		0	Cleared when the cpu reads the interrupt register, or sets the resume bit of this register
		1	Detect Suspend mode
0	SUSPENDEN	Set by the CPU to enable entry into suspend mode when suspend signaling is received on the bus	
		0	Disable.
		1	Enable.

20.7.3 USB_INTRIN1: Interrupt for EP0 + IN EP1 to EP5 Register

INTIN indicates which of the interrupts for IN Endpoints 1-5 are currently active. It also indicates whether the Endpoint 0 interrupt is currently active. All active interrupts will be cleared when this register is read.

USB_INTRIN1=0x3000_2008

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5	4	3	2	1	0
Reserved				EP5IFG	EP4IFG	EP3IFG	EP2IFG	EP1IFG	EP0IFG
-				0	0	0	0	0	0
-				R0	R0	R0	R0	R0	R0

5	EP5IFG	IN Endpoint 5 interrupt	
		0	Clear
		1	Issued interrupt IN EP5
4	EP4IFG	IN Endpoint 4 interrupt	
		0	Clear
		1	Issued interrupt IN EP5
3	EP3IFG	IN Endpoint 3 interrupt	
		0	Clear
		1	Issued interrupt IN EP3
2	EP2IFG	IN Endpoint 2 interrupt	
		0	Clear
		1	Issued interrupt IN EP2
1	EP1IFG	IN Endpoint 1 interrupt	
		0	Clear
		1	Issued interrupt IN EP1
0	EP0IFG	Endpoint 0 interrupt	
		0	Clear
		1	Issued interrupt EP0

20.7.4 USB_INTROUT1: Interrupt for OUT EP1 to EP5 Register

INTOUT indicates which of the interrupts for OUT Endpoints 1-5 are currently active.

All active interrupts will be cleared when this register is read.

USB_INTROUT1=0x3000_2010

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5	4	3	2	1	0
Reserved				EP5IFG	EP4IFG	EP3IFG	EP2IFG	EP1IFG	Reserved
-				0	0	0	0	0	-
-				R0	R0	R0	R0	R0	-

5	EP5IFG	IN Endpoint 5 interrupt	
		0	Clear
		1	Issued interrupt IN EP5
4	EP4IFG	IN Endpoint 4 interrupt	
		0	Clear
		1	Issued interrupt IN EP5
3	EP3IFG	IN Endpoint 3 interrupt	
		0	Clear
		1	Issued interrupt IN EP3
2	EP2IFG	IN Endpoint 2 interrupt	
		0	Clear
		1	Issued interrupt IN EP2
1	EP1IFG	IN Endpoint 1 interrupt	
		0	Clear
		1	Issued interrupt IN EP1

20.7.5 USB_INTRUSB: Interrupt for Common USB

INTRUSB indicates which USB interrupts are currently active.

All active interrupts will be cleared when this register is read.

USB_INTRUSB=0x3000_2018

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1	0
Reserved				SOFIFG	RESETIFG	RESUMEIFG	SUSPENDIFG
-				0	0	0	0
.				RO	RO	RO	RO

3	SOFIFG	Set at the start of each frame.	
		0	Clear
		1	Detect SOF
2	RESETIFG	Set when reset signaling is detected on the bus.	
		0	Clear
		1	Detect Reset
1	RESUMEIFG	Set when resume signaling is detected on the bus while this Device is in suspend mode.	
		0	Clear
		1	Detect Resume
0	SUSPENDIFG	Set when suspend signaling is detected on the bus.	
		0	Clear
		1	Detect Suspend

20.7.6 USB_INTRIN1E: Interrupt Enable for INTIN

INTINE provides interrupt enable bits for the interrupts in INTIN register. On reset, the bits corresponding to End point 0 and the IN endpoints included in the design are set to 1, while the remaining bits are set to 0.

