Brief Description

The *ZAMC4100 Actuator and Motor Controller* is an integrated, single package solution that features a high-performance ARM® Cortex™-M0* microcontroller (MCU), a 10-bit ADC, four general purpose analog inputs, four half-bridge drivers, four high-side switches, one output buffer for electrochromatic (EC) mirror control with programmable output voltage (6 bit DAC), current source for external diode-based temperature measurements, switchable voltage source for external resistive sensors, and LIN (Local Interconnect Network) bus interface.

This single-package solution with its rich diagnostic features, optimized thermal performance, and LIN bus interface, is well suited for automotive applications, such as high-end automotive exterior mirror controllers, where small form-factor circuit board designs and light wiring harnesses are required.

Features

- Two main operation modes
  - NORMAL Mode with all functionality available
  - SLEEP Mode with very low current consumption (less than 80µA)
- 10-bit ADC for measuring various parameters
- 4 general purpose ADC inputs configurable for absolute or ratiometric mode
- 4 low RDS (ON) half-bridge drivers and 4 low RDS (ON) high-side switches
- Analog EC mirror output controlled via 6-bit DAC
- All outputs short circuit protected
- Hardware 8-bit PWM control for half-bridge and high-side drivers
- Over-temperature protection with automatic driver shut-down
- Over-voltage and under-voltage detection with automatic driver shut-down
- Overload and open/short diagnostic for all outputs
- Embedded LIN 2.2/SAE J2602-2 transceiver
- Embedded ARM® Cortex™-M0 microcontroller
- 8 user-configurable GPIO pins

Benefits

- Multi-chip-module concept with high functional integrity
- Low number of external components
- Small footprint plastic QFN package with exposed pad allowing better thermal management
- AEC-Q100 qualified product, optimized for automotive environment
- Smart power management concept for achieving low sleep current consumption (<80µA)

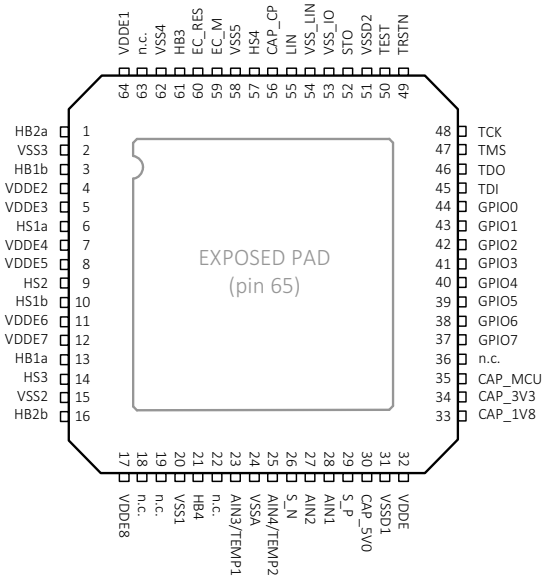
Available Support

- Data sheet and application notes
- Evaluation and Application Kits
- Software Development Kit (SDK)

Physical Characteristics

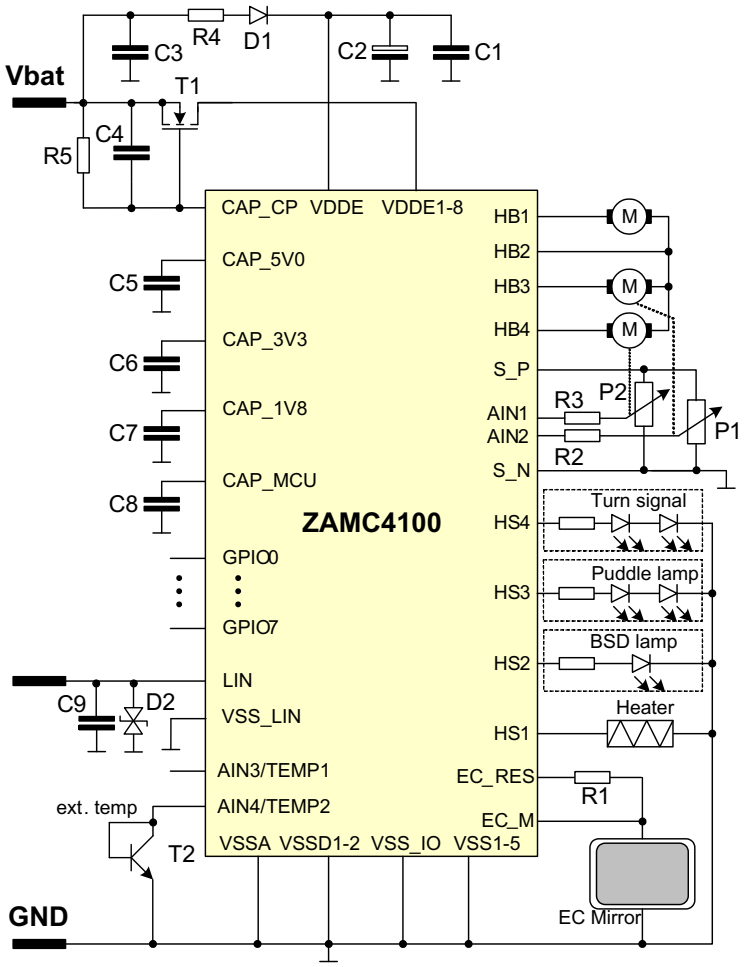
- Ambient operation temperature: -40°C to +85°C
- Wide power supply voltage range: 6V to 18V
- Plastic QFN64 9x9mm package; exposed die pad

ZAMC4100
PQFN64 TOP VIEW



* Note: The ARM® and Cortex™ trademarks are owned by ARM, Ltd.

ZAMC4100 Application Circuit for Automotive Mirror Control

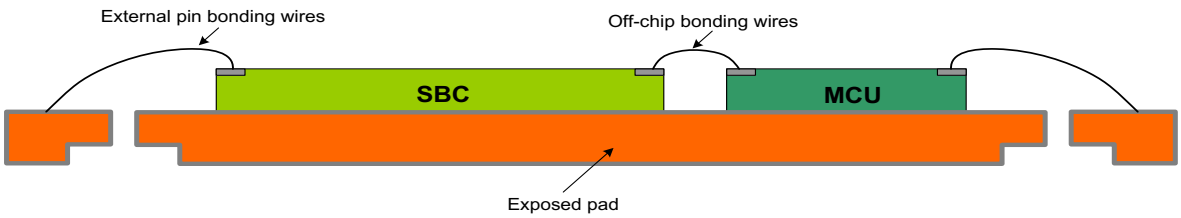


Typical Applications

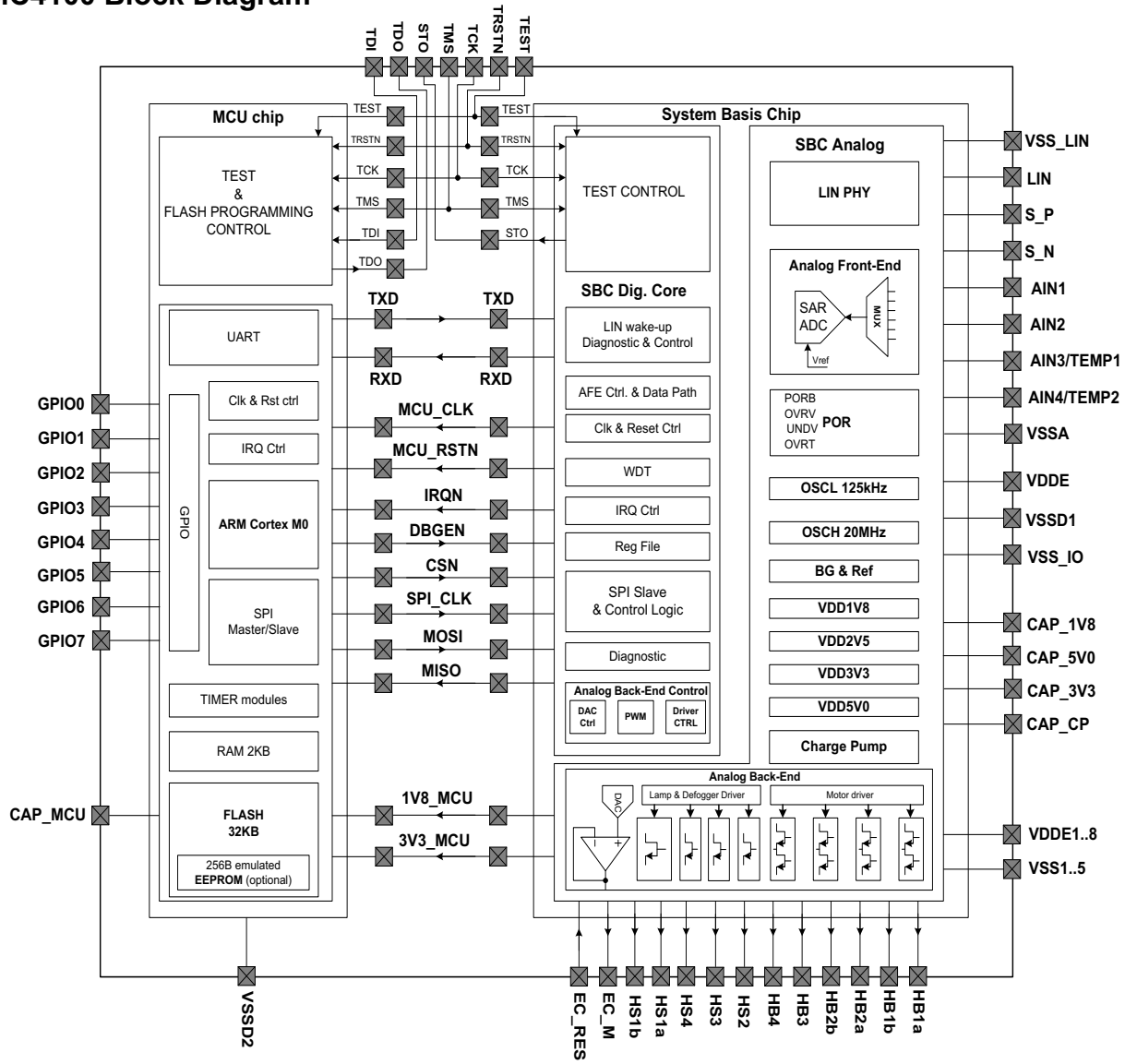
- Automotive exterior rear view mirror controllers
- Intelligent DC motor drivers
- Stand alone or slave actuator controllers
- Power management for low voltage electrical systems
- Single chip solution for optimized white goods' controllers

Note: Refer to the *ZAMC4100 Data Sheet* for the values and functions of the external components in the application circuit.

Multi-Chip-Module Assembly: Microcontroller Unit (MCU) and System Basis Chip (SBC)



ZAMC4100 Block Diagram



SENSORS INTERFACE PINS	DESCRIPTION
S_P	Positive supply for sensors (5V)
S_N	Negative supply for sensors (0V)
AIN1	General purpose ADC input
AIN2	General purpose ADC input
AIN3/TEMP1	General purpose ADC input/temperature sensor input
AIN4/TEMP2	General purpose ADC input/temperature sensor input

ACTUATORS INTERFACE PINS	DESCRIPTION
HB1a	Output of half-bridge driver 1
HB1b	
HB2a	Output of half-bridge driver 2
HB2b	
HB3	Output of half-bridge driver 3
HB4	Output of half-bridge driver 4
HS1a	Output of high-side switch 1
HS1b	
HS2	Output of high-side switch 2
HS3	Output of high-side switch 3
HS4	Output of high-side switch 4
EC_M	Output for EC mirror control
EC_RES	Connection to external resistor for EC
SERVICE PINS	**DESCRIPTION**
CAP_5V0	Connection to external capacitor for 5V (SBC analog core supply)
CAP_3V3	Connection to external capacitor for 3.3V (MCU peripheral supply)
CAP_1V8	Connection to external capacitor for 1.8V (SBC digital core supply)
CAP_MCU	Connection to external capacitor for 1.8V (MCU core supply)
CAP_CP	Connection to external capacitor for CP
EXPOSED PAD	Connected to digital ground

COMMUNICATION INTERFACE PINS	DESCRIPTION
LIN	LIN bus
GPIO0 to GPIO7	General purpose I/O pins
TEST INTERFACE PINS	**DESCRIPTION**
TEST	Global test enable pin
TRSTN	JTAG Low-active Reset
TCK	JTAG clock
TMS	JTAG Mode Select
TDO	JTAG data out pin
TDI	JTAG data in pin
STO	SBC test data out
SUPPLY PINS	**DESCRIPTION**
VDDE	Supply for the system
VDDE1 to VDDE8	High current supply for the drivers
VSS1 to VSS5	High current ground for drivers
VSS_LIN	Ground for LIN transceiver
VSSA	Ground for analog blocks (low noise)
VSSD1	Ground for digital blocks of SBC
VSSD2	Ground for MCU
VSS_IO	Ground for SBC digital IOs

Ordering Information

Product Sales Code	Description	Package
ZAMC4100GA2R	ZAMC4100GA PQFN64 – Temperature range: -40°C to +85°C	13” Tape and Reel
ZAMC4100GA2V	ZAMC4100GA PQFN64 – Temperature range: -40°C to +85°C	Tray
ZAMC4100 Evaluation Kit V2.0 including the ZAMC4100 Evaluation Board; software available at *www.IDT.com/ZAMC4100*		
ZAMC4100 Application Kit V1.0 including the ZAMC4100 Application Board		

Contents

1 IC Characteristics 14
1.1. Absolute Maximum Ratings 14
1.2. Electrical Parameters 15
1.2.1. Supply and Supply Monitoring 15
1.2.2. Digital Interface (GPIOs) 17
1.2.3. LIN Physical Layer 17
1.2.4. Output Drivers 18
1.2.5. Internal Reference Sources and Sensors 21
1.2.6. Analog-to-Digital Converter 22
1.3. Timing Parameters 22
1.3.1. Oscillators (OSCL and OSCH) 22
1.3.2. Analog-to-Digital Converter 23
1.3.3. SPI Bus Timing 24
1.3.4. System Power-up Timing 25
1.4. MCU Flash Memory Parameters 26
2 System Blocks Description 27
2.1. Power Supply 27
2.2. High-Frequency and Low-Frequency Oscillator 28
2.3. Watchdog Timer 28
2.4. Analog-to-Digital Converter 28
2.5. Output Drivers 29
2.6. Driver's Power Capability and Package Temperature 29
2.7. LIN Interface 30
2.8. SPI Bus 31
2.9. Diagnostic Functionalities 31
2.10. Microcontroller ARM® Cortex™ M0 31
2.10.1. MCU GPIOs 32
2.10.2. MCU Timer Module 32
2.10.3. MCU SPI Master 32
2.10.4. MCU LIN Controller 32
3 ZAMC4100 SBC Functional Description 33
3.1. System Power-Up Sequence 33
3.2. Clock and Reset Sources of the ZAMC4100 34
3.2.1. Clock Sources 34
3.2.2. Reset Sources 35
3.3. Watchdog Timer (WDT) 38
3.3.1. WDT Structure and Operation Principals 38

© 2019 Renesas Electronics Corporation

3.3.2. WDT during SLEEP and STANDBY Mode 40
3.3.3. WDT Configuration Register 40
3.4. System Supply Monitoring 41
3.4.1. Over-Voltage and Under-Voltage Detection 41
3.4.2. ZAMC4100 Operation during Over-Voltage 42
3.4.3. ZAMC4100 Operation during Under-Voltage 42
3.4.4. Charge Pump Voltage Monitoring 42
3.5. ZAMC4100 System Operational Modes 43
3.5.1. NORMAL Mode 43
3.5.2. SLEEP Mode 44
3.5.3. STANDBY Mode 46
3.5.4. Supported Wake-up Sources 46
3.5.5. Operational Modes Control 47
3.6. SBC Interrupts 49
3.6.1. SBC Interrupts: Enabling, Flagging, and Clearing Procedure 50
3.6.2. SBC Global Interrupts Registers 50
3.7. Serial Peripheral Interface (SPI) Bus Interface 54
3.7.1. SBC and MCU SPI Connection Principles 55
3.7.2. SPI Write Access 56
3.7.3. SPI Pipelined Write/Read Access 57
3.7.4. SPI Read Access 57
3.7.5. SPI Slave Status Byte 58
3.8. Half-Bridge Drivers 59
3.8.1. Half-Bridge Drivers Control and Status Register 60
3.8.2. Half-Bridge Over-Current Protection 61
3.8.3. Half-Bridge Driver Output Diagnostic Check 63
3.9. High-Side Drivers 65
3.9.1. High-Side Control and Status Register 66
3.9.2. High-Side Over-Current Protection 66
3.9.3. High-Side Drivers Output Diagnostic Check 66
3.10. Pulse Width Modulation (PWM) 68
3.10.1. Overview 68
3.10.2. PWM Signal Generation 69
3.10.3. Drivers PWM Control 70
3.10.4. PWM Initialization 71
3.10.5. PWM Module in SLEEP and STANDBY Mode 72
3.11. EC Mirror Driver 72
3.11.1. ECM Driver Current/Voltage Limitation and Protection against Short to GND or Vbat 73