USB_INTRIN1E=0x3000_201C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5	4	3	2	1	0
Reserved				EP5IEN	EP4IEN	EP3IEN	EP2IEN	EP1IEN	EP0IEN
-				1	1	1	1	1	1
-				RW	RW	RW	RW	RW	RW

5	EP5IEN	Interrupt enable for IN EP5	
		0	Disable.
		1	Enable.
4	EP4IEN	Interrupt enable for IN EP4	
		0	Disable.
		1	Enable.
3	EP3IEN	Interrupt enable for IN EP3	
		0	Disable.
		1	Enable.
2	EP2IEN	Interrupt enable for IN EP2	
		0	Disable.
		1	Enable.
1	EP1IEN	Interrupt enable for IN EP1	
		0	Disable.
		1	Enable.
0	EP0IEN	Interrupt enable for EP0	
		0	Disable.
		1	Enable.

20.7.7 USB_INTROUT1E: Interrupt Enable for INTOUT

INTOUTE provides interrupt enable bits for the interrupts in INTOUT register. On reset, the bits corresponding to the OUT endpoints included in the design are set to 1, while the remaining bits are set to 0.

USB_INTROUT1E=0x3000_2024

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5	4	3	2	1	0
Reserved				EP5IEN	EP4IEN	EP3IEN	EP2IEN	EP1IEN	Reserved
-				1	1	1	1	1	-
-				RW	RW	RW	RW	RW	-

5	EP5IEN	Interrupt enable for OUT EP5	
		0	Disable.
		1	Enable.
4	EP4IEN	Interrupt enable for OUT EP4	
		0	Disable.
		1	Enable.
3	EP3IEN	Interrupt enable for OUT EP3	
		0	Disable.
		1	Enable.
2	EP2IEN	Interrupt enable for OUT EP2	
		0	Disable.
		1	Enable.
1	EP1IEN	Interrupt enable for OUT EP1	
		0	Disable.
		1	Enable.

20.7.8 USB_INTRUSBE: Interrupt Enable for INTUSB

INTUSBE provides interrupt enable bits for each of the interrupts in INTUSB register.

USB_INTRUSBE=0x3000_202C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1	0
Reserved				SOFIEN	RESETIEN	RESUMEIEN	SUSPENDIEN
-				0	1	1	0
-				RW	RW	RW	RW

3	SOFIEN	Interrupt enable for SOF	
		0	Disable.
		1	Enable.
2	RESETIEN	Interrupt enable for Reset	
		0	Disable.
		1	Enable.
1	RESUMEIEN	Interrupt enable for Resume	
		0	Disable.
		1	Enable.
0	SUSPENDIEN	Interrupt enable for Suspend	
		0	Disable.
		1	Enable.

20.7.9 USB_FRAME1: Frame Number bit 0 to 7

FRM1 holds the lower 8 bits of the last received frame number.

USB_FRAME1=0x3000_2030

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			FRMNO[7:0]
-			0x00
-			R0

7 0	FRMNO[7:0]	Lower 8 bits of Frame Number

20.7.10 USB_FRAME2: Frame Number bit 8 to 10

FRM2 holds the upper 3 bits of the last received frame number.

USB_FRAME2=0x3000_2034

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			FRMNO[2:0]
-			0
-			R0

2 0	FRMNO[2:0]	Upper 3 bits of Frame Number

20.7.11 USB_INDEX: Index for Selecting Endpoint Register

INDEX determines which endpoint control/status registers are accessed at addresses 0x3000_2050 (USB_INMAXP) to 0x3000_206C (USB_OUTCOUNT2).

USB_INDEX=0x3000_2038

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			INDEX[3:0]
-			0
-			RW

3 0	INDEX[3:0]	Index bits to access any endpoint control/status registers. Each IN endpoint and each OUT endpoint have their own set of control/status registers. Only one set of IN control/status and one set of OUT control/status registers appear in the memory map at any one time. Before accessing an endpoint's control/status registers, the endpoint number should be written to the Index register to ensure that the correct control/status registers appear in the memory map.

20.7.12 USB_PUPDCR: USB DP/DM Pins Pull-up/Down Resistor Control Register.

USB_PUPDCR control Pull-up or Pull-down for DP/DN.

USB_PUODCR=0x3000_2040

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5	4	3 2	1	0
Reserved				DNPDEN	DNPUEN	Reserved	DPPDEN	DPPUEN
-				0	0	-	0	0
-				RW	RW	-	RW	RW

5	DNPDEN	USBDM Pin Pull-down resistor enable bit	
		0	Disable.
		1	Enable.
4	DNPUEN	USBDM Pin Pull-up resistor enable bit	
		0	Disable.
		1	Enable.
1	DPPDEN	DP Pin Pull-down resistor enable bit	
		0	Disable.
		1	Enable.
0	DPPUEN	DP Pin Pull-up resistor enable bit	
		0	Disable.
		1	Enable.

20.7.13 USB_PHYCR: USB Power Control Register.

USB_PHYCR control power.

USB_PHYCR=0x3000_2044

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			PHYDIS
-			1
-			RW

0	PHYDIS	USB PHY Master Disable bit
	0	USB PHY is controlled by suspend and resume signal of USB.
	1	Disable USB PHY.

20.7.14 USB_INMAXP: Maximum Package Size for IN Register

INMAXP holds the maximum packet size for transactions through the currently-selected IN endpoint – in units of 8 bytes, except that a value of 128 sets the maximum packet size to 1023 (the maximum size for an Isochronous packet transferred in a full-speed transaction) rather than 1024. In setting this value, you should note the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transactions in Full-speed operations.

USB_INMAXP=0x3000_2050

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			INMAXPS[7:0]
-			0x00
-			RW

7 0	INMAXPS[7:0]	Maximum Packet Size / transaction for IN endpoint. 8 bytes unit. (ex. 0x04 = 32 bytes, but 0x80 = 1023 bytes)

20.7.15 USB_CSR0: Control Status Register 0 for EP0 Register

CSR0 provides control and status bits for end point 0. It is reset to 0.

USB_CSR0=0x3000_2054

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			SERVICEDSETUPEND	SERVICEDOUTPKTRDY	SENDSTALL	SETUPEND	DATAEND	SENTSTALL	INPKTRDY	OUTPKTRDY
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RO	RW	RW	RW	RO

7	SERVICEDSETUPEND	The software writes a '1' to this bit to clear the SETUPEND bit. It is cleared automatically.
6	SERVICEDOUTPKTRDY	The software writes a '1' to this bit to clear the OUTPKTRDY bit. It is cleared automatically.
5	SENDSTALL	The software writes a '1' to this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically.
4	SETUPEND	This bit will be set when a control transaction ends before the DATAEND bit has been set. An interrupt will be generated and the FIFO flushed at this time. The bit is cleared by the software writing a '1' to the SERVICEDSETUPEND bit.
3	DATAEND	The software sets this bit: 1. When setting INPKTRDY for the last data packet. 2. When clearing OUTPKTRDY after unloading the last data packet. 3. When setting INPKTRDY for a zero-length data packet. It is cleared automatically.
2	SENTSTALL	This bit is set when a STALL handshake is transmitted. The software should clear this bit.
1	INPKTRDY	The software sets this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet has been transmitted. An interrupt is generated when the bit is cleared.
0	OUTPKTRDY	This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The software clears this bit by setting the SERVICEDOUTPKTRDY bit.

20.7.16 USB_INCSR1: Control Status Register 1 for IN Register

INCSR1 provides control and status bits for IN transactions through the currently selected endpoint. There is an INSCR1 register for each IN endpoint (not including Endpoint 0).