© 2019 Renesas Electronics Corporation

3.11.2. ECM Control Register 75
3.11.3. ECM Driver Output Diagnostic Check 76
3.12. ZAMC4100 Analog to Digital Converter 77
3.12.1. ADC Overview 77
3.12.2. ADC External Inputs Configuration 79
3.12.3. ADC Input MUX Control 81
3.12.4. ADC Voltage References and Result Calculation 84
3.12.5. ADC Configuration and ADC Status Register 85
3.12.6. ADC Sample Rate 88
3.12.7. ADC Single and Continuous Conversion 88
3.12.8. ADC Result Formatting Options 89
3.12.9. ADC Comparator and Counter 91
3.12.10. External ADC Analog Inputs Open/Short Diagnostics and Protection 92
3.13. LIN Physical Layer (LIN PHY) 95
3.13.1. Overview 95
3.13.2. LIN PHY Operation 96
3.14. SBC Registers 99
3.14.1. SBC Registers Address Map 100
3.14.2. Registers Write Access Signatures 101
3.15. SBC Trimming 102
4 ZAMC4100 MCU Functional Description 103
4.1. Overview 103
4.2. Memory Organization 104
4.2.1. Accessing Invalid Memory Location 105
4.2.2. FLASH Memory 105
4.2.3. INFO Area 107
4.2.4. RAM Memory 2kB 107
4.3. MCU Clock and Reset Sources 108
4.4. MCU INTERRUPTS 110
4.4.1. Interrupts Organization 110
4.4.2. Interrupts Configuration 111
4.5. GPIO Module 111
4.5.1. Overview 111
4.5.2. GPIO Input Mode (default) 111
4.5.3. GPIO Output Mode 112
4.5.4. GPIO as Timer Trigger Operation 112
4.5.5. GPIO Interrupt Functionality 112
4.5.6. GPIO Registers 112
4.6. 32-Bit TIMER Module 112

© 2019 Renesas Electronics Corporation

4.6.1. Timer Mode (MODETC = 0) 113
4.6.2. Counter Mode (MODETC = 1) 113
4.6.3. TIMER Module Interrupt 113
4.6.4. TIMER Module Registers 113
4.7. SysTick 113
4.8. Master SPI Module 114
4.8.1. Overview 114
4.8.2. Setup Requirements for the Slave SPI of SBC 114
4.8.3. SPI Data Transfer Process 115
4.8.4. Continuous SPI Data Streaming 115
4.8.5. Abrupt SPI Discontinuity 115
4.8.6. Interrupts and Status Flags in the Z1_SPISTAT Register 115
4.8.7. Syncing the Data Transfer Rate via a Software Pause 116
4.8.8. Master SPI Registers 116
4.9. SW-LIN Module 117
4.9.1. The Inactivity Timer 117
4.9.2. The BREAK/SYNC Field Detector 118
4.9.3. SW-LIN Data Unit 118
4.9.4. Description of the Receive Operation 119
4.9.5. Description of Transmit Operation 120
4.9.6. General Remarks for SW-LIN Usage 122
4.9.7. SW-LIN Registers 122
4.10. MCU Registers Description 123
4.10.1. System Registers 123
4.10.2. GPIO Registers 125
4.10.3. 32 Bit Timer Registers 128
4.10.4. SPI Registers 130
4.10.5. SW-LIN Registers 132
4.10.6. FLASH Memory Registers 136
5 ZAMC4100 Protection and Diagnostic Features 140
5.1. Full-Time Diagnostic and Protection Features 140
5.1.1. Over-Current (OC) Protection 140
5.1.2. Over-Load (OVL) Detection using the ADC 140
5.1.3. Over-Temperature (OVT) Protection 141
5.1.4. Over-Voltage (OV) Detection 141
5.1.5. Under-Voltage (UV) Detection 141
5.1.6. Short to GND or Vbat Protection of Sensor Supply Pin S_P and Analog Input Pins AIN1 and AIN2 141
5.1.7. OSCL Output Clock Failure Detection 141

© 2019 Renesas Electronics Corporation

5.2. MCU Activated Diagnostic Functions 142
5.2.1. Open-Load Detection 142
5.2.2. Short to Vbat or GND Detection 142
5.2.3. Floating Analog Input Detection 142
5.2.4. Short or Open Sensor Supply Detection 143
5.2.5. ADC Ratiometric Test 143
5.2.6. ADC Test Absolute 143
5.2.7. Reference Voltage Test 144
5.2.8. 5.0V Supply Test 144
5.2.9. 2.5V Supply Test 144
5.2.10. Current Multiplexer Test 145
6 Application Circuit and External Components 146
6.1. Application Circuit Diagram for Mirror Controller 146
6.2. External Components for Automotive Mirror Control Application 147
7 Multi-chip Module Assembly, Pin Layout, and Pin Assignments 148
8 ZAMC4100 Outline Dimensions: Plastic QFN64 9x9mm (Package Type 1) 152
9 Ordering Information 153
10 Related Documents 153
11 Glossary 153
12 Document Revision History 155

List of Figures

Figure 1.1 ADC Timing Parameters Definition 23
Figure 1.2 SPI Communication Timing Diagram 24
Figure 1.3 ZAMC4100 System Power-Up Sequence 25
Figure 2.1 ZAMC4100 Supply Structure 27
Figure 2.2 ZAMC4100 LIN Interface and Open Systems Interconnection (OSI) Layers Coverage 30
Figure 2.3 LIN Slave Node Implementation in the ZAMC4100 30
Figure 3.1 SBC Power-Up State Diagram 33
Figure 3.2 ZAMC4100 Clock Sources 35
Figure 3.3 ZAMC4100 Reset Sources 36
Figure 3.4 ZAMC4100 Watchdog Timer 38
Figure 3.5 WDT Operation 39
Figure 3.6 ZAMC4100 Operational Modes 43
Figure 3.7 ZAMC4100 SLEEP and STANDBY Mode Behavior 45
Figure 3.8 SBC Interrupt Logic Organization 49
Figure 3.9 MCU-to-SBC Connection via SPI 55

© 2019 Renesas Electronics Corporation

Figure 3.10 SPI Frames 56
Figure 3.11 SPI Single or Burst Access 57
Figure 3.12 Structure of the Half-Bridge Drivers 59
Figure 3.13 Operation Principle of Over-Current Counter Integrator 62
Figure 3.14 Structure of High-Side Drivers 65
Figure 3.15 PWM Block Diagram 68
Figure 3.16 PWM Output Signal 69
Figure 3.17 Recommended PWM Initialization Procedure with HS3 as an Example 71
Figure 3.18 Effect of Driver Enable When the PWM is Running 72
Figure 3.19 ECM Driver Structure 73
Figure 3.20 ADC Functional Diagram 78
Figure 3.21 Ratiometric versus Absolute Measurement Inputs Configuration 79
Figure 3.22 Using AIN3 and AIN4 for External Temperature Measurement 80
Figure 3.23 Pipelined Measurement and Results Reading for the Driver Current 89
Figure 3.24 ADC Result Formatting Options 90
Figure 3.25 ADC Analog Inputs Protection and Open/Short Detection Circuitry 92
Figure 3.26 LIN PHY Block Diagram 95
Figure 3.27 LIN Wake-up Detection Depending on the Value of the LINWUMD Bit 96
Figure 3.28 SBC Register File Organization 99
Figure 3.29 Structure of SBC Trimming Data 102
Figure 4.1 Structure of FLASH Memory and Page Details 105
Figure 4.2 Example of RAMSPLIT Address Configuration 108
Figure 4.3 RESET and CLOCK between SBC and MCU 109
Figure 4.4 MCU Interrupt Logic Organization 110
Figure 4.5 Master SPI Block Diagram 114
Figure 4.6 SW-LIN Block Diagram 117
Figure 4.7 Structure of LIN Frame 118
Figure 4.8 Block Diagram of the LIN Data Unit 118
Figure 4.9 Frame Format of Each LIN Field (PID, DATA, Checksum) and RX Sample Position 119
Figure 4.10 RX Control and Status Signal Waveforms 120
Figure 4.11 Waveforms of the TX and RX Control and Status Signals 121
Figure 6.1 ZAMC4100 Application Circuit for Automotive Mirror Control 146
Figure 7.1 ZAMC4100 Multi-chip-Module Assembly 148
Figure 7.2 Pin Layout for ZAMC4100 PQFN65 – Top View 148
Figure 8.1 ZAMC4100 Package Dimensions – Plastic QFN64 152

© 2019 Renesas Electronics Corporation

List of Tables

Table 1.1 Absolute Maximum Ratings 14
Table 1.2 Thermal Specifications 14
Table 1.3 ESD Protection 15
Table 1.4 Supply Voltages and Currents 15
Table 1.5 System Resets and Supply Monitoring 16
Table 1.6 GPIO Electrical Characteristics 17
Table 1.7 LIN Electrical Parameters 17
Table 1.8 Drivers Static Parameters 18
Table 1.9 EC Driver Static Parameters 19
Table 1.10 Drivers Dynamic Parameters 20
Table 1.11 Internal Die Temperature Sensor 21
Table 1.12 Internal Reference Sources 21
Table 1.13 ADC and Analog Input Pins Static Parameters 22
Table 1.14 Timing Parameters: Low-Power Oscillator and High-Frequency Oscillator 22
Table 1.15 ADC Timing Parameters 23
Table 1.16 ZAMC4100 SPI Bus Timing Parameters 24
Table 1.17 System Power-up and Reset Timing Parameters 25
Table 1.18 Flash Timing Parameters 26
Table 3.1 RSTSTAT Register Bits Mapping 37
Table 3.2 RSTSTAT Register Bits Description 37
Table 3.3 WDTCONF Register Bits Mapping 40
Table 3.4 WDTCONF Register Bits Description 41
Table 3.5 SMDCTRL Register Bits Mapping 47
Table 3.6 SMDCTRL Register Bits Description 48
Table 3.7 IRQCTRL Register Bits Mapping 51
Table 3.8 IRQCTRL Register Bits Description 52
Table 3.9 IRQSTAT Register Bits Mapping 53
Table 3.10 IRQSTAT Register Bits Description 53
Table 3.11 SPI Lines Description 55
Table 3.12 SPI Slave Status Bits Descriptions (SSB) 58
Table 3.13 HBDCTRL Register Bits Mapping 60
Table 3.14 HBDCTRL Bits Description 60
Table 3.15 HBDSTAT Register Bits Mapping 61
Table 3.16 HBDSTAT Register Bit Descriptions 61
Table 3.17 HBDDIAG Register Bits Mapping 63
Table 3.18 HBDDIAG Register Bits Description 63
Table 3.19 Half-Bridge Diagnostic Bit Descriptions 64

© 2019 Renesas Electronics Corporation

Table 3.20 HSDCTRL Register Bits Mapping 66
Table 3.21 HSDCTRL Register Bits Description 66
Table 3.22 HSDSTAT Register Bits Mapping 67
Table 3.23 HSDSTAT Register Bits Description 67
Table 3.24 High-Side Diagnostic Bits Description 67
Table 3.25 PWMDREN Register Bits Mapping 70
Table 3.26 PWMDREN Register Bits Description 70
Table 3.27 ECMCTRL Register Bits Mapping 75
Table 3.28 ECMCTRL Register Bits Description 75
Table 3.29 ECMDIAG Register Bits Mapping 76
Table 3.30 ECMDIAG Register Bits Description 76
Table 3.31 ECMOSF Diagnostic Bit Meaning 77
Table 3.32 AINCONF Register Bits Mapping 80
Table 3.33 AINCONF Register Bits Description 81
Table 3.34 ADCCTRL Register Bits Mapping 82
Table 3.35 ADCCTRL Register Bits Description 82
Table 3.36 Input MUX Control Bits Description 83
Table 3.37 ADCCONF Register Bits Mapping 86
Table 3.38 ADCCONF Register Bits Description 86
Table 3.39 ADCSTAT Register Bits Mapping 87
Table 3.40 ADCSTAT Register Bits Description 87
Table 3.41 ADC Sample-Rate Configuration 88
Table 3.42 ADCCMPH and ADCCMPL Bits Description 91
Table 3.43 AINDIAG Register Bits Mapping 93
Table 3.44 AINDIAG Register Bits Description 93
Table 3.45 External ADC Inputs Diagnostic Bits Description 94
Table 3.46 LINCTRL Registers Bit Mapping 97
Table 3.47 LINCTRL Registers Bits Description 98
Table 3.48 LINSTAT Register Bits Mapping 98
Table 3.49 LINSTAT Register Bits Mapping 98
Table 3.50 SBC Registers Address Map 100
Table 4.1 MCU Memory Map 104
Table 4.2 INFO Page 0 Structure 107
Table 4.3 MCU Peripheral Interrupts 111
Table 4.4 Configuration of Timer Trigger Behavior 113
Table 4.5 Register [0x4000_0000] SYS_CLKCFG 123
Table 4.6 Register [0x4000_0004] SYS_MEMPORTCFG 123
Table 4.7 Register [0x4000_0008] SYS_MEMINFO 124

© 2019 Renesas Electronics Corporation

Table 4.8 Register [0x4000_000C] SYS_RSTSTAT 124
Table 4.9 Register [0x4000_1400] GPIO_DIR 125
Table 4.10 Register [0x4000_1404] GPIO_IN 125
Table 4.11 Register [0x4000_1408] GPIO_OUT 126
Table 4.12 Register [0x4000_140C] GPIO_SETCLR 126
Table 4.13 Register [0x4000_1410] GPIO_IRQSTAT 126
Table 4.14 Register [0x4000_1414] GPIO_IRQEN 127
Table 4.15 Register [0x4000_1418] GPIO_IRQEDGE 127
Table 4.16 Register [0x4000_141C] GPIO_TRIGEN 127
Table 4.17 Register [0x4000_1000] T32_CTRL 128
Table 4.18 Register [0x4000_1004] T32_TRIGSEL 129
Table 4.19 Register [0x4000_1008] T32_CNT 129
Table 4.20 Register [0x4000_100C] T32_REL 129
Table 4.21 Register [0x4000_1820] Z1_SPICFG 130
Table 4.22 Register [0x4000_1824] Z1_SPIDATA 130
Table 4.23 Register [0x4000_1828] Z1_SPICLKCFG 131
Table 4.24 Register [0x4000_182C] Z1_SPISTAT 131
Table 4.25 Register [0x4000_1800] Z1_LINCFG 132
Table 4.26 Register [0x4000_1804] Z1_LINSTAT 133
Table 4.27 Register [0x4000_1808] Z1_LINDATA 134
Table 4.28 Register [0x4000_180C] Z1_LINIRQEN 134
Table 4.29 Register [0x4000_1810] Z1_LINBAUDLOW 135
Table 4.30 Register [0x4000_1814] Z1_LINBAUDHIGH 135
Table 4.31 Register [0x4000_0800] FC_RAM_ADDR 136
Table 4.32 Register [0x4000_0804] FC_FLASH_ADDR 136
Table 4.33 Register [0x4000_080C] FC_EXE_CMD 136
Table 4.34 Register [0x4000_0808] FC_CMD_SIZE 137
Table 4.35 Register [0x4000_0810] FC_IRQ_EN 137
Table 4.36 Register [0x4000_0814] FC_STAT_CORE 138
Table 4.37 Register [0x4000_0818] FC_STAT_PROG 139
Table 4.38 Register [0x4000_081C] FC_STAT_DATA 139
Table 6.1 Parameters for External Components in Typical Automotive Mirror Control Application 147
Table 7.1 Pin Description 149

1 IC Characteristics

1.1. Absolute Maximum Ratings

Stresses above those listed in this clause may cause immediate and permanent device failure. It is not implied that more than one of these conditions can be applied simultaneously. Operation outside the operating ranges for extended periods may affect device reliability. Total cumulative dwell time above the maximum operating rating for power supply or temperature must be less than 100 hours.

Table 1.1 Absolute Maximum Ratings

Parameter	Symbol	Value	UNIT
Supply Voltage			
System supply voltage	V_{VDDE}	-0.3 to 28	V
System supply voltage under transient conditions	V_{VDDE_max}	-0.3 to 40	V
Input Pin Voltage			
AIN1-4	$V_{IN(ANALOG)}$	-0.3 to 5.5	V
GPIO0-7	$V_{IN(MCU)}$	-0.3 to 3.6	V
LIN Bus Voltage			
Bus voltage	$V_{BUS(SS)}$	-18 to 40	V
Transient input voltage (per *ISO7637 Specification*)	$V_{BUS(PK)}$	-150 to 100	V
GPIO Supply current			
Total GPIO sink/source current (referenced to 3.3V)	I_{GPIO}	5	mA

Table 1.2 Thermal Specifications

Parameter	Symbol	Value	UNIT
Thermal Ratings			
Operating ambient temperature	T_A	-40 to 85	˚C
Operating junction temperature [1), 2)]	T_J	-40 to 125	˚C
Storage temperature	T_{STG}	-40 to 150	˚C
Thermal Characteristics			
Thermal resistance junction to case [3),4)]	Θ_{JC}	0.4	˚C/W
Thermal resistance junction to ambient [3),4)]	Θ_{JA}	21	˚C/W

1) The limiting factor is the operating temperature of the FLASH memory within the MCU where the data retention is guaranteed for temperatures up to 85˚C. In order to keep the MCU temperature below 85˚C when the SBC is running at maximum temperature, the application should provide a sufficient heat sink. For more information, contact IDT technical support.

2) The device will automatically shut down the drivers at 125°C.

3) IDT uses test boards in conformance with the *JESD51-7 (JEDEC)* for thermal impedance measurement.

4) This value applies for the JEDEC board profile and specific environment only for package performance comparison. Care should be taken when applying in the actual application where the printed circuit board (PCB) profile and environment are different. Contact IDT technical support for assistance.

© 2019 Renesas Electronics Corporation

Table 1.3 ESD Protection

Test Method	Test Setup Standard	Guaranteed ESD/Latch-up Robustness on IC Level
HBM (Human Body Model)	MIL 883C, Method 3015.7 or EIA/JESD22A114-A	± 2kV (all pins) with 100pF and 1.5kΩ
CDM (Charged Device Model)	AEC – Q100 – 011 Rev. B	± 500V (all pins)
System Level (Human Body Discharge)	IEC 61000-4-2	± 6kV for LIN(1) and VBAT(2)
Latch-up	EIA/JESD78A	100mA@25˚C

(1) External circuitry is required for meeting the system-level ESD specification for the LIN interface.

(2) VBAT is the common supply terminal on the application PCB where all ZAMC4100 supply pins are connected.