USB_INCSR1=0x3000_2054

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved				CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONOTEMPTY	INPKTRDY
-				0	0	0	0	0	0	0
-				RW	RW	RW	RW	RW	RW	RW

6	CLRDATATOG	The software writes a ‘1’ to this bit to reset the endpoint IN data toggle to ‘0’.
5	SENTSTALL	This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the INPKTRDY bit is cleared. The software should clear this bit.
4	SENDSTALL	The software writes a ‘1’ to this bit to issue a STALL handshake to an IN token. The software clears this bit to terminate the stall condition. This bit has no effect if the IN endpoint is in ISO mode.
3	FLUSHFIFO	The software writes a ‘1’ to this bit to flush the next packet to be transmitted from the endpoint IN FIFO. The FIFO pointer is reset and the INPKTRDY bit is cleared. It is cleared automatically. **NOTE:** 1. If the FIFO contains two packets, FlushFIFO will need to be set twice to completely clear the FIFO.
2	UNDERRUN	In ISO mode, this bit is set when a zero-length data packet is sent after receiving an IN token with the INPKTRDY bit not set. In Bulk/Interrupt mode, this bit is set when a NAK is returned in response to an IN token. The software should clear this bit.
1	FIFONOTEMPTY	This bit is set when there is at least one packet in the IN FIFO.
0	INPKTRDY	The software sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

20.7.17 USB_INCSR2: Control Status Register 2 for IN Register

INCSR2 provides further control bit for IN transactions through the currently-selected endpoint. There is an INCSR2 register for each IN endpoint (not including Endpoint 0)

USB_INCSR2=0x3000_2058

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2 1 0
Reserved			AUTOSET	ISO	MODE	DMAENAB	FRCDATATOG	Reserved
-			0	0	0	0	0	-
-			RW	RW	RW	RW	RW	-

7	AUTOSET	If the software sets this bit, then INPKTRDY will be automatically set when data of the maximum packet size (value in INMAXP) is loaded into the IN FIFO. If a packet of less than the maximum packet size is loaded, then INPKTRDY will have to be set manually.
6	ISO	The software sets this bit to enable the IN endpoint for isochronous transfers and clears it to enable the IN endpoint for bulk or interrupt transfers.
5	MODE	The software sets this bit to enable the endpoint direction as IN and clears it to enable the endpoint direction as OUT. Valid only where the same endpoint FIFO is used for both IN and OUT transactions.
4	DMAENAB	The software sets this bit to enable the DMA request for the IN endpoint.
3	FRCDATATOG	The software sets this bit to force the endpoint IN data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by interrupt IN endpoints that are used to communicate rate feedback for isochronous endpoints.

20.7.18 USB_OUTMAXP: Maximum Package Size for OUT Register

OUTMAXP holds the maximum packet size for transactions through the currently selected OUT endpoint - in units of 8 bytes, except that a value of 128 sets the maximum packet size to 1023 (the maximum size for an isochronous packet) rather than 1024. In setting this value, you should note the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt, and Isochronous transactions in Full-speed operations.

There is an OUTMAXP register for each OUT endpoint (except Endpoint 0).

The value written to this register should match the MaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint. A mismatch could cause unexpected results.

The Total amount of data represented by the value written to this register must not exceed the FIFO size for the OUT endpoint and should not exceed half the FIFO size if double-buffering is required. If a value greater than the configured OUT FIFO size for the endpoint is written to this register, the value will be automatically changed to the OUT FIFO size. If the value written to this register is less than, or equal to, half the OUT FIFO size, two OUT packets can be buffered.

USB_OUTMAXP=0x3000_205C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			OUTMAXPS[7:0]
-			0x00
-			RW

7 0	OUTMAXPS[7:0]	Maximum Packet Size / transaction

20.7.19 USB_OUTCSR1: Control Status Register 1 for OUT Register

OUTSCR1 provides control and status bits for OUT transactions through the currently selected endpoint. It is reset to 0.

USB_OUTCSR1=0x3000_2060

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved			CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERROR	OVERRUN	FIFOFULL	OUTPKTRDY
-			0	0	0	0	0	0	0	0
-			RW	RW	RW	RW	RO	RW	RO	RW

7	CLRDATATOG	The software writes a ‘1’ to this bit to reset the endpoint data toggle to 0.
6	SENTSTALL	This bit is set when a STALL handshake is transmitted. The software should clear this bit.
5	SENDSTALL	The software writes a ‘1’ to this bit to issue a STALL handshake. The software clears this bit to terminate the stall condition. This bit has no effect if the OUT endpoint is in ISO mode.
4	FLUSHFIFO	The software writes a ‘1’ to this bit to flush the next packet to be read from the endpoint OUT FIFO. It is cleared automatically. **NOTE:** 1. If the FIFO contains two packets, FLUSHFIFO will need to be set twice to completely clear the FIFO.
3	DATAERROR	This bit is set when OUTPKTRDY is set if the data packet has a CRC or bit-stuff error. it is cleared when OUTPKTRDY is cleared. The bit is only valid in ISO mode.
2	OVERRUN	This bit is set if an OUT packet cannot be loaded into the OUT FIFO. The CPU should clear this bit. The bit is only valid in ISO mode.
1	FIFOFULL	This bit is set when no more packets can be loaded into the OUT FIFO.
0	OUTPKTRDY	This bit is set when a data packet has been received. The software should clear this bit when the packet has been unloaded from the OUT FIFO. An interrupt is generated when the bit is set.

20.7.20 USB_OUTCSR2: Control Status Register 2 for OUT Register

OUTSCR1 provides control and status bits for OUT transactions through the currently selected endpoint. It is reset to 0.

USB_OUTCSR2=0x3000_2064

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6	5	4	3 2 1 0
Reserved			AUTOCLEAR	ISO	DMAENAB	DMAMODE	Reserved
-			0	0	0	0	-
-			RW	RW	RW	RW	-

7	AUTOCLEAR	If the software sets this bit, then the OUTPKTRDY bit will be automatically cleared when a packet of OUTMAXP bytes has been unloaded from the OUT FIFO. When packets of less than the maximum packet size are unloaded, OUTPKTRDY will have to be cleared manually.
6	ISO	The software sets this bit to enable the OUT endpoint for isochronous transfers and clears it to enable the OUT endpoint for bulk or interrupt transfers.
5	DMAENAB	The software sets this bit to enable the DMA request for the OUT endpoint.
4	DMAMODE	DMA Operation Mode Select bit 0 = DMA Mode 0 in which a DMA request is generated for all received packets, together with an interrupt (if enabled). It can be used with any endpoint (Control, Bulk, Isochronous, or Interrupt transactions). 1 = DMA Mode 1 in which a DMA request (but no interrupt) is generated for OUT packets of size OUTMAXP bytes, and an interrupt (but no DMA request) is generated for OUT packets of any other size. It can only be used with endpoints for Bulk transactions.

20.7.21 USB_COUNT0: Number of Bytes for EP0 Register

OUTCNT0 indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned is valid while OUTPKTRDY(INCSR0[0]) is set.

USB_COUNT0=0x3000_2068

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			EP0OUTCNT[6:0]
-			0x00
-			R0

6 0	EP0OUTCNT[6:0]	Endpoint 0 Out Count

20.7.22 USB_OUTCOUNT1: Number of Bytes for Lower Byte Register

OUTCNT1 holds the lower 8 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY is set.

USB_OUTCOUNT1=0x3000_2068

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			EPOUTCNT[7:0]
-			0x00
-			R0

c

7 0	EPOUTCNT[7:0]	Endpoint Out count – lower 8 bits

20.7.23 USB_OUTCOUNT2: Number of Bytes for Upper Byte Register

OUTCNT2 holds the upper 3 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY is set.

USB_OUTCOUNT2=0x3000_206C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			EPOUTCNT[10:8]
-			0
-			R0

c

2 0	EPOUTCNT[10:8]	Endpoint Out count – upper 3 bits

20.7.24 USB_FIFOn: FIFO for EP0 to EP5 Register

This address range provides six addresses for CPU access to the FIFOs for each endpoint. Writing to these addresses loads data into the IN FIFO for the corresponding endpoint. Reading from these addresses unloads data from the OUT FIFO for the corresponding endpoint.