1.2. Electrical Parameters

Note: Parameters are specified for junction temperature range -40°C to 125°C unless otherwise stated.

1.2.1. Supply and Supply Monitoring

Table 1.4 Supply Voltages and Currents

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage Range (voltage drop in external circuitry not included)					
Nominal operating voltage	V_{SUP}	6	-	18	V
LIN operating voltage	V_{SUP_LIN}	7	-	18	V
Supply Current Range					
NORMAL Mode	I_{RUN}	-	14	-	mA
STANDBY Mode	I_{STBY}	-	10	-	mA
SLEEP Mode (measured at T_J=27°C; V_{SUP}=13V through all VSS pins)	I_{SLEEP}	-	-	80	µA
Switchable Analog Supply Voltage (to S_P pin)					
Voltage range (S_P pin unloaded)	V_{SP}	4.5	5	5.5	V
Voltage range (S_P pin loaded with 5mA)	V_{SP_5mA}	3.8	4.5	5.2	V
Nominal output current	I_{SP}	0	-	5	mA
Current limitation in case of short to GND	I_{SP_OC}	6	-	9	mA
GPIO Supply Voltage					
Voltage range	V_{GPIO}	2.97	3.3	3.63	V

Charge Pump					
Charge pump output voltage (with respect to VDDE1; drivers disabled)	V_{CP}	4.5	5.0	5.5	V
Charge pump output current (through CAP_CP pin)	I_{CAP_CP}	-	-	10	µA
External Capacitor Pins Voltage					
REG5V0 capacitor pin CAP_5V0	V_{CAP_5V0}	4.5	5.0	5.5	V
REG3V3 capacitor pin CAP_3V3	V_{CAP_3V3}	2.97	3.3	3.63	V
REG1V8 capacitor pin CAP_1V8 (on SBC side)	V_{CAP_1V8}	1.62	1.8	1.98	V
REG1V8 capacitor pin CAP_MCU (on MCU side)	V_{CAP_MCU}	1.62	1.8	1.98	V
Charge pump capacitor pin CAP_CP (with respect to VDDE1 pin)	V_{CAP_CP}	4.3	-	5.5	V

Table 1.5 System Resets and Supply Monitoring

Parameter	Symbol	Min.	Typ.	Max.	Unit
Power-On Reset					
Threshold [1)]	POR_{TH}	4.2	4.5	4.8	V
Hysteresis	POR_{HYST}	0.2	0.3	0.4	V
Under-Voltage Detection					
Threshold [1)]	UVD_{TH}	5.7	6.0	6.3	V
Hysteresis	UVD_{HYST}	0.05	0.2	0.3	V
Over-Voltage Detection					
Threshold [1)]	OVD_{TH}	17.7	18.6	19.5	V
Hysteresis	OVD_{HYST}	0.3	0.5	0.7	V
Over-Temperature Drivers Shut-Down					
Threshold [2)]	OVT_{TH}	-	125	-	°C
Hysteresis [2)]	OVT_{HYST}	-	5	-	°C

1) These parameters are defined with respect to the potential at the VDDE pin.

2) Junction temperature of the SBC chip. Parameter guaranteed by process monitoring but not tested during production.

© 2019 Renesas Electronics Corporation

1.2.2. Digital Interface (GPIOs)

Table 1.6 GPIO Electrical Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Input low voltage	V_{IL}	-0.3	-	0.8	V
Input high voltage	V_{IH}	2	-	3.6	V
Schmitt Trigger Low to High Threshold Point	V_{T+}	1.4	1.5	1.6	V
Schmitt Trigger High to Low Threshold Point	V_{T-}	1.1	1.2	1.3	V
Input leakage current (at V_I =3.3V or 0V)	I_I	-1	-	1	µA
Tri-state output leakage current (at V_I =3.3V or 0V)	I_{OZ}	-1	-	1	µA
Output low voltage	V_{OL}	-	-	0.4	V
Output high voltage	V_{OH}	2.4	-	-	V
Single GPIO source current	I_{SOURCE}	-	-	5	mA
Single GPIO sink current	I_{SINK}	-	-	5	mA

1.2.3. LIN Physical Layer

Note: LIN parameters are specified for an ambient temperature range from -40°C to 125°C. Although the internal over-temperature protection might be activated due to device self-heating, LIN communication and MCU operation remain unaffected.

Table 1.7 LIN Electrical Parameters

Note: See important notes at end of table.

Parameter	Symbol	LIN Spec 2.2 Reference	Min	Typ.	Max	Unit
Current limitation for driver dominant state	I_{BUS_LIM}	Parameter 12	40	-	200	mA
Input leakage current, dominant state, driver off	$I_{BUS\_PAS\_dom}$	Parameter 13	-1	-	-	mA
Input leakage current, recessive state, driver off	$I_{BUS\_PAS\_rec}$	Parameter 14	-	-	20	µA
Control unit disconnected from ground [1)]	$I_{BUS\_NO\_GND}$	Parameter 15	-1	-	1	mA
V_{BAT} supply disconnected	$I_{BUS\_NO\_BAT}$	Parameter 16	-	-	100	µA
Receiver dominant state (V_{SUP} > 7V)	V_{BUSdom}	Parameter 17	-	-	0.4	V_{SUP}
Receiver recessive state (V_{SUP} > 7V)	V_{BUSrec}	Parameter 18	0.6	-	-	V_{SUP}
Center of receiver threshold	V_{BUS_CNT}	Parameter 19	0.475	0.5	0.525	V_{SUP}
Receiver hysteresis voltage	V_{HYS}	Parameter 20	-	-	0.175	V_{SUP}
Voltage drop at serial diodes [2)]	$V_{SerDiode}$	Parameter 21	0.4	0.7	1	V
Battery shift[(2)]	V_{SHIFT_BAT}	Parameter 22	-	-	0.115	V_{BAT}
Ground shift [2)]	V_{BUS_GND}	Parameter 23	-	-	0.115	V_{BAT}

© 2019 Renesas Electronics Corporation

Parameter	Symbol	LIN Spec 2.2 Reference	Min	Typ.	Max	Unit
Difference between battery shift and ground shift [2]	$V_{SHIFT\_Difference}$	Parameter 24	0	-	8	%
LIN pull-up resistor	R_{SLAVE}	Parameter 26	20	40	60	kΩ
Duty cycle 1	D1	Parameter 27	0.396	-	-	-
Duty cycle 2	D2	Parameter 28	-	-	0.581	-
Duty cycle 3	D3	Parameter 29	0.417	-	-	-
Duty cycle 4	D4	Parameter 30	-	-	0.590	-
Receiver propagation delay	t_{RX_pdr}	Parameter 31	-	-	6	µs
Receiver propagation delay symmetry between rising and falling edge	t_{RX_sym}	Parameter 32	-2	-	2	µs
Capacitance of slave node [2]	C_{SLAVE}	Parameter 23	-	-	250	pF
LIN pin capacitance [1]	C_{LIN}	-	-	-	30	pF
Over-current shutdown threshold	I_{OC_LIN}	-	45	-	175	mA
LIN wake-up filter time	t_{LINWU}	-	100	125	150	µs
TxD dominant time out time	t_{DOM}	-	11	-	-	ms
LIN fast mode data rate (Cbus = 1nF, Rbus = 1kΩ) [1]	LB_{FAST}	-	-	-	115	kbps

1) Parameter guaranteed by process monitoring but not tested during production.

2) Standard LIN network parameter which specifies operating condition for the IC.

1.2.4. Output Drivers

Table 1.8 Drivers Static Parameters

Parameter	Symbol	Min.	Typ.	Max.	Unit
HS1 (High-Side Switch)					
Switch-on resistance (at 27°C; V_{SUP}=13.5V; I_{HS1}=1A)	R_{DSON_HS1}	110	170	230	mΩ
Over-current shut down	I_{OC_HS1}	5.5	6.1	6.7	A
Leakage current	$I_{LEAKAGE\_HS1}$	-	<0.5	15	µA
Current sense ratio	$CSR_{\_HS1}$	650000	695000	740000	-

© 2019 Renesas Electronics Corporation

HS2-4 (High-Side Switches)					
Switch-on resistance (at 27°C; V_{SUP}=13.5V; $I_{HS2,3,4}$=0.3A)	$R_{DSON\_HS2,3,4}$	500	700	900	mΩ
Over-current shut down	$I_{OC\_HS2,3,4}$	0.35	0.45	0.55	A
Leakage current	$I_{LEAKAGE\_HS2,3,4}$	-	<0.5	5	µA
Current sense ratio	$CSR_{\_HS2,3,4}$	47000	52500	58000	-
HB1,2 (Half-Bridges) [1)]					
Switch-on resistance (at 27°C; V_{SUP}=13.5V; $I_{HB1,2}$=1A)	$R_{DSON\_HB1,2}$	150	240	330	mΩ
Over-current shut down	$I_{OC\_HB1,2}$	3.0	3.5	4.0	A
Leakage current	$I_{LEAKAGE\_HB1,2}$	-	<0.5	10	µA
Current sense ratio	$CSR_{\_HB1,2}$	350000	385000	420000	-
HB3,4 (Half-Bridges) [1)]					
Switch-on resistance (at 27°C; V_{SUP}=13.5V; $I_{HB3,4}$=0.5A)	$R_{DSON\_HB3,4}$	700	900	1100	mΩ
Over-current shut down	$I_{OC\_HB3,4}$	0.6	0.7	0.8	A
Leakage current	$I_{LEAKAGE\_HB3,4}$	-	<0.5	7	µA
Current sense ratio	$CSR_{\_HB3,4}$	70000	76500	83000	-
Open/Short Diagnostic Parameters (all HB and HS Drivers)					
Diagnostic current source/sink	$I_{HB1..4\_CS}$, $I_{HS1..4\_CS}$	80	100	120	µA
Open/short flag threshold level	V_{OSF_DR}	0.4	-	1.5	V

1) Parameters specified for both halves: high-side and low-side.

Table 1.9 EC Driver Static Parameters

Parameter	Symbol	Min.	Typ.	Max.	Unit
High-Side					
Over-current shut down	$I_{OC\_EC\_RES}$	0.4	-	-	A
Leakage current	$I_{LEAKAGE\_EC\_RES}$	-	0.1	7	µA
Low-Side					
Switch-on resistance (at 27°C ; V_{SUP}=13.5V; I_{EC_RES}=0.1A)	$R_{DSON\_EC\_M}$	500	1000	1500	mΩ
Over-current shut down	$I_{OC\_EC\_M}$	0.4	-	-	A
Leakage current	$I_{LEAGAGE\_EC\_M}$	-	5	10	µA

© 2019 Renesas Electronics Corporation

Parameter	Symbol	Min.	Typ.	Max.	Unit
EC Driver Output Voltage					
Nominal output voltage range	V_{EC}	0	-	1.38	V
Maximum output voltage	V_{ECMAX}	1.311	1.38	1.449	V
Output voltage tolerance	dV_{EC}	-5	-	5	%
DAC resolution	EC_{DAC_RES}	-	6	-	bit
Open/Short Diagnostic Parameters (for EC_M and EC_RES Pins)					
Diagnostic current source/sink	I_{ECM_CS}, I_{ECRES_CS}	50	120	190	µA
Open/short flag threshold level	V_{OSF_ECM}	0.6	-	0.9	V

Table 1.10 Drivers Dynamic Parameters

Note: See important notes at end of table.

Parameter	Symbol	Min.	Typ.	Max.	Unit
HS1 (High-Side Driver)					
Propagation delay [1), 2)]	t_{D_HS1}	5	11	17	µs
Over-current shut down integration time [3)]	t_{OC_HS1}	-	20	-	µs
Slew rate (I_{HS1}=5A) [2), 4)]	SL_{HS1}	0.3	0.8	1.1	V/µs
HS2-4 (High-Side Drivers)					
Propagation delay [1), 2)]	$t_{D\_HS2,3,4}$	3	7	11	µs
Over-current shut down integration time [3)]	$t_{OC\_HS2,3,4}$	-	20	-	µs
Slew rate ($I_{HS2,3,4}$=0.3A) [2), 4)]	$SL_{HS2,3,4}$	0.6	1.3	2.0	V/µs
HB1,2 (Half-Bridges)					
Propagation delay [1), 2)]	$t_{D\_HB1,2}$	5	11	17	µs
Over-current shut down integration time [3)]	$t_{OC\_HB1,2}$	-	20	-	µs
Slew rate ($I_{HB1,2}$=2.7A) [2), 4)]	$SL_{HB1,2}$	0.6	1.1	1.6	V/µs
HB3,4 (Half-Bridges)					
Propagation delay [1), 2)]	$t_{D\_HB3,4}$	4	12.5	21	µs
Over-current shut down integration time [3)]	$t_{OC\_HB3,4}$	-	20	-	µs
Slew rate ($I_{HB3,4}$=0.6A) [2), 4)]	$SL_{HB3,4}$	0.6	1.1	1.6	V/µs

© 2019 Renesas Electronics Corporation

Parameter	Symbol	Min.	Typ.	Max.	Unit
EC Driver					
Transition time [5), 2)]	t_{S_EC}	4	5	10	µs

1) The time between driver enable/disable and reaching 10% of the reaction at the output.
2) Parameter guaranteed by process monitoring but not tested during production.
3) Accumulative time for over-current event. After this period, the driver is automatically switched off (see sections 3.8.2 and 3.9.2).
4) Between 10% and 90% of the output voltage swing.
5) After DAC code switching from 0 to 31 for the range of 10% to 90% at the output reaction.

1.2.5. Internal Reference Sources and Sensors

Table 1.11 Internal Die Temperature Sensor

Parameter	Symbol	Min.	Typ.	Max.	Unit
Voltage/Temperature Slope [1)]	S_{VT}	1.76	1.80	1.84	mV/°C
Output Voltage (at 27°C)	V_{t27}	0.52	0.54	0.56	V

1) Parameter guaranteed by process monitoring but not tested during production.

Table 1.12 Internal Reference Sources

Parameter	Symbol	Min.	Typ.	Max.	Unit
Reference voltage	V_{REF}	-	2.4	-	V
Reference voltage accuracy [1)]	dV_{REF}	-2	-	2	%
Reference current (at 27°C after trimming)	I_{REF}	9.7	10	10.3	µA
Reference current temperature drift [1)]	dI_{REF}	-3	-	3	%

1) Parameter guaranteed by process monitoring but not tested during production.

© 2019 Renesas Electronics Corporation

1.2.6. Analog-to-Digital Converter

Table 1.13 ADC and Analog Input Pins Static Parameters

Parameter	Symbol	Min.	Typ.	Max.	Unit
ADC Parameters					
Input voltage (AIN1 to AIN4) ratiometric measurement [1]	V_{IN_RM}	V_{SN}	-	V_{SP}	V
Input voltage (AIN1 to AIN4) absolute voltage measurement [2]	V_{IN_AM}	0	-	V_{REF}	V
Resolution	B_{ADC}	-	10	-	Bits
Integral non-linearity	INL	-3	-	+3	LSB
Differential non-linearity	DNL	-2	-	+2	LSB
Offset error	OFE	-3	-	+3	LSB
Gain error [3]	GE	-0.2	-	0.2	% FSR
AIN1 to AIN4, S_N, and S_P Pins Parameters					
Switch resistance between S_N and VSSA pins	R_{DSON_SN}	10	-	40	Ω
Equivalent input capacitance (AIN1 to AIN4 pins)	C_{IN}	-	-	15	pF
Equivalent input series resistance (AIN1 to AIN4)	R_{S_IN}	-	-	1.5	kΩ
Diagnostic current source/sink (S_P and AIN1 to AIN4)	I_{AIN_CS}, I_{SP_CS}	8	10	12	µA
Open/Short flag threshold level (S_P and AIN1 to AIN4)	V_{OSF_AIN}	2.3	-	3.3	V

1) V_{SN} is the potential of the S_N pin. Considered to be 0V in the application.
2) The minimum specification for V_{IN_AM} is 0 with respect to the VSSA pin.
3) As a percent of the input full-scale range (FSR).

1.3. Timing Parameters

Note: Parameters are specified for a junction temperature range from -40°C to 125°C unless otherwise stated.

1.3.1. Oscillators (OSCL and OSCH)

Table 1.14 Timing Parameters: Low-Power Oscillator and High-Frequency Oscillator

Parameter	Symbol	Min.	Typ.	Max.	Unit
OSCL (Low Power Oscillator)					
Base frequency (at 27°C, after trimming)	F_{OSCL}	122	125	128	kHz
Frequency temperature drift	F_{DRIFT_OSCL}	-10	-	10	%
OSCH (High Frequency Oscillator)					
Base frequency (at 27°C, after trimming)	F_{OSCH}	19.7	20	20.3	MHz
Frequency temperature drift	F_{DRIFT_OSCH}	-5	-	5	%

© 2019 Renesas Electronics Corporation

1.3.2. Analog-to-Digital Converter

Table 1.15 ADC Timing Parameters

Note: See Figure 1.1 for parameter definitions. All timing parameters below are calculated for the typical value of the OSCH frequency.

Parameter	Symbol	Min.	Typ.	Max.	Unit
ADC maximum sample rate [1), 2)]	SR_{MAX}	-	250	-	Ksps
ADC continuous conversion phase	t_{CCNV}	4	-	512	µs
ADC continuous conversion phase (external temperature mode)	t_{CCNV_TM}	32	-	512	µs
ADC single conversion phase	t_{SCNV}	14	-	522	µs
ADC single conversion phase (external temperature mode) [3)]	t_{SCNV_TM}	-	42	-	µs
ADC settling time [4)]	t_{ADC_STL}	-	10	-	µs
ADC sampling time [5)]	t_{ADC_SMP}	1	-	509	µs
ADC sampling time (external temperature mode) [6)]	$t_{ADC\_SMP\_TM}$	-	29	-	µs
ADC conversion time	t_{ADC_CNV}	-	3	-	µs
ADC idle time [3), 6), 7)]	t_{ADC_IDLE}	0	-	480	µs

1) The absolute maximum IDT target for ADC sample rate is 250ksps. IDT reserves the right to change the maximum ADC sample rate if driven by further design considerations.