USB_FIFO0=0x3000_2080, USB_FIFO1=0x3000_2090, USB_FIFO2=0x3000_20A0
USB_FIFO3=0x3000_20B0, USB_FIFO4=0x3000_20C0, USB_FIFO5=0x3000_20D0

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Reserved			FIFO[7:0]
-			0x00
-			RW

7 0	FIFO[7:0]	IN/OUT FIFO for each endpoint n

20.7.25 USB_DMAINTR: USB DMA Interrupt Pending Register

USB_DMAINTR indicates whether the USB DMA x interrupt is currently active.

USB_DMAINTR=0x3000_2200

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1	0
Reserved				UDMA2IFG	UDMA1IFG
-				0	0
-				RO	RO

1 0	UDMAxIFG	USB DMA x Interrupt flag bit x: 1 and 2 This bit indicates whether the USB DMA x interrupt is currently active.

NOTE:

1. **All** interrupt flag bits of this register will be cleared when this register is read.

20.7.26 USB_DMA1CNTL: USB DMA Channel 1 Control Register

USB_DMA1CNTL provides control and status bits for DMA channel 1.

USB_DMA1CNTL=0x3000_2204

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14 13 12 11 10 9 8	7 6 5 4	3	2	1	0
Reserved		BUSERR	MAXPS[6:0]	EPNO[3:0]	DMAIEN	MODE	DIR	DMAENAB
-		0	0	0	0	0	0	0
-		RO	RW	RW	RW	RW	RW	RW

15	BUSERR	BUS Error bit	
		0	No bus error occurred
		1	Bus error occurred. The DMA controller will immediately terminate the DMA transfer and interrupt the processor with this bit set.
14 8	MAXPS[6:0]	DMA 1 Maximum Packet Size in units of 8 bytes. Required for Mode 1 only	
7 3	EPNO[3:0]	DMA 1 Endpoint Number bits	
3	DMAIEN	DMA 1 Interrupt Enable bit	
		0	Disable
		1	Enable
2	MODE	DMA 1 Interrupt Enable bit	
		0	DMA Mode 0 (DMA requests every packet)
		1	DMA Mode 1 (DMA requests every OUTMAXP)
1	DIR	DMA 1 Transfer Direction Select bit	
		0	OUT endpoint. Transfer is from FIFO to memory through DMA buffer
		1	IN endpoint. Transfer is from memory to FIFO through DMA buffer
0	DMAENAB	DMA 1 Enable bit	
		0	Disable
		1	Enable

20.7.27 USB_DMA1ADDR: USB DMA Channel 1 Memory Address Register

USB_DMA1ADDR provides Memory Address for DMA channel 1.

USB_DMA1ADDR=0x3000_2208

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ADDR[31:0]			
0x0000_0000			
RW			

31 0	ADDR[31:0]	DMA Channel 1 Memory Address bits

20.7.28 USB_DMA1COUNT: USB DMA Channel 1 Transfer Byte Count Register

USB_DMA1COUNT provides Transfer byte count value for DMA channel 1.

USB_DMA1COUNT=0x3000_220C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
TRSCNT[31:0]			
0x0000_0000			
RW			

31 0	TRSCNT[31:0]	DMA Channel 1 Transfer Byte Count bits

20.7.29 USB_DMA2CNTL: USB DMA Channel 2 Control Register

USB_DMA2CNTL provides control and status bits for DMA channel 1.

USB_DMA2CNTL=0x3000_2214

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14 13 12 11 10 9 8	7 6 5 4	3	2	1	0
Reserved		BUSERR	MAXPS[6:0]	EPNO[3:0]	DMAIEN	MODE	DIR	DMAENAB
-		0	0	0	0	0	0	0
-		RO	RW	RW	RW	RW	RW	RW

15	BUSERR	BUS Error bit	
		0	No bus error occurred
		1	Bus error occurred. The DMA controller will immediately terminate the DMA transfer and interrupt the processor with this bit set.
14 8	MAXPS[6:0]	DMA 2 Maximum Packet Size in units of 8 bytes. Required for Mode 1 only	
7 3	EPNO[3:0]	DMA 2 Endpoint Number bits	
3	DMAIEN	DMA 2 Interrupt Enable bit	
		0	Disable
		1	Enable
2	MODE	DMA 2 Interrupt Enable bit	
		0	DMA Mode 0 (DMA requests every packet)
		1	DMA Mode 1 (DMA requests every OUTMAXP)
1	DIR	DMA 2 Transfer Direction Select bit	
		0	OUT endpoint. Transfer is from FIFO to memory through DMA buffer
		1	IN endpoint. Transfer is from memory to FIFO through DMA buffer
0	DMAENAB	DMA 2 Enable bit	
		0	Disable
		1	Enable

20.7.30 USB_DMA2ADDR: USB DMA Channel 2 Memory Address Register

USB_DMA2ADDR provides Memory Address for DMA channel 2.

USB_DMA2ADDR=0x3000_2218

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ADDR[31:0]			
0x0000_0000			
RW			

31 0	ADDR[31:0]	DMA Channel 2 Memory Address bits

20.7.31 USB_DMA2COUNT: USB DMA Channel 2 Transfer Byte Count Register

USB_DMA2COUNT provides Transfer byte count value for DMA channel 2.

USB_DMA2COUNT=0x3000_221C

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
TRSCNT[31:0]			
0x0000_0000			
RW			

31 0	TRSCNT[31:0]	DMA Channel 2 Transfer Byte Count bits

20.7.32 USB Register Map Summary

Table 99. USB Register Map Summary

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USB_FADDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UPDATE	FADDR[6:0]						
	Reset value																									0	0	0	0	0	0	0	0
0x04	USB_POWER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ISOUPDATE	Res.	Res.	Res.	RESET	RESUME	SUSPENDMD	SUSPENDEN
	Reset value																									0				0	0	0	0
0x08	USB_INTRIN1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EP5IFG	EP4IFG	EP3IFG	EP2IFG	EP1IFG	EP0IFG
	Reset value																											0	0	0	0	0	0
0x10	USB_INTROUT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EP5IFG	EP4IFG	EP3IFG	EP2IFG	EP1IFG	Res.
	Reset value																											0	0	0	0	0	
0x18	USB_INTRUSB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOFIFG	RESETIFG	RESUMEIFG	SUSPENDIFG
	Reset value																													0	0	0	0
0x1C	USB_INTRIN1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EP5IEN	EP4IEN	EP3IEN	EP2IEN	EP1IEN	EP0IEN
	Reset value																											1	1	1	1	1	1
0x24	USB_INTROUT1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EP5IEN	EP4IEN	EP3IEN	EP2IEN	EP1IEN	Res.
	Reset value																											1	1	1	1	1	
0x2C	USB_INTRUSBE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOFIEN	RESETIEN	RESUMEIEN	SUSPENDIEN
	Reset value																													0	1	1	0

Table 99. USB Register Map Summary (continued)

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	USB_FRAME1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRMNO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	USB_FRAME2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRMNO[10:8]		
	Reset value																														0	0	0
0x38	USB_INDEX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INDEX[3:0]			
	Reset value																													0	0	0	0
0x40	USB_PUPDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DNPDEN	DNPUEN	Res.	Res.	DPPDEN	DPPUEN
	Reset value																											0	0			0	0
0x44	USB_PHYCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PHYDIS
	Reset value																																1
0x50	USB_INMAXP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INMAXPS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x54	USB_CSR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SERVICEDSETUPEND	SERVICEDOUTPKTRDY	SENDSTALL	SETUPEND	DATAEND	SENTSTALL	INPKTRDY	OUTPKTRDY
	Reset value																									0	0	0	0	0	0	0	0
0x54	USB_INCSR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONOTEMPTY	INPKTRDY
	Reset value																										0	0	0	0	0	0	0
0x58	USB_INCSR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AUTOSET	ISO	MODE	DMAENAB	FRCDATATOG	Res.	Res.	Res.
	Reset value																									0	0	0	0	0			

Table 99. USB Register Map Summary (continued)