2) The maximum possible sample rate for external temperature conversion is $SR_{MAX/8}$.

3) The ADC idle time is 0 for single external temperature conversion and continuous conversion at maximum SR.

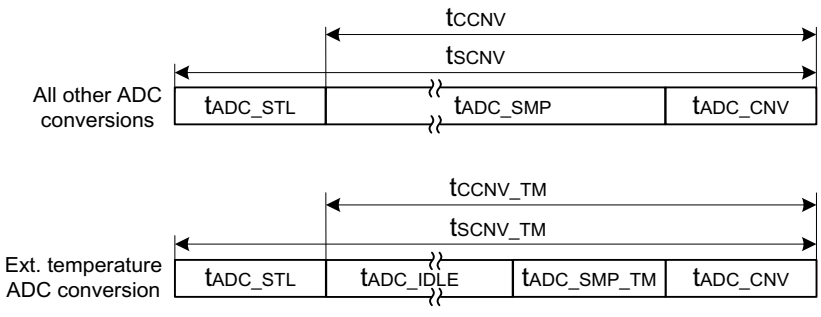
4) Added to each single conversion and to the first conversion in continuous mode.

5) The parameter t_{ADC_SMP} is calculated using the formula: $t_{ADC\_SMP} = \left(2^n / SR_{MAX}\right) - t_{ADC\_CNV}$, where n = [0:7] and n is the decimal equivalent of the value configured in bit field ADCSR[2:0] in the ADCCONF register (see Table 3.37).

6) For external temperature measurement, the ADC sample time is fixed. Changing ADCSR[2:0] changes the ADC idle time.

7) The parameter t_{ADC_IDLE} is calculated using the formula: $t_{ADC\_IDLE} = \left(2^n / SR_{MAX}\right) - t_{ADC\_SMP\_TM} - t_{ADC\_CNV}$, where n = [3:7] and n is the decimal equivalent of the value configured in bits ADCSR[2:0]. Important: For n = [0:2], t_{ADC_IDLE} = 0.

Figure 1.1 ADC Timing Parameters Definition



1.3.3. SPI Bus Timing

Refer to Figure 1.2 and Table 1.16 for the definitions of the SPI timing parameters.

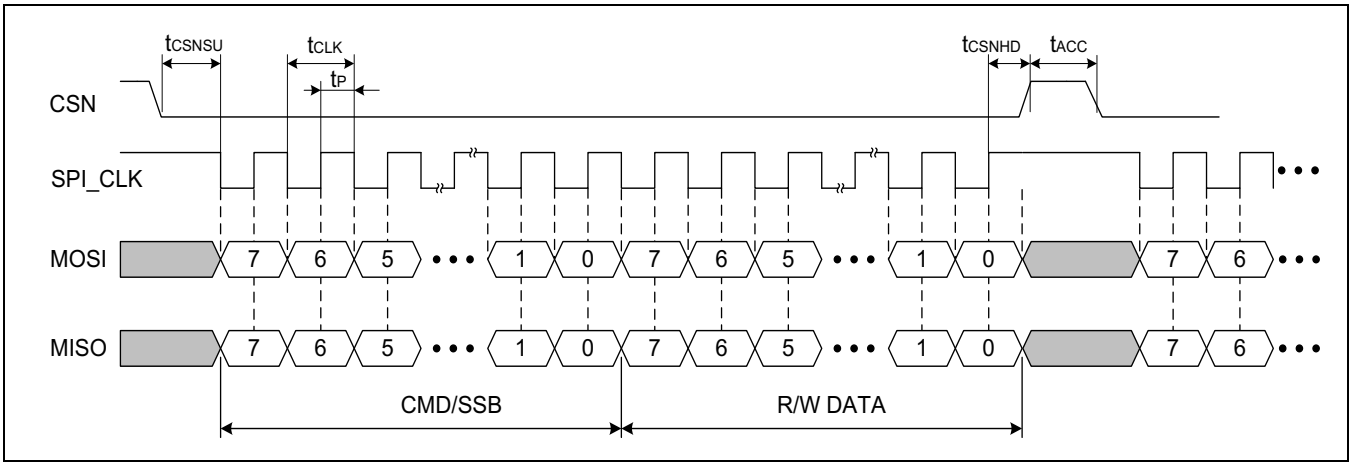
Table 1.16 ZAMC4100 SPI Bus Timing Parameters

Parameter	Symbol	Min.	Typ.	Max.	Unit
SPI clock period [1)]	t_{CLK}	0.3	-	6.4	µs
SPI pulse width	t_P	40% t_{CLK}	50% t_{CLK}	60% t_{CLK}	µs
Chip-select setup time [2)]	t_{CSNSU}	1.0	-	-	µs
Chip-select hold time [2)]	t_{CSNHD}	1.0	-	-	µs
SPI access time	t_{ACC}	1.0	2.0	-	µs

1) The maximum t_{CLK} value is determined by the maximum value that can be programmed in MCU register Z1_SPICLKCFG[7:2] (see Table 4.23).

2) The parameters t_{CSNSU} and t_{CSNHD} require at least 3 OSCH clock cycles. When oscillator OSCH is not trimmed, it runs at a lower frequency.

Figure 1.2 SPI Communication Timing Diagram



1.3.4. System Power-up Timing

Refer to Figure 1.3 and Table 1.17 for the definitions of the power-up and reset timing parameters.

Table 1.17 System Power-up and Reset Timing Parameters

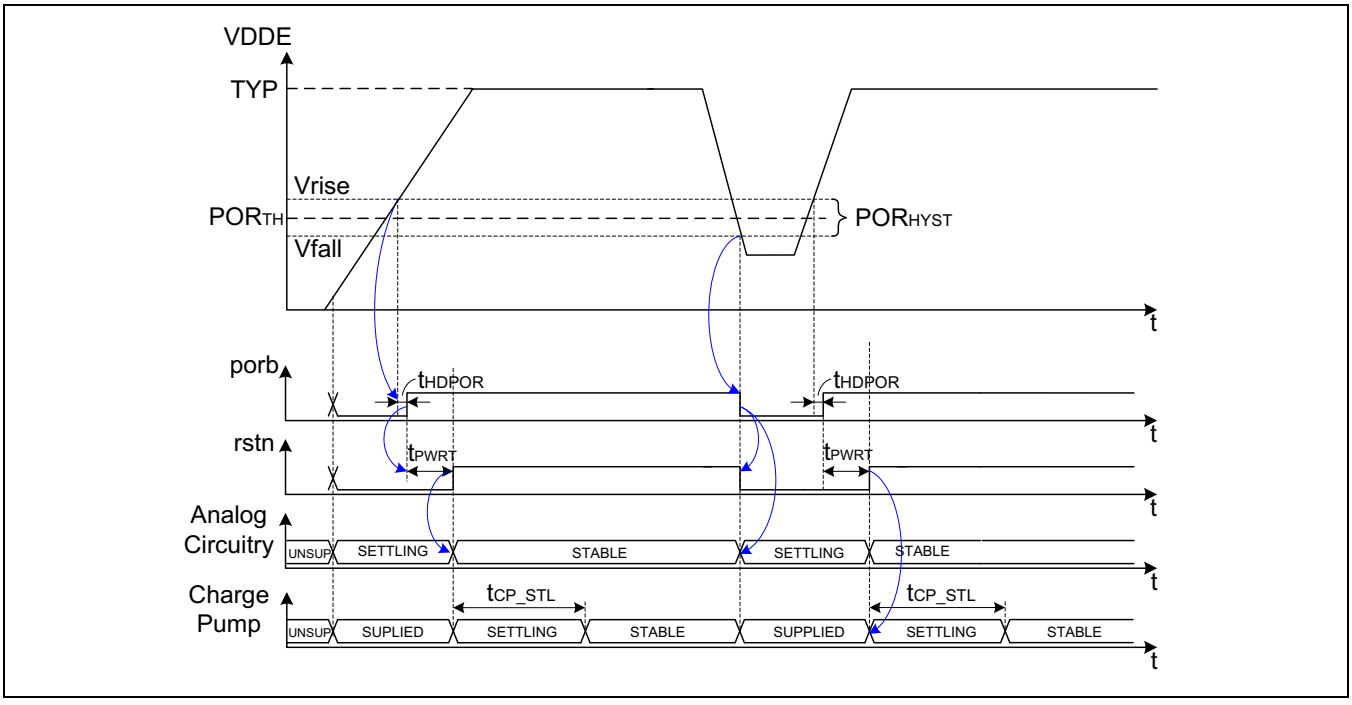
Parameter	Symbol	Min.	Typ.	Max.	Unit
Power-on-reset (POR) hold time	t_{HDPOR}	10	-	80	µs
Power-up timer delay	t_{PWRT}	0.32	0.78	1.24	ms
MCU supply check time [1)]	t_{MSC}	1.1	1.66	2.5	ms
Charge pump voltage settling time [2)]	t_{CP_STL}	-	-	2	ms
MCU reset pulse duration generated by the watchdog timer (WDT) [3)]	t_{MCU_RST}	-	8	-	µs
System reset pulse duration generated by WDT	t_{SYS_RST}	0.4	0.95	1.5	ms

1) Time window in which the valid flags for both MCU supplies are expected. For more information, refer to section 3.1.

2) The charge pump is considered as settled when the ZAMC4100 supply voltage is in the nominal operating range specified by parameter V_{SUP} (see Table 1.4).

3) Parameters calculated for the typical value of OSCL frequency.

Figure 1.3 ZAMC4100 System Power-Up Sequence



1.4. MCU Flash Memory Parameters

Table 1.18 Flash Timing Parameters

Parameter	Symbol	Min.	Typ.	Max.	Unit
Page erase time [1), 2)]	t_{PERASE}	-	17	-	ms
Mass erase time [1), 2)]	t_{MERASE}	-	17	-	ms
Single word program time [1), 2)]	t_{WPROG}	-	78	-	µs

1) Parameters given for the typical value of OSCH frequency.

2) Timing for flash controller commands execution. During that time the program execution is halted.

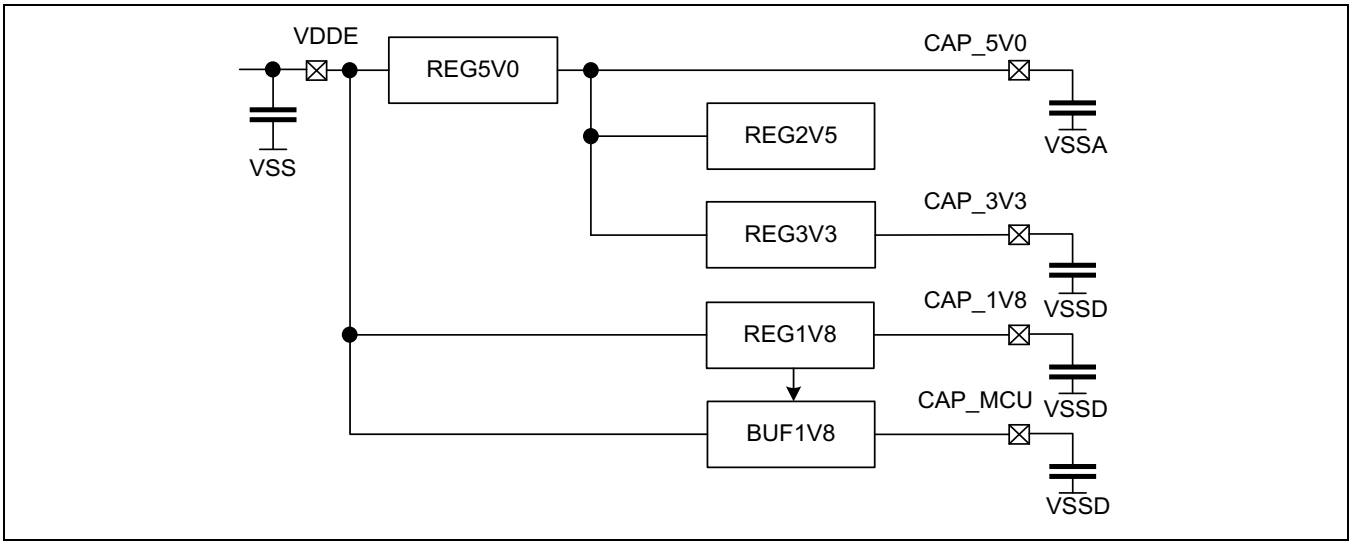
2 System Blocks Description

2.1. Power Supply

As illustrated in Figure 2.1, the internal supply voltages of the system are generated by four linear regulators and one buffer:

- **REG5V0** Primary analog supply voltage: 5.0V typical. REG5V0 is not switched-off in SLEEP Mode.
- **REG2V5** Secondary analog supply voltage: 2.5V typical. REG2V5 is switched-off in SLEEP Mode.
- **REG3V3** MCU periphery (GPIO, FLASH) digital supply voltage: 3.3V typical. REG3V3 is switched off in SLEEP Mode.
- **REG1V8** SBC digital supply voltage: 1.8V typical. This voltage is used as a reference for the MCU core supply. REG1V8 is not switched off in SLEEP Mode.
- **BUF1V8** Power supply for the MCU core referenced to the SBC digital supply voltage. BUF1V8 is switched off in SLEEP Mode.

Figure 2.1 ZAMC4100 Supply Structure



2.2. High-Frequency and Low-Frequency Oscillator

The ZAMC4100 system has two built-in oscillators: OSCH and OSCL. Oscillator OSCH generates the 20 MHz clock used directly in the MCU chip and divided by 5 in SBC chip. OSCL is a very low-power oscillator which provides 125kHz clock frequency used by LIN wake-up logic during SLEEP Mode and used as an autonomous clock for the watchdog timer. When SLEEP Mode is enabled, the SBC switches from OSCH to OSCL clock and vice versa when SLEEP Mode is exited.

2.3. Watchdog Timer

The SBC features an integrated watchdog timer (WDT), running on an autonomous 125kHz clock provided by OSCL oscillator. The WDT enables monitoring unresponsive states within the MCU and the SPI bus.

The WDT will generate configurable time-out intervals and when expired will reset the MCU. To avoid this, the MCU firmware should periodically reset the WDT via the SPI interface.

The WDT has a separate counter that counts the number of MCU resets (N = 4, 8 as configured in the WDTCONF register; see Table 3.3). When expired, it generates global reset for the whole system. This guarantees that all drivers are automatically disabled if abnormal MCU behavior occurs (e.g., frozen code).

2.4. Analog-to-Digital Converter

The SBC features a high-accuracy 10-bit successive approximation ADC with a front-end multiplexer (MUX) that selects one of the external or internal input signals for sampling and further processing by the MCU.

The ADC supports three types of external measurements, depending on the configuration setting in the firmware: ratiometric mode, absolute mode, or temperature measurement using internal current sources. For more details, refer to section 3.12.2.

Summary of possible measurements:

- Absolute voltage measurement with internal reference voltage
- Absolute current measurement with internal reference current
- Ratiometric measurement
- Measurements of internal input signals for the ADC:
  - Supply voltage; i.e., the voltage between the VDDE and VSSA pins
  - Internal temperature: the average temperature of the SBC die
  - Current through each half-bridge and high-side driver output (does not include EC mirror driver)
  - Actual output voltage of EC mirror driver (at EC_M pin)
  - Diagnostic measurements, which are used during the self-diagnostic mode
- External analog inputs AIN1-AIN4. Pins AIN3/TEMP and AIN4/TEMP feature switchable current sources allowing optional direct connection to external temperature sensors

© 2019 Renesas Electronics Corporation

2.5. Output Drivers

The ZAMC4100 features three types of drivers:

- **Half-bridge drivers** – Low-side and high-side switches are connected between the positive supply rail and the negative supply rail
- **High-side drivers** – High-side switches connected to the positive supply rail VDDE
- **EC mirror control driver** (ECM driver) – Output power stage, digital-to-analog converter (DAC) for controlling the output voltage level, and analog feedback circuit

Each driver provides the following functionalities:

- Low-resistance power switches optimized for high-current operation.
- PWM control available for all half-bridge and high-side drivers.
- The ECM driver is a special type of power driver designed for electrochromatic mirror control. For more details, refer to section 3.11.
- Integrated diagnostic and protection features.
- Detects high impedance output state.
- Except for the ECM, all switches have an integrated current measurement capability. If selected, these signals are digitized by the ADC. For the ECM driver, a dedicated internal channel is implemented for direct measurement of the actual output voltage (potential at EC_M pin).

All switches use an NMOS device for optimal low resistance channels. For driving the high-side switches, an integrated charge pump provides voltages higher than VDDE.

2.6. Driver's Power Capability and Package Temperature

One of the features of the ZAMC4100 is very low channel resistance; therefore, the package is less likely to overheat in performance of high current operations. However it is the responsibility of the application design engineer to properly evaluate thermal behavior in the application to avoid premature safety shutdown by the internal temperature protection circuitry.

The thermal behavior of the ZAMC4100 is very dependent on the PCB layout, encapsulate material, and heat sinking. Poor quality PCB layout resulting in high thermal resistance will likely cause over-heating, which will trigger an over-temperature event and prematurely shut down the driver. It is likely that there will be a delay before the package junction returns to a safe temperature level, depending on the rate of heat dispersion into the ambient environment.

The MCU could incorporate additional firmware referred to as Smart Power Management, depending on the application; for example for an automotive heater mirror, where demand for a high current level can be reduced by implementing a low-frequency PWM method when the ZAMC4100's package temperature exceeds a specified temperature level. This feature is designed to prolong the operation period and therefore enhanced reliability. For more information, contact IDT's technical support.