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	USB_OUTMAXP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OUTMAXPS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x60	USB_OUTCSR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERROR	OVERRUN	FIFOFULL	OUTPKTRDY
	Reset value																									0	0	0	0	0	0	0	0
0x64	USB_OUTCSR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AUTOCLEAR	ISO	DMAENAB	DMAMODE	Res	Res	Res	Res
	Reset value																									0	0	0	0				
0x68	USB_COUNT0	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EP0OUTCNT[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x68	USB_OUTCOUNT 1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EPOUTCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x6C	USB_OUTCOUNT 2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EPOUTCNT[10:8]		
	Reset value																														0	0	0

Table 99. USB Register Map Summary (continued)

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x80	USB_FIFO0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x90	USB_FIFO1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xA0	USB_FIFO2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xB0	USB_FIFO3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xC0	USB_FIFO4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0xD0	USB_FIFO5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x200	USB_DMAINTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDMA2IFG	UDMA1IFG
	Reset value																															0	0
0x204	USB_DMA1CNTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSERR	MAXPS[6:0]							EPNO[3:0]				DMAIEN	MODE	DIR	DMAENAB
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x208	USB_DMA1ADDR	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20C	USB_DMA1COUNT	TRSCNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x214	USB_DMA2CNTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSERR	MAXPS[6:0]							EPNO[3:0]				DMAIEN	MODE	DIR	DMAENAB
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x218	USB_DMA2ADDR	ADDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x21C	USB_DMA2COUNT	TRSCNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Abbreviations for Registers

The abbreviations in the table below are those used in the register descriptions:

Table 100. Abbreviations for Registers

Abbreviation	Definition
Read and write (RW)	Software can read and write to this bit.
Read-only (RO)	Software can only read this bit.
Write-only (WO)	Software can only write to this bit. Reading this bit returns the reset value.
Read and clear (RC)	Software can read to clear this bit.
Read and write 0 clear (RWC0)	Software can read and clear this bit by writing 0.
Read and write 1 clear (RWC1)	Software can read and clear this bit by writing 1.
Reserved (Res.)	Reserved bit, must be kept at reset value.

Glossary

This section gives a brief definition of acronyms and abbreviations used in this document:

1. **Word**: Data of 32-bit length.
2. **Half-word**: Data of 16-bit length.
3. **Byte**: Data of 8-bit length.
4. **PGM**: Flash programming.
5. **AHB**: Advanced high-performance bus.
6. **APB**: Advanced peripheral bus.

Revision History

Revision	Date	Notes
1.00	Jan. 06, 2025	Initial release.

Korea

Regional Office, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
www.abovsemi.com

HQ, Ochang
R&D, QA, and Test Center
37, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,
28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
www.abovsemi.com

Domestic Sales Manager
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

Global Sales Manager
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

China Sales Manager
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr

ABOV Disclaimer

IMPORTANT NOTICE – PLEASE READ CAREFULLY

ABOV Semiconductor (“ABOV”) reserves the right to make changes, corrections, enhancements, modifications, and improvements to ABOV products and/or to this document at any time without notice. **ABOV DOES NOT GIVE WARRANTIES AS TO THE ACCURACY OR COMPLETENESS OF THE INFORMATION INCLUDED HEREIN.** Purchasers should obtain the latest relevant information of ABOV products before placing orders. Purchasers are entirely responsible for the choice, selection, and use of ABOV products and ABOV assumes no liability for application assistance or the design of purchasers’ products. **NO LICENSE, EXPRESS OR IMPLIED, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY ABOV HEREIN. ABOV DISCLAIMES ALL EXPRESS AND IMPLIED WARRANTIES AND SHALL NOT BE RESPONSIBLE OR LIABLE FOR ANY INJUIRES OR DAMAGES RELATED TO USE OF ABOV PRODUCTS IN SUCH UNAUTHORIZED APPLICATIONS.** ABOV and the ABOV logo are trademarks of ABOV. For additional information about ABOV trademarks, please refer to https://www.abov.co.kr/en/about/corporate_identity.php. All other products or service names are the property of their respective owners. Information in this document supersedes and replaces the information previously supplied in any former versions of this document.

© 2025 ABOV Semiconductor – All rights reserved