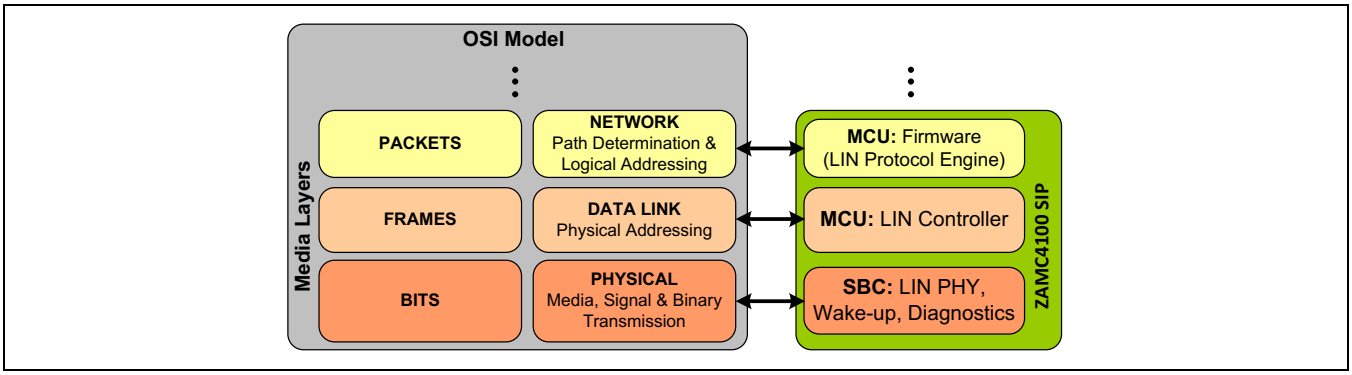
© 2019 Renesas Electronics Corporation

2.7. LIN Interface

The ZAMC4100 features a LIN slave interface with hardware resources covering the physical and data link communication layers (see Figure 2.2). The network layer must be implemented as part of the MCU firmware.

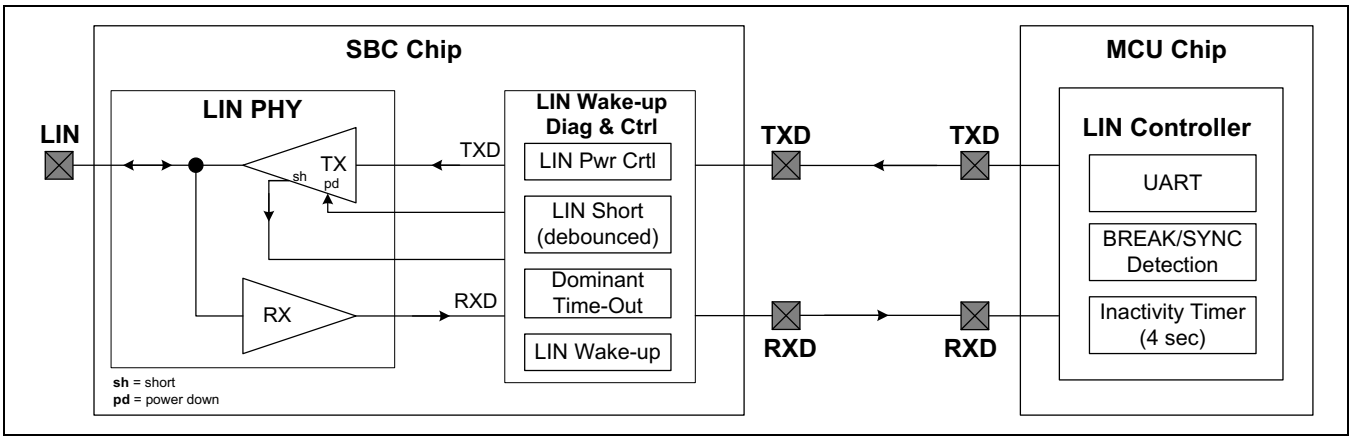
The LIN PHY is built into the SBC chip and is LIN2.0/2.1 compatible. It supports LIN wake-up detection, short-circuit detection/protection and dominant time-out detection/protection. All events are captured as interrupt flags in the SBC interrupt controller, and the MCU can access them via the SPI interface.

Figure 2.2 ZAMC4100 LIN Interface and Open Systems Interconnection (OSI) Layers Coverage



The block diagram in Figure 2.3 shows the ZAMC4100 LIN interface structure and the partitioning between the SBC and MCU chips. The hardware for covering the LIN data link layer is implemented in the MCU chip as a LIN controller. The MCU LIN controller features a UART module for the LIN frame generation, BREAK/SYNC detection fields, and LIN bus inactivity timer.

Figure 2.3 LIN Slave Node Implementation in the ZAMC4100



2.8. SPI Bus

The Serial Peripheral Interface (SPI) bus is used as a primary communication bus between the MCU and SBC peripherals, including the drivers. The SPI mastering is done by the MCU which configures and controls the SBC peripherals (output drivers, ADC, LIN, etc.). The SBC SPI slave supports byte-wise frame transfer with 8 data bits per frame. In order to secure the communication, the SBC SPI supports frame length validation.

2.9. Diagnostic Functionalities

ZAMC4100 has a number of built-in protection and diagnostic features that enable the MCU to perform system self-checks and to detect abnormal functionality. The main diagnostic functions supported by the ZAMC4100 are as follows:

- Open/short detection for each driver output
- System supply monitoring and under/over-voltage detection
- Open/short detection at ADC inputs
- ADC internal circuitry check
- LIN over-current and dominant time-out detection
- SBC-to-MCU SPI connectivity check

2.10. Microcontroller ARM® Cortex™ M0

The microcontroller unit (MCU) of the ZAMC4100 features a 32-bit ARM® Cortex™-M0 processor core with memory resources:

- 32 Kbytes program FLASH memory featuring
    - Minimum 10,000 erase/write cycles
    - 10 years data retention at 85°C
- 2 Kbytes SRAM

The MCU peripherals include the following:

- 8 configurable GPIO
- SW-LIN controller
- 32-bit timer module
- Master SPI
- Interrupt controller

© 2019 Renesas Electronics Corporation

2.10.1. MCU GPIOs

Each general-purpose input/output (GPIO) pin can be individually configured to operate as an input or output. If configured as an output, the value driven at the GPIO can be directly written or controlled via a set-clear register. Additionally, each GPIO pin can be enabled to be used as a trigger source or as interrupt source with a selectable edge.

2.10.2. MCU Timer Module

The MCU timer module provides event counting with a 32-bit resolution. It is capable of counting clock events in timer mode and events from a selectable external trigger signal in counter mode. The external trigger can be configured to operate on rising or falling edges as well as on a low or high level.

2.10.3. MCU SPI Master

The integrated SPI is a pure master interface that is used for communication with the SBC chip. The CSN line must be controlled by software. The SPI clock frequency is configurable by software with a maximum frequency of half of the system clock frequency.

2.10.4. MCU LIN Controller

The software-controlled LIN controller provides the logic to handle the LIN bus communication on the frame level (see Figure 2.3). It consists out of BREAK/SYNC detector, 4-second inactivity timer, and a UART data unit for frame receiving/transmitting. The LIN wake-up event is detected in the SBC chip and the interrupt on the IRQN line is generated. The packet-level LIN communication (network layer in Figure 2.2) is handled by the MCU firmware.

© 2019 Renesas Electronics Corporation

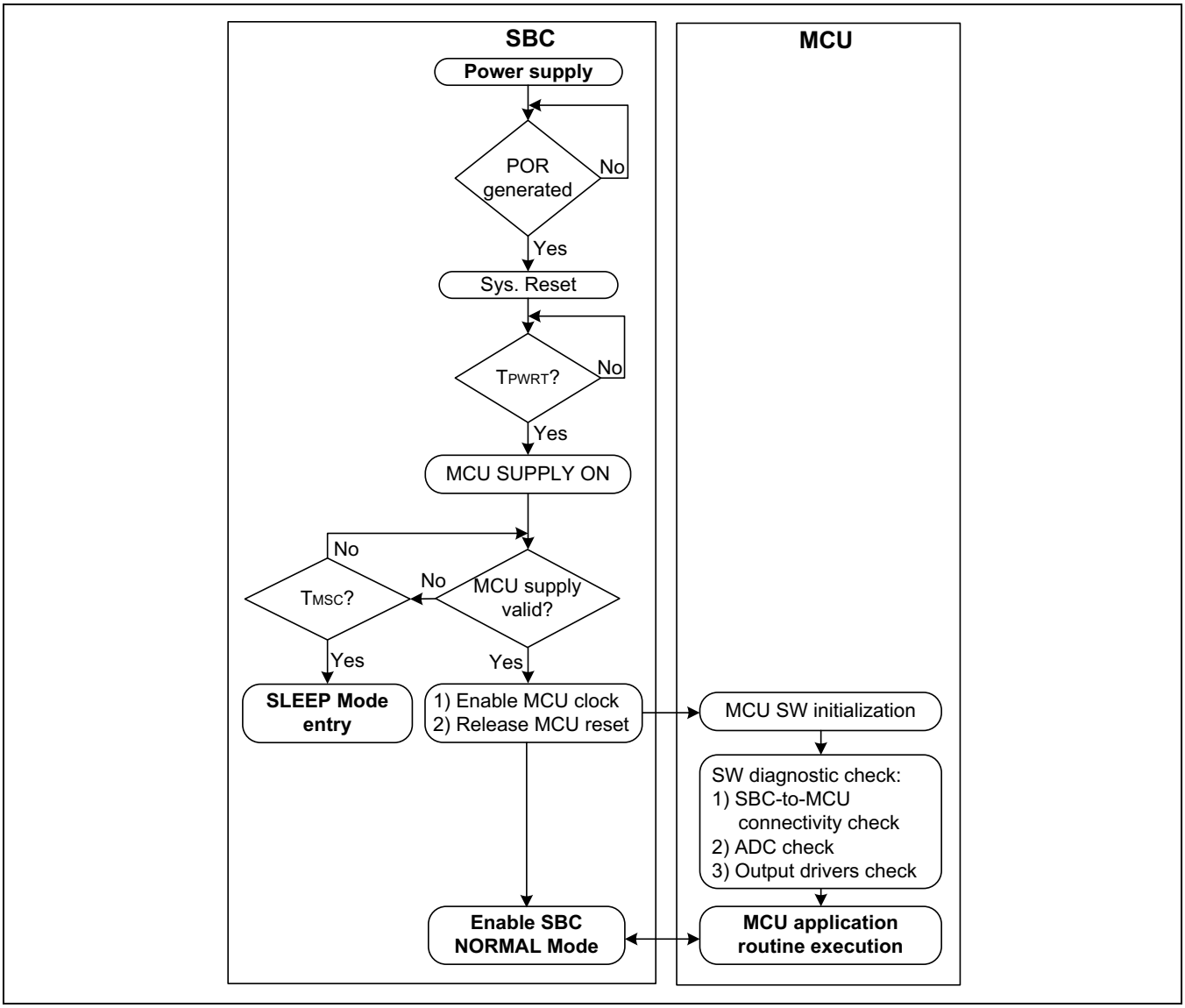
3 ZAMC4100 SBC Functional Description

This section describes the functionality of the system with respect to the modes of operation - SBC analog peripherals including the transistor drivers, analog-to-digital converter (ADC), LIN interface, interrupt sources, SPI interface, etc.

3.1. System Power-Up Sequence

After powering up, ZAMC4100 enters the NORMAL operation mode executing the sequence shown in Figure 3.1.

Figure 3.1 SBC Power-Up State Diagram



The power-up sequence of the whole system is handled by the SBC chip. When the voltage at the VDDE pin reaches the power-on-reset level (refer to Table 1.5), the system reset is generated. After power-on-reset, the power-up timer in the SBC starts counting a time t_{PWRT} needed for stabilizing the oscillators and the other analog circuitry (see Table 1.17). When the analog circuitry is stable, the SBC enables the MCU power supply and checks its validity. During the MCU supply validity check, the SBC checks that both supply voltages (BUF1V8 and REG3V3) needed for the MCU operation are in range.

When the supply is valid, the SBC enables the MCU clock and releases its reset. At this point, the SBC chip switches to the NORMAL Mode, and the MCU can start executing the initialization firmware, running the diagnostic checks, and switch to the application routine.

If the MCU supply is not valid, the SBC chip executes a time loop in which the internal supply-valid flags are continuously checked. If the time t_{MSC} expires and the MCU supply is still not in range, the SBC switches to SLEEP Mode. Switching to SLEEP Mode in the event of an MCU supply failure guarantees that the system stays passive with very low current consumption (less than 80µA). The ZAMC4100 can switch back to NORMAL Mode if a valid LIN wake-up is detected. In this case, the MCU supply is checked again and depending on the result, the system stays in NORMAL Mode or switches to SLEEP Mode.

The ZAMC4100 start-up timing parameters are specified in Table 1.17. Details regarding the ZAMC4100 SLEEP Mode are given in section 3.5.2.

3.2. Clock and Reset Sources of the ZAMC4100

All clocks and resets for ZAMC4100 system are generated in the SBC chip. The clock structure of the ZAMC4100 is shown in Figure 3.2, and the reset structure is shown in Figure 3.3.

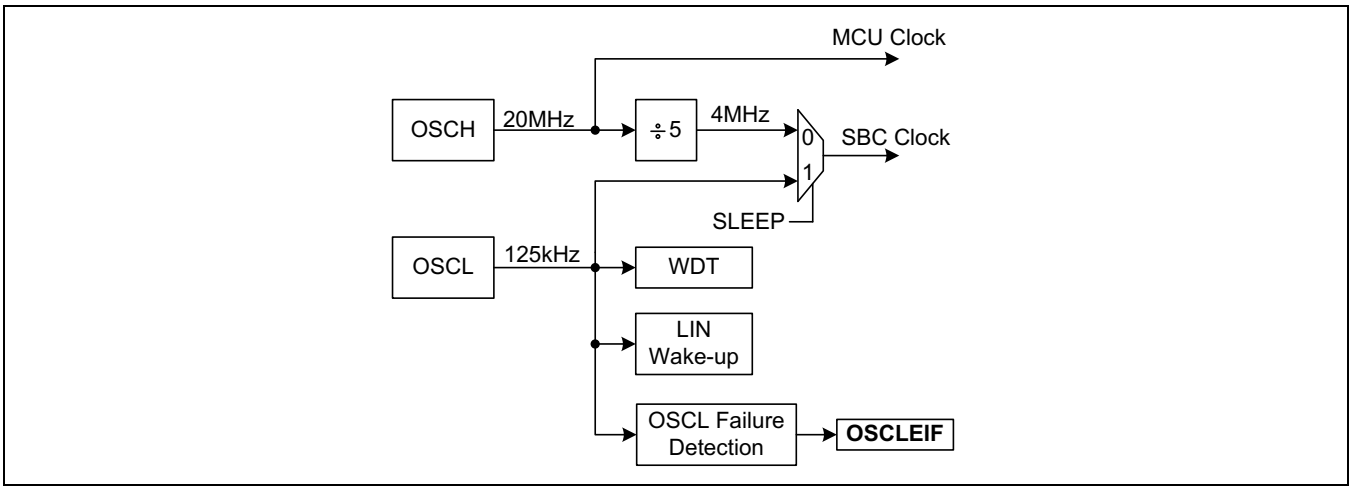
3.2.1. Clock Sources

OSCH is the main clock generator for the SBC and MCU. The output clock from OSCH goes directly to the MCU at 20MHz and to the divider for the SBC resulting in a 4MHz SBC clock.

In SLEEP Mode, the MUX sets the SBC clock to the125kHz OSCL clock. The main purpose of this clock in SLEEP Mode is to drive the LIN wake-up detection. Also the OSCL clock is used continuously by the WDT.

© 2019 Renesas Electronics Corporation

Figure 3.2 ZAMC4100 Clock Sources



OSCL Failure Detection

In order to increase the system robustness, the ZAMC4100 features a mechanism for failure detection in the OSCL clock generation. This ensures that the MCU is alerted by an interrupt in the event of the watchdog timer not functioning (caused by an OSCL clock stop).

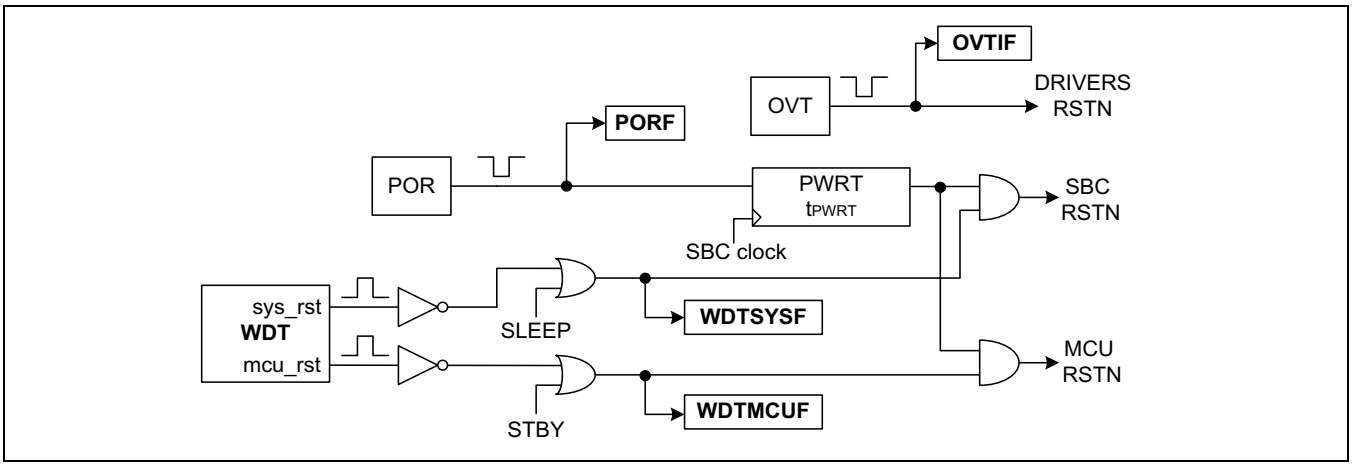
The OSCL clock is observed by dedicated circuitry running on the OSCH clock. If for any reason the OSCL clock stops, then an OSCL failure event is detected, captured in the flag OSCLEIF (see Table 3.1), and propagated as an interrupt to the MCU. Then the MCU firmware can determine whether to perform any failure protection actions (for example, a failure report to the electronic control unit ECU via LIN) or to continue with the regular firmware execution.

Note: In SLEEP Mode, the OSCL failure detection is disabled because the OSCH clock is stopped, i.e. no OSCL clock observation is possible.

3.2.2. Reset Sources

The ZAMC4100 system has three resets all generated within the SBC chip. Each reset source has its own flag bit. The MCU can distinguish between the different resets by reading the contents of register RSTSTAT (Table 3.1 and Table 3.2). The flags of the RSTSTAT register are cleared by writing a ‘1’ in the corresponding bit of the RSTSTAT register.

Figure 3.3 ZAMC4100 Reset Sources



Power-on-Reset (POR)

This global system reset is generated when the supply at the VDDE pin crosses the voltage thresholds specified in Table 1.5.

WDT Reset

There are two types of watchdog timer resets: the MCU reset and the global system reset. In the event of an abnormal condition in which the MCU fails to periodically clear the WDT counter, the WDT will attempt to reset the MCU. However after a configurable number of consecutive MCU resets, the WDT will invoke a global system reset where both the SBC and MCU will be re-initialized.

OVT Drivers Control Reset

The over-temperature (OVT) reset is generated if the SBC chip temperature reaches the value specified by the parameter OVT_{TH} (125ºC; see Table 1.5). If an OVT reset occurs, it clears all drivers' control registers. From a functional point of view, the OVT reset is equivalent to a shut-down signal for the drivers. It is implemented in a way that ensures that the drivers shut down even if the system clock is not present.

The driver control registers that are reset by an OVT reset event are HBDCTRL (Table 3.13), HBDDIAG (Table 3.17), HSDCTRL (Table 3.20), and ECMCTRL (Table 3.27). The OVT reset does not influence the MCU or the other SBC registers. If an OVT event has taken place, it is indicated in interrupt flag OVTIF of register RSTSTAT (see Table 3.1), and if interrupt by OVT is enabled (bit OVTIE = 1), the IRQN line is forced low.

In order to recover from an OVT event (i.e., to enable the drivers again), the MCU software should first clear interrupt flag OVTIF and then write the driver control registers again. As soon as the OVT event is active, the driver control registers are kept in reset; i.e., the drivers cannot be enabled. In order to ensure that the OVT event has been resolved, the MCU should try to clear OVTIF and re-check the flag. Successfully clearing OVTIF means that the chip has cooled-down to a temperature below 125°C; i.e., it will be possible to re-enable the drivers. As an additional step before re-enabling the drivers, it is recommended that the MCU measures the internal chip temperature (using the ADC). This avoids enabling the drivers when the SBC temperature is close to 125°C (e.g., 120°C) and as a result, the chip goes in OVT again a short time after the drivers are enabled.

© 2019 Renesas Electronics Corporation

Table 3.1 RSTSTAT Register Bits Mapping

Name	RSTSTAT							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	OSCLEIF	CPVEIF	OVIF	UVIF	OVTIF	WDTSYSF	WDTMCUF	PORF
Reset	0	0	0	0	0	0	0	1
Access	R/Wc	R/Wc	R/Wc	R/Wc	R/Wc	R/Wc	R/Wc	R/Wc
Address	**0x00**							
R/Wc = Read/Write '1' to clear; **U** = Unimplemented, read as '0.'								

Table 3.2 RSTSTAT Register Bits Description

Bit	Description
7	**OSCLEIF:** Low frequency oscillator (OSCL) error interrupt flag. 1 = The 125kHz clock has stopped due to OSCL failure. 0 = The OSCL is operating and the 125kHz clock is present.
6	**CPVEIF:** Charge pump voltage error interrupt flag. 1 = The charge pump voltage is out of range during driver operation or in the act of driver enabling. If this event occurs, all drivers are automatically disabled. 0 = Charge pump voltage is in range.
5	**OVIF:** Overvoltage interrupt flag. 1 = Overvoltage event occurred. If OVDP=1, all drivers are automatically disabled (see Table 3.5). 0 = Overvoltage event did not occur.
4	**UVIF:** Under-voltage interrupt flag. 1 = Under-voltage event occurred. If UVDP=1, all drivers are automatically disabled (see Table 3.5). 0 = Under-voltage event did not occur.
3	**OVTIF:** Over-temperature interrupt flag 1 = Over-temperature event occurred, and all drivers are disabled. 0 = Over-temperature event did not occur.
2	**WDTSYSF:** Watchdog timer system reset flag. 1 = The MCU has failed several times in clearing the WDT timer and system reset was generated. 0 = There is no system reset by WDT.
1	**WDTMCUF:** Watchdog timer MCU reset flag. 1 = MCU reset was caused by the WDT. 0 = There is no MCU reset by WDT.
0	**PORF:** Power-on-reset flag. 1 = System reset was caused by the POR block. 0 = There is no system reset by the POR block.

3.3. Watchdog Timer (WDT)

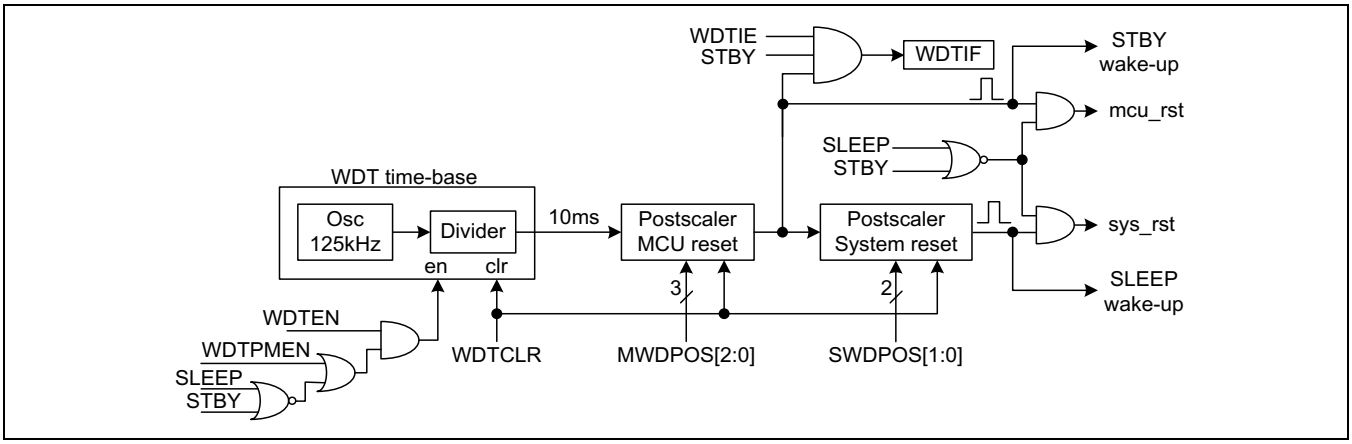
The ZAMC4100 has a watchdog timer implemented in the SBC chip with the following features:

- MCU reset and system reset generation
- Running on the dedicated oscillator clock (125kHz)
- 8 programmable time-outs for MCU reset generation
- 4 settings for system reset generation
- Configuration register lock bit for avoiding unintentional configuration changes
- WDT clearing mechanism using bit toggling
- WDT reset status bit showing the reset event (MCU or system reset)
- Usage as a wake-up event during SLEEP and STANDBY Mode

3.3.1. WDT Structure and Operation Principals

The watchdog timer (WDT) has a dedicated free running oscillator (OSCL), which allows operation even when the main oscillator has stopped (e.g., in SLEEP Mode). The WDT structure is shown in Figure 3.4.

Figure 3.4 ZAMC4100 Watchdog Timer



The WDT is configured and controlled via register WDTCONF (see Table 3.3). The WDT is enabled with bit WDTEN = 1. Once enabled, it continuously generates 10ms periods that are counted by two post-scalers and two time-out resets are generated:

1. Time-out for a MCU reset: after the configured time, the WDT resets the MCU.
2. Time-out for system reset: after the configured number of consecutive MCU resets, the WDT resets the whole system (SBC and MCU chip).

Toggling the WDTCLR bit periodically will clear the WDT as well as both post-scalers which will consequently avoid the WDT reset event. In order to toggle the WDTCLR bit, the MCU should do the following sequence:

1. Read the WDTCONF register
2. Invert the WDTCLR bit
3. Write back the WDTCONF register

This method requires SPI interface to be functional, which is effectively self-diagnostic. This ensures firmware within the MCU is operational and free from abnormal behavior.

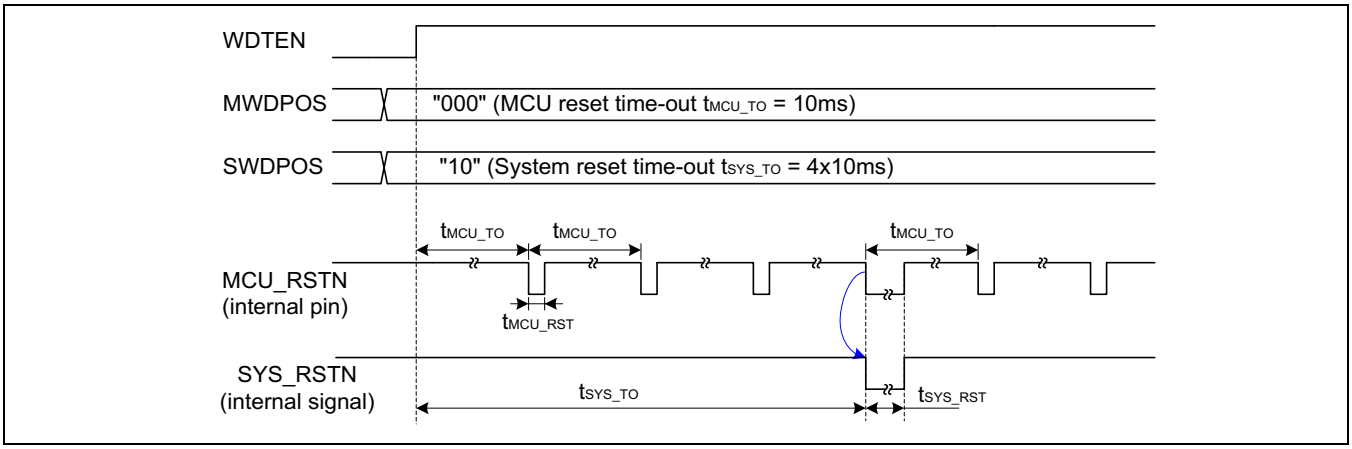
Once the WDT is configured, the write access to register WDTCONF can be locked with setting bit WDTLCK = 1. This avoids unintentional WDT reconfiguration or disabling caused by an MCU firmware malfunction. If the WDT is disabled, the current post-scaler values are automatically cleared.

Figure 3.5 shows an example of the MCU and system reset generation by the WDT. Parameters t_{MCU_RST} and t_{SYS_RST} are specified in Table 1.17. The duration of the system reset pulse is longer than the MCU reset pulse because the power-off/on cycle of the MCU regulators is executed.

Important notes:

1. If the WDT is enabled and the bit WDTLCK is set, the MCU cannot disable the WDT until the next power-on-reset.
2. Once the WDTLCK bit is set, it can be cleared only by a power-on reset.
3. The WDTLCK bit does not lock the write for the WDTCLR bit.
4. If the WDT is enabled, the configuration for the MCU reset and system reset time-out (bits SWDPOS and MWDPOS of register WDTCONF) cannot be changed.
5. The WDT time-base period of 10ms is calculated for the typical value of the OSCL frequency (see Table 1.14).
6. If a system reset by WDT is generated, the same flow is executed as in the case of a power-on reset (see Figure 3.1).

Figure 3.5 WDT Operation



3.3.2. WDT during SLEEP and STANDBY Mode

By default the WDT is disabled when the SLEEP or STANDBY Mode is activated. Since it uses a dedicated clock, if needed, the WDT can continue running when the system is in SLEEP or STANDBY Mode. This feature is enabled by setting bit WDTPMEN = 1 in register SMDCTRL (see Table 3.5).

If the WDT is running during the SLEEP Mode, then the system reset time-out (if enabled by SWDPOS configuration in the WDTCONF register) generates a wake-up from sleep instead of a system reset. Refer to section 3.5.4 for more information regarding wake-up sources.

If the WDT is running during STANDBY Mode, the “MCU reset time-out” generates a wake-up from standby instead of an MCU reset. Refer to section 3.5.4 for more information regarding wake-up sources.

Note: If the WDT configuration register is locked, the time-out intervals cannot be modified before SLEEP or STANDBY Mode entry.

3.3.3. WDT Configuration Register

The WDT configuration register bits mapping and description are given in Table 3.3 and Table 3.4 respectively. After POR, the WDT is enabled with a 160ms MCU time-out reset and no system reset generation.

Table 3.3 WDTCONF Register Bits Mapping

Name	**WDTCONF**							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	WDTCLR	SWDPOS[1]	SWDPOS[0]	MWDPOS[2]	MWDPOS[1]	MWDPOS[0]	WDTLCK	WDTEN
Reset	0	0	0	1	0	0	0	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	0x04							
R/W = Read/Write								

Table 3.4 WDTCONF Register Bits Description

Bit	Description
7	**WDTCLR:** WDT clear bit. Toggling this bit periodically to clear the WDT in order to avoid MCU reset. 1 = Next written value should be 0. 0 = Next written value should be 1. Note: The WDTLCK bit does not lock the write for the WDTCLR bit.
6:5	**SWDPOS:** WDT post-scaler for system time-out reset generation. 00 = No system reset generated by the WDT. 01 = System reset is generated at each MCU reset by the WDT. 10 = System reset is generated after 4 consecutive MCU resets. 11 = System reset is generated after 8 consecutive MCU resets.
4:2	**MWDPOS:** WDT postscaler for MCU time-out reset generation. 000 = 1:1 WDT time-out after 10 ms. 001 = 1:2 WDT time-out after 20 ms. 010 = 1:4 WDT time-out after 40 ms. 011 = 1:8 WDT time-out after 80 ms. 100 = 1:16 WDT time-out after 160 ms. 101 = 1:32 WDT time-out after 320 ms. 110 = 1:64 WDT time-out after 640 ms. 111 = 1:128 WDT time-out after 1280 ms. Note: WDT time-out period is calculated for the typical value of the OSCL frequency.
1	**WDTLCK:** WDT configuration lock bit. 1 = The writing to the WDTCONF register is not possible and writing can be enabled only after a power-on reset. 0 = The write access of WDTCONF is enabled.
0	**WDTEN:** WDT enable bit. 1 = The WDT is enabled, and if a time-out occurs, it will generate an MCU reset. 0 = The WDT is disabled. No reset is generated by WDT.

3.4. System Supply Monitoring

The ZAMC4100 continuously monitors the supply at the VDDE pin. If the VDDE voltage crosses the upper or lower ends of the supply range (see Table 1.5), an over-voltage or under-voltage interrupt is generated. In addition, the charge pump voltage is monitored for being in or out of range.

3.4.1. Over-Voltage and Under-Voltage Detection

If an over-voltage or under-voltage event occurs, it is indicated in the OVIF and UVIF flags respectively in the RSTSTAT register (see Table 3.1). Both interrupt flags in an OR-function generate the supply error interrupt flag (SEIF), which is accessible via the IRQSTAT register (see

Table 3.9). In order to clear the SEIF flag, the MCU should clear the OVIF and UVIF flags.

In order to prevent false over-voltage or under-voltage events being flagged (caused by supply spikes), the OV and UV interrupt signals are passed through a de-bounce circuit that filters out all events shorter than 20µs.

If the over-voltage or under-voltage event still exists, the OVIF and UVIF flags remain high even if the MCU is attempting to clear them.

If the SEIFAC bit of register SMDCTRL (Table 3.5) is set high, the OVIF and UVIF flags are automatically cleared when the over-voltage or under-voltage event is corrected and the supply remains stable for more than 20µs.

3.4.2. ZAMC4100 Operation during Over-Voltage

During an over-voltage event, the SBC and MCU are fully functional. By default, the over-voltage event automatically disables all drivers in order to prevent load damage. The MCU can re-enable the drivers by clearing the OVIF flag. If this protection is not needed, it can be disabled by clearing the OVDP bit of the SMDCTRL register (section 3.5.5).

3.4.3. ZAMC4100 Operation during Under-Voltage

When an under-voltage event occurs, the MCU chip is still fully functional. The SBC analog circuitry is operational, but its supply is out of range; therefore the analog parameters cannot be guaranteed.

During an under-voltage event, the internal charge pump voltage is out of range, and proper driver operation is not guaranteed. By default the under-voltage event automatically disables all drivers. The MCU can re-enable the drivers by clearing the UVIF flag in the RSTSTAT register (Table 3.1).

Important note: The under-voltage driver protection can be disabled by clearing the UVDP bit in the SMDCTRL register (section 3.5.5). This feature is implemented for test purposes only and is not recommended to be used in the regular application.

3.4.4. Charge Pump Voltage Monitoring

ZAMC4100 has internal circuitry for charge pump (CP) voltage monitoring, which is continually enabled during system operation. If the charge pump voltage is out of range, an interrupt is generated and the drivers are disabled. In order to prevent a false “charge pump voltage error” event flag (caused by supply spikes), the CP interrupt signal passes through de-bounce circuitry that filters out all events shorter than 20µs.

A charge pump voltage error interrupt is indicated by the CPVEIF flag in the RSTSTAT register (Table 3.1). If CPVEIF = HIGH, i.e. the CP voltage is out of range, all drivers are automatically disabled regardless of the value of the driver enable bits.

The MCU can clear the interrupt by writing ‘1’ in the CPVEIF flag, but only if the CP voltage has recovered to its valid range. If the CPVEIFAC bit of the SMDCTRL register (Table 3.5) is set high, then the CPVEIF flag is automatically cleared when the CP voltage recovers to its valid range and remains valid for more than 20µs.

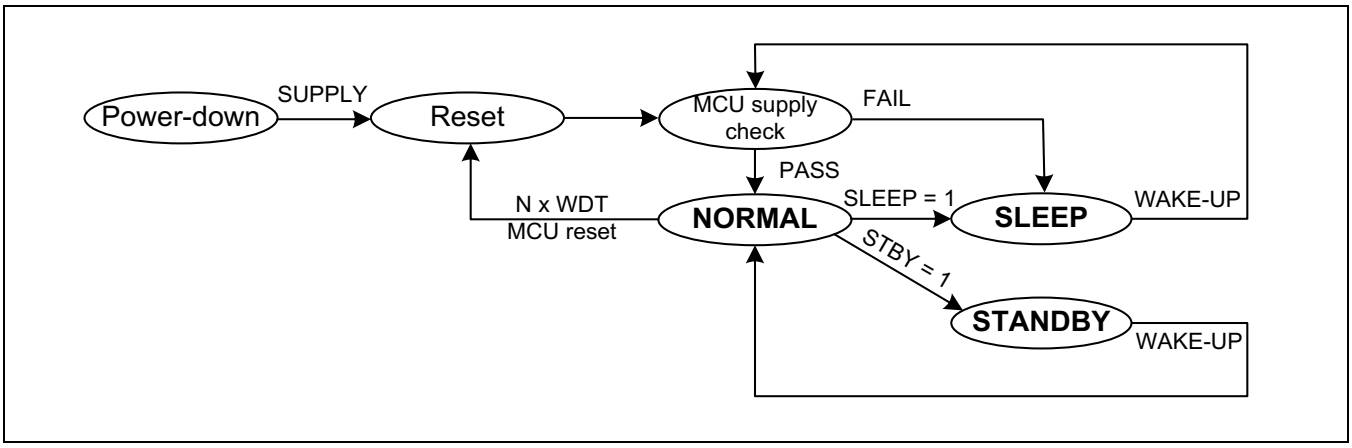
The CP voltage needs a settling time specified in Table 1.17 by parameter t_{CP_STL}. Because t_{CP_STL} is greater than the power-up timer delay (t_{PWRT}), after POR and wake-up from SLEEP, the CPVEIF flag will be set HIGH. If CPVEIF auto-clear is enabled, the flag will be automatically cleared when time t_{CP_STL} has expired, i.e. the CP voltage has reached its valid range.

© 2019 Renesas Electronics Corporation

3.5. ZAMC4100 System Operational Modes

The ZAMC4100 supports three operational modes: NORMAL, SLEEP, and STANDBY. The transition between the different operational modes is shown in Figure 3.6.

Figure 3.6 ZAMC4100 Operational Modes



After power-on-reset, the SBC performs an MCU supply check ensuring that both MCU voltages (BUF1V8 and REG3V3 – see section 2.1) are present and in range. Once the supply check has passed, the SBC enters NORMAL Mode; otherwise it enters SLEEP Mode and no further operation is allowed. In this way, it assures the correct functionally of the system in the event of unexpected faults.

3.5.1. NORMAL Mode

In NORMAL Mode, the system is fully functional waiting to receive LIN commands and controlling the drivers. From this mode, the system can branch in three directions:

- Power-save in SLEEP Mode (see section 3.5.2)
- Suspended MCU in STANDBY Mode (see section 3.5.3)
- WDT reset event, where software fails to clear the WDT's flag within the appropriate timing (see section 3.3.1)

© 2019 Renesas Electronics Corporation

3.5.2. SLEEP Mode

In SLEEP Mode, current consumption is very low, less than 80µA. The MCU chip is powered-down and the SBC supports only the functionality necessary for detecting a LIN wake-up or WDT wake-up event (see section 3.5.4). In this mode, all drivers are automatically disabled. The MCU enables SLEEP Mode by setting the SLEEP bit in the SMDCTRL register (Table 3.5). SLEEP Mode is exited if the SBC detects a wake-up event from the LIN bus and consequently it re-activates the MCU as well as re-enabling the drivers.

Figure 3.7 shows the system behavior when the SLEEP Mode is enabled. The procedure starts with switching-off the MCU supply and SBC analog blocks supply. In order to minimize power consumption, the SBC digital clock switches to the OSCL (125kHz).

Important note: If there is a failure in the OSCL (see section 3.2), the SLEEP Mode entry command is ignored. As a result, the system avoids undefined states with a powered-off MCU but with power supplied for the analog circuitry (see SLEEP Mode entry flow in Figure 3.7).

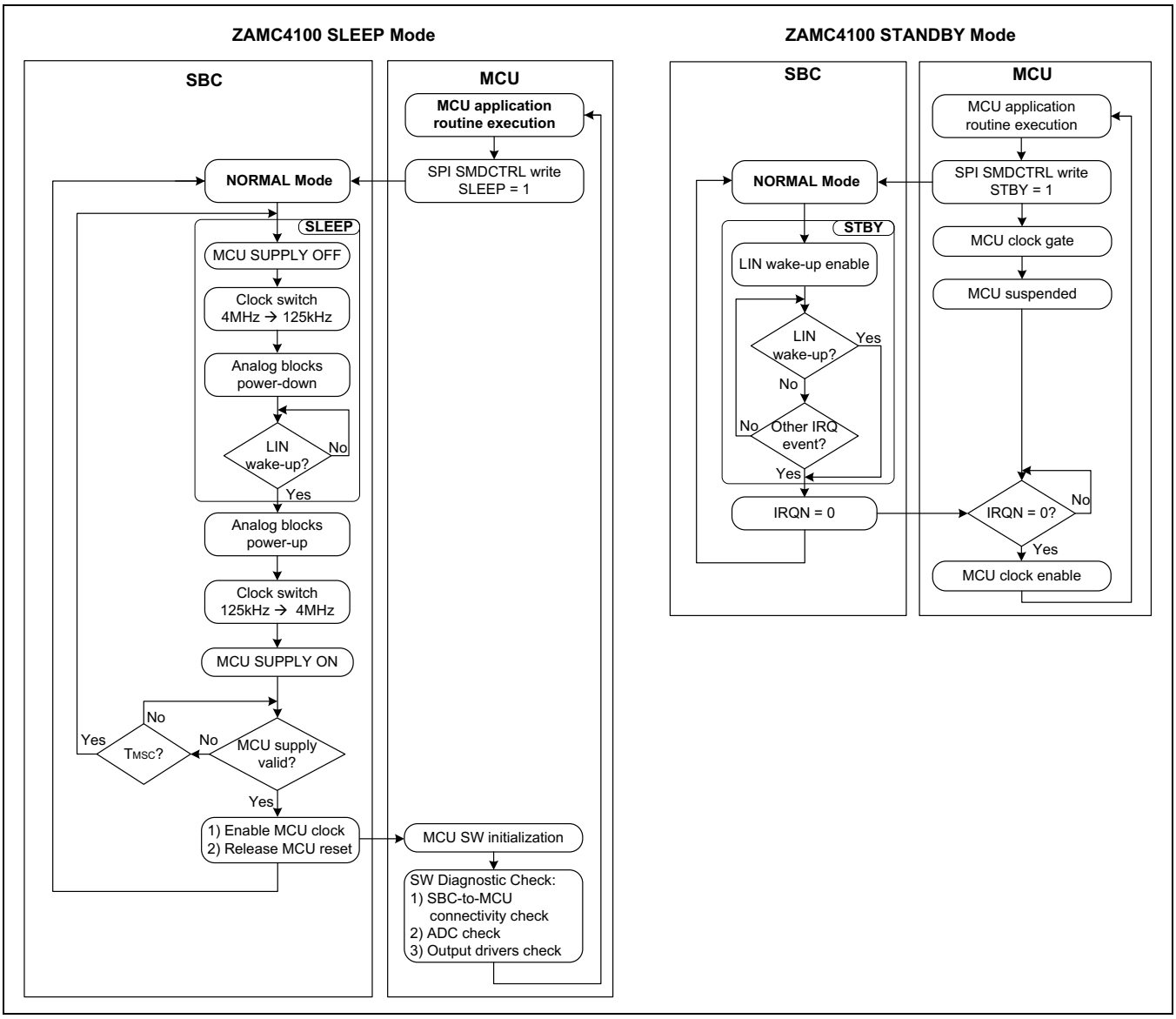
When a valid LIN wake-up is detected, the SBC chip starts the procedure for returning to the NORMAL Mode. During this procedure, the analog blocks are powered-up, the SBC digital clock is switched to 4MHz, and the MCU supply is enabled. Before enabling the MCU clock and releasing the MCU reset, the SBC performs an MCU supply check that is the same as the checks at system start-up. If the MCU supply voltages are in range, it starts operating; otherwise the system switches back to SLEEP Mode.

Important notes:

1) Activation of a driver immediately after wake-up from Sleep Mode is not possible due to the relatively slow settling time of the charge pump (see parameter t_{CP_STL} in Table 1.17). The CP condition is continuously monitored by the CPVEIF flag, which can be auto-cleared when a stable CP voltage is present if that functionality is enabled. Refer to section 3.4.4 for more details regarding charge pump voltage monitoring.
2) When SLEEP Mode is enabled, all drivers are automatically disabled. If needed, after wake-up the MCU should re-enable the drivers by setting the corresponding enable bits.

© 2019 Renesas Electronics Corporation

Figure 3.7 ZAMC4100 SLEEP and STANDBY Mode Behavior



3.5.3. STANDBY Mode

In STANDBY Mode, the clock in the MCU is suspended rather than powered down. This MCU suspension reduces power consumption (by a factor of two compared to the NORMAL Mode). Because the SBC analog circuitry and MCU remain supplied, the system quickly returns to functionality when coming out from STANDBY Mode.

The sequence of STANDBY Mode entry is shown in Figure 3.7. The MCU enables this mode by setting the STBY bit in the SMDCTRL register (Table 3.5). After enabling the STANDBY Mode, the MCU firmware must execute a dedicated clock gating instruction that will stop the MCU clock and therefore the program execution.

The MCU clock is enabled again if it receives an SBC interrupt on the IRQN line (see block diagram on page 4). The interrupt can be generated by a LIN wake-up event or from another SBC peripheral event. The IRQN interrupt generation automatically clears the STBY bit and the SBC switches to NORMAL Mode.

Note:

1) Before entering the STANDBY Mode, the MCU must clear all SBC interrupt flags in order to have the IRQN line = '1'. Otherwise setting the STBY bit = 1 will not take effect.
2) When STANDBY Mode is enabled, all drivers are automatically disabled. If needed after wake-up, the MCU should re-enable the drivers by setting the corresponding enable bits.
3) In order to have a LIN wake-up in STANDBY Mode, the interrupt enable bit LINIE in the IRQCTRL register (Table 3.7) should be set high.

3.5.4. Supported Wake-up Sources

When the ZAMC4100 system is in SLEEP or STANDBY Mode, it returns to NORMAL operation mode when a valid wake-up event is detected. The next sections describe the wake-up events supported by the ZAMC4100.

LIN Bus Wake-up

The wakeup detection protocol of the ZAMC4100 is based on the LIN 2.1 specification and discussed in detail in section 3.13.2. The LIN wake-up detection is automatically enabled when SLEEP or STANDBY Mode is invoked. When a LIN wake-up event occurs, it generates an interrupt that is indicated in the LINWUIF flag. Disabling the LIN wake-up interrupt does not disable the wake-up event.

WDT Wake-up on MCU Reset Time-Out (STANDBY Mode)

This wake-up is generated only when the system is in STANDBY Mode and the watchdog timer is enabled to run in STANDBY Mode. This is enabled by setting the WDTPMEN bit to '1' in the SMDCTRL register (Table 3.5). If the MCU reset time-out occurs it does not generate an MCU reset, only a wake-up event and return to NORMAL Mode.

If the WDT interrupt is enabled, it is indicated by the WDTIF bit in the IRQSTAT register (

© 2019 Renesas Electronics Corporation

Table 3.9). This interrupt is useful if the MCU has executed a clock-gating instruction and the software execution is stopped until an event on the IRQN line occurs.

WDT Wake-up on System Reset Time-Out (SLEEP Mode)

This wake-up is generated only when the system is in SLEEP Mode and the watchdog timer is enabled to run in SLEEP Mode (WDTPMEN = 1). When a system reset time-out occurs, it does not generate system reset, only a wake-up event and return to the NORMAL Mode.

Wake-up on SBC Interrupt (STANDBY Mode)

The previous sections described the most common sources for waking up the system and returning to the NORMAL Mode: the LIN wake up and WDT time-out. In general when the system is in STANDBY Mode, any SBC interrupt propagated on the IRQN line will restore the MCU clock and switch the system to the NORMAL Mode.

3.5.5. Operational Modes Control

Switching from NORMAL to SLEEP or STANDBY Mode is initiated by the MCU by setting the SLEEP or STBY bits of register SMDCTRL (Table 3.5).

The SLEEP Mode is activated by setting the SLEEP bit = 1. The STANDBY Mode is activated by setting the STBY bit = 1. If the MCU reads the SLEEP or STBY bits, the returned result is 0.

NOTE: If the MCU tries to set both bits at the same time, the write command will be ignored.

Table 3.5 SMDCTRL Register Bits Mapping

Name	**SMDCTRL**							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name		CPVEIFAC	SEIFAC	WDTPMEN	OVDP	UVDP	STBY	SLEEP
Reset	U	0	0	0	1	1	0	0
Access	U	R/W	R/W	R/W	R/W	R/W	W	W
Address	**0x03**							
R/W = Read/Write bit; **W** = Write-only bit, read as 0; **U** = Unimplemented, read as '0'								

Table 3.6 SMDCTRL Register Bits Description

Bit	Description
6	**CPVEIFAC:** “Charge Pump Voltage Error Interrupt Flags Automatic Clear” enable bit. 1 = CPVEIF in the RSTSTAT register (Table 3.1) is automatically cleared when the interrupt event is resolved. 0 = Automatic clearing of CPVEIF is disabled.
5	**SEIFAC:** “Supply Error Interrupt Flags Automatic Clear” enable bit. 1 = Supply error flags UVIF and OVIF in the RSTSTAT register are automatically cleared when the interrupt event is resolved. 0 = Automatic clear of supply error flags is disabled.
4	**WDTPMEN:** Bit for enabling the WDT during SLEEP and STANDBY Modes. 1 = WDT is enabled and running in SLEEP and STANDBY Modes. 0 = WDT is disabled in SLEEP and STANDBY Modes.
3	**OVDP:** “Over-Voltage Drivers Protection” enable bit. 1 = Over-voltage event automatically disables all drivers (HB, HS, ECM). 0 = The drivers are enabled even if an over-voltage event is detected.
2	**UVDP:** “Under-Voltage Drivers Protection” bit. 1 = Under-voltage event automatically disables all drivers (HB, HS, ECM). 0 = The drivers are enabled even if an under-voltage event is detected.
1	**STBY:** STANDBY Mode enable bit. 1 = Enables the STANDBY Mode. 0 = The system stays in NORMAL Mode.
0	**SLEEP:** SLEEP Mode enable bit. 1 = Enables the SLEEP Mode. 0 = The system stays in NORMAL Mode.

© 2019 Renesas Electronics Corporation

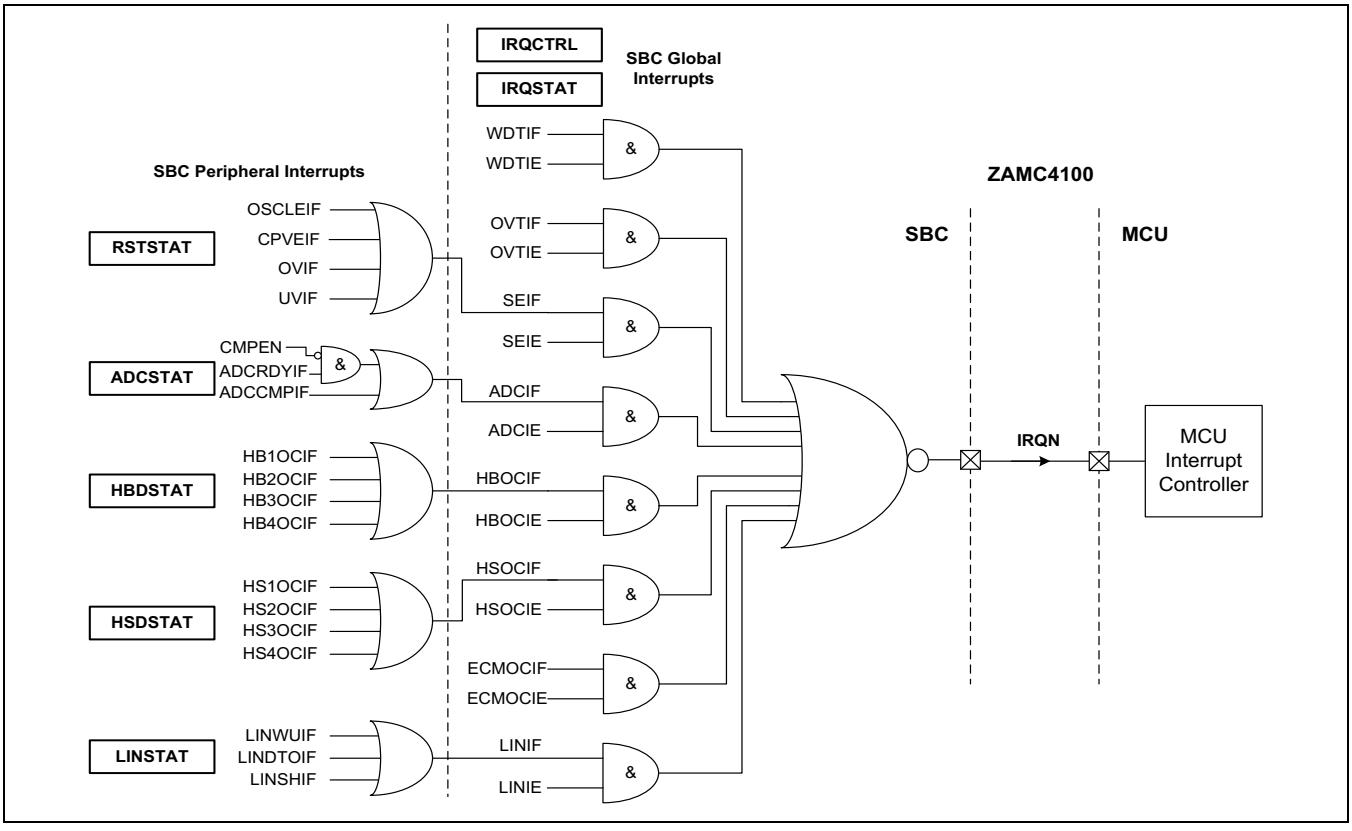
3.6. SBC Interrupts

Most of the interrupts are generated by the peripherals within SBC and MCU chips. The interrupt organization of the SBC chip is shown in Figure 3.8. The SBC interrupts are captured by the interrupt controller and after OR-reduction are propagated to the MCU via IRQN line. The IRQN line is active-low and appears as an external interrupt input to the MCU.

As shown in Figure 3.8, the interrupt logic has two levels: global interrupts and peripheral interrupts. Each interrupt source has an individual interrupt enable (IE) and interrupt flag (IF) bit. Some of the IF bits are mapped to the slave status byte and are shifted out to the MCU whenever a SPI command byte is shifted in to the SBC (Figure 3.10-D).

The global interrupts are controlled via the IRQCTRL register and read via IRQSTAT (see section 3.6.2). The peripheral interrupts are grouped by source and functionality and are combined using an OR-function to be a single global interrupt bit. This allows the MCU to first determine which peripheral is causing an interrupt and then to check the particular event. For example if the ADCIF interrupt flag bit is set, the reason might be that the ADC conversion is finished (ADCRDYIF=1) or it might come from the ADC result comparator (ADCCMPIF=1) (see section 3.7.5).

Figure 3.8 SBC Interrupt Logic Organization



3.6.1. SBC Interrupts: Enabling, Flagging, and Clearing Procedure

Each SBC global interrupt source has an individual interrupt enable (IE) bit. If an interrupt is enabled and the interrupt event occurs, it affects the IRQN line by forcing it to low. On power-on-reset all interrupts are disabled.

All interrupt sources are continuously observed during the system operation. If an interrupt event occurs, it is detected and indicated by setting the corresponding interrupt flag (IF) bit to '1,' which remains high until the MCU clears it. The interrupt events are detected and the corresponding interrupt flag is set even if the interrupt is disabled.

Each interrupt flag can be cleared by the MCU. In order to clear an interrupt, the MCU should write a logic '1' into its flag bit. This is a dummy write that is executed by the SBC as an IF clearing. In order to clear a global interrupt flag, the MCU must clear all peripheral interrupts belonging to the global flag. If the MCU executes the interrupt clearing procedure but the interrupt event is still present, the flag will not be cleared.

The ADCRDYIF flag (Table 3.12) supports a second clearing method, which is reading the ADC result register (see section 3.12.8). Another exception is the WDTIF watchdog interrupt, which is cleared by clearing the WDT (Table 3.4).

3.6.2. SBC Global Interrupts Registers

The SBC registers used for global interrupt enabling and reading are IRQCTRL and IRQSTAT. All global interrupt enable and flag bits are given in Table 3.7 and

© 2019 Renesas Electronics Corporation

Table 3.9. The details for each peripheral interrupt are given in the sections where the corresponding periphery is described.

Table 3.7 IRQCTRL Register Bits Mapping

Name	**IRQCTRL**							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	WDTIE	OVTIE	SEIE	ADCIE	HBOCIE	HSOCIE	ECMOCIE	LINIE
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	**0x02**							
R/W = Read/Write								

Table 3.8 IRQCTRL Register Bits Description

Bit	Description
7	**WDTIE:** Watchdog timer interrupt enable bit during STANDBY Mode. 1 = WDT interrupt is enabled. 0 = WDT interrupt is disabled. Note: This interrupt is generated only when the system is in STANDBY Mode.
6	**OVTIE:** Over-temperature interrupt enable bit. 1 = OVT interrupt is enabled. 0 = OVT interrupt is disabled. **Important:** If an over-temperature event occurs, the drivers will be shut down automatically regardless of the interrupt enable bit status.
5	**SEIE:** Supply error (over-voltage or under-voltage) interrupt enable bit. 1 = Supply error interrupt is enabled. 0 = Supply error interrupt is disabled.
4	**ADCTIE:** Global ADC interrupt enable bit. 1 = ADC interrupt is enabled. 0 = ADC interrupt is disabled.
3	**HBOCIE:** Global half-bridge driver over-current interrupt enable bit. 1 = Half-bridge driver over-current interrupt is enabled. 0 = Half-bridge driver over-current interrupt is disabled.
2	**HSOCIE:** Global high-side driver over-current interrupt enable bit. 1 = High-side driver over-current interrupt is enabled. 0 = High-side driver over-current interrupt is disabled.
1	**ECMOCIE:** Electrochromatic mirror (ECM) driver over-current interrupt enable bit. 1 = ECM over-current interrupt is enabled. 0 = ECM over-current interrupt is disabled.
0	**LINIE:** Global LIN PHY interrupt enable bit. 1 = LIN PHY interrupt is enabled. 0 = LIN PHY interrupt is disabled.

© 2019 Renesas Electronics Corporation

Table 3.9 IRQSTAT Register Bits Mapping

Name	IRQSTAT [1)]							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	WDTIF	OVTIF	SEIF	ADCIF	HBOCIF	HSOCIF	ECMOCIF	LINIF
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Address	**0x01**							
R = Read-only bit								
1) The ECMOCIF is physically implemented in register ECMDIAG (Table 3.30) and mapped to IRQSTAT.								

Table 3.10 IRQSTAT Register Bits Description

Bit	Description
7	WDTIF: WDT interrupt flag. 1 = Watchdog timer time-out occurred during STANDBY Mode. 0 = Watchdog timer interrupt did not occur. Note: In order to clear WDTIF, the MCU must clear the WDT by toggling the WDTCLR bit (see Table 3.4).
6	OVTIF: Over-temperature interrupt flag. 1 = Over-temperature event occurred. 0 = Over-temperature event did not occur. Note: In order to clear OVTIF, the MCU should clear the OVTF flag in register RSTSTAT (Table 3.1).
5	SEIF: Supply error interrupt flag. 1 = The system supply voltage is out of range. There was an overvoltage or under-voltage event, and as a result, one of the flags OVIF or UVIF is set. 0 = A supply error interrupt has not occurred. The supply voltage is in range. Note: The SEIF cannot be directly cleared. In order to clear SEIF, the MCU should clear the CPVEIF, OVIF, and UVIF flags in the RSTSTAT register.
4	ADCIF: ADC global interrupt flag. 1 = ADC interrupt occurred. The ADC conversion is ready or the comparator threshold has been reached. 0 = ADC interrupt did not occur. Note: The ADCIF interrupt flag cannot be directly cleared. ADCIF is cleared by clearing the correlated ADCRDYIF or ADCCMPIF flags in the ADCSTAT register (Table 3.39).
3	HBOCIF: Half-bridge driver over-current global interrupt flag. 1 = An over-current event was detected in one or more of the half-bridge drivers. To determine the specific over-current source, the MCU should read the HBxOCIF bits in the HBDSTAT register (Table 3.15). 0 = A half-bridge driver over-current event was not detected. Note: The HBOCIF interrupt flag cannot be directly cleared. In order to clear HBOCIF, the MCU should clear the HBxOCIF flags in the HBDSTAT register.

Bit	Description
2	HSOCIF: High-side driver over-current global interrupt flag. 1 = An over-current event has been detected in one or more high-side drivers. To determine the specific over-current source, the MCU should read the HSxOCIF bits in the HSDSTAT register. 0 = Half-bridge driver over-current event was not detected. Note: The HSOCIF interrupt flag cannot be directly cleared. In order to clear HSOCIF, the MCU should clear HSxOCIF flags of register HSDSTAT (Table 3.22).
1	ECMOCIF: Over-current interrupt flag for the ECM driver. This bit is physically implemented in register ECMDIAG and mapped to the IRQSTAT. 1 = An over-current event has been detected at the ECM output and the driver has been disabled. 0 = There is no over-current at the ECM output. Note: The ECMOCIF interrupt flag can be cleared in the ECMDIAG register (Table 3.29).
0	LINIF: LIN PHY global interrupt flag. 1 = LIN PHY interrupt has occurred. There was a LIN wake-up, dominant time-out, or LIN short detected. 0 = LIN PHY interrupt did not occur. Note: The LINIF interrupt flag cannot be directly cleared. In order to clear LINIF, the MCU should clear the correlated LINWUIF, LINDTOIF, or LINSHIF flags of the LINSTAT register (Table 3.48).

3.7. Serial Peripheral Interface (SPI) Bus Interface

The ZAMC4100 features an integrated Serial Peripheral Interface (SPI) that connects between the MCU and SBC. It operates as a synchronous serial interface performing full-duplex data exchange driven by the clock generated by the MCU. In the MCU, the SPI circuit block operates in MASTER mode and controls the SBC peripherals (drivers, ADC, LIN).

SPI Features:

- Byte-wise frame structure
- 8-bit data read/write access
- Clock polarity = 1 (clock idle = 1, falling edge first)
- Clock phase = 1
  - Data shift-out on falling edge
  - Data capturing on rising edge
- Low-active chip select (CSN)
- 8-bit SBC status automatically shifted out to MCU
- Pipelined write/read access for fastest SBC to MCU interaction
- Data write frame validation feature
- The SPI read starts at the specified address, and if the clock continues, it continues shifting out the data from next address
- Continuous data read mode (of a single address) without intermediate CSN toggling
  - Useful for ADC result data access

© 2019 Renesas Electronics Corporation

3.7.1. SBC and MCU SPI Connection Principles

The SBC to MCU connections using the SPI bus are shown in Figure 3.9. The SPI line definitions are given in Table 3.11.

Figure 3.9 MCU-to-SBC Connection via SPI

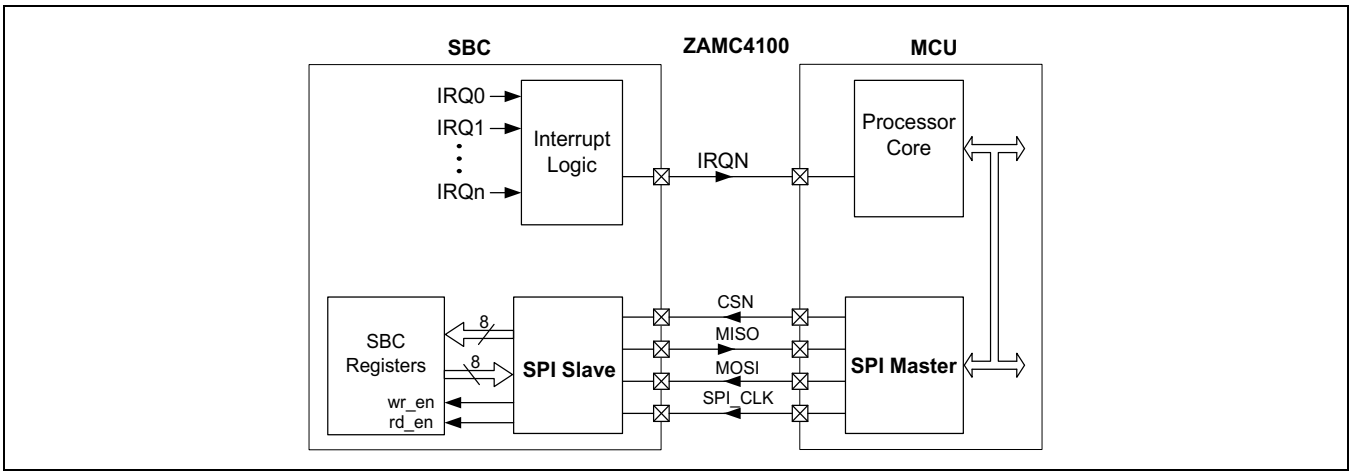


Table 3.11 SPI Lines Description

Line Name	Description
CSN	Active-low chip select with an internal pull-up in the SBC. The falling edge of the CSN line initiates the SPI data transfer. A high level on CSN disables the SPI of SBC.
MOSI	Master-Out Slave-In. The SBC has an internal pull-down resistor.
MISO	Master-In Slave-Out.
SPI_CLK	SPI clock with an internal pull-up in the SBC. In inactive state, the SPI_CLK line is in high. The value on the MISO and MOSI lines is changed on the falling edge of SPI_CLK and captured on the rising edge of SPI_CLK.

The SPI slave recognizes two frames: the data WRITE frame and the data READ frame (see Figure 3.10).

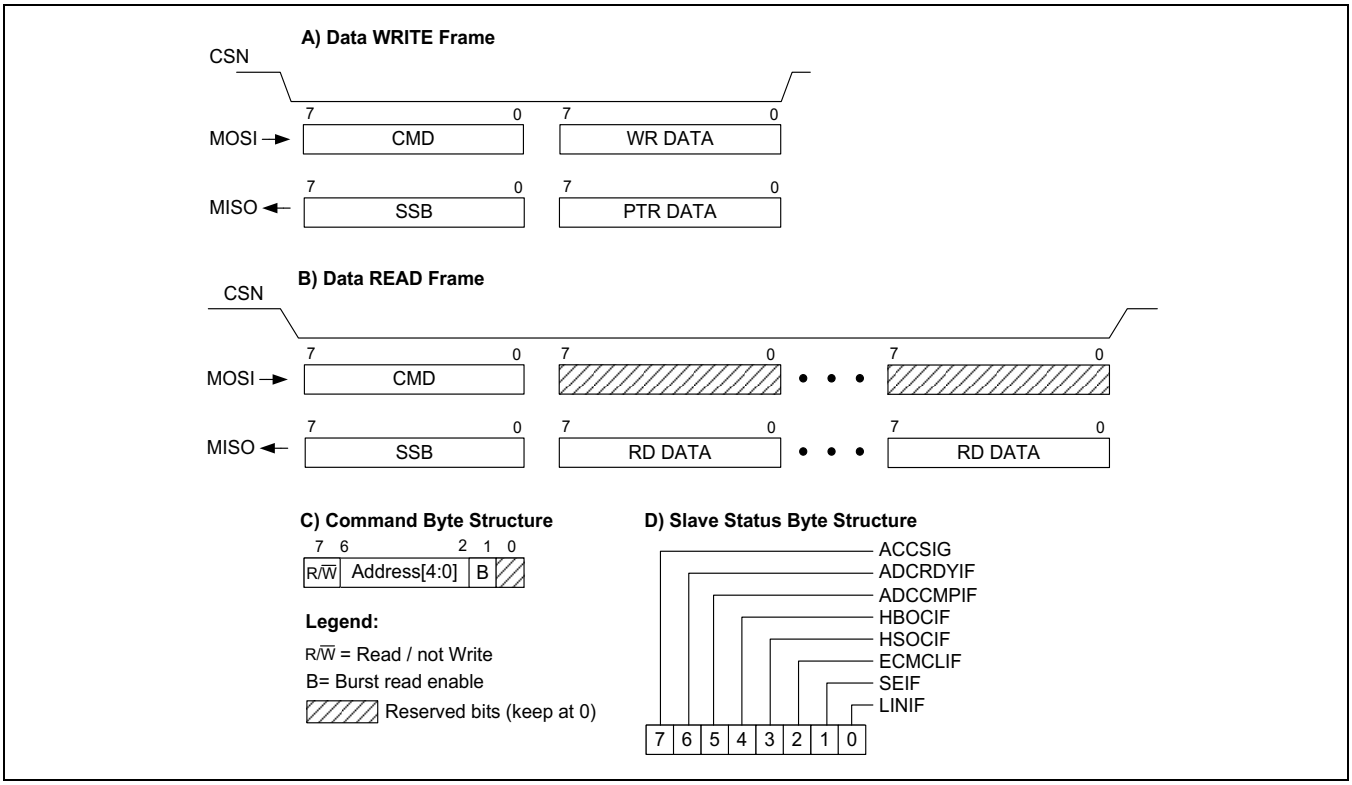
The SPI communication starts when the MCU switches the CSN line to LOW and ends when the CSN line goes HIGH. Each SPI frame begins with a command byte followed by the data byte(s).

The SPI command frame (Figure 3.10–C) contains the R/$\overline{W}$ bit for indicating the access type, five address bits for pointing to the accessed SBC register, and the B bit for enabling the burst read mode.

As the command byte is shifted in, the SBC automatically shifts out the slave status byte containing the important interrupt status bits (Figure 3.10–D).

© 2019 Renesas Electronics Corporation

Figure 3.10 SPI Frames



3.7.2. SPI Write Access

In order to write data in the SBC registers, the MCU should send a WRITE frame as shown in Figure 3.10-A. There are two bytes that the MCU master should send to SBC:

1. Command byte (CMD) with the R/$\overline{W}$ bit = 0, 5-bit address value and bit B = 0.
2. Data byte with the binary value that is going to be written into the corresponding address

When the data is shifted into the SBC, it shifts out the following bytes:

3. Slave status byte.
4. The value of the register pointed to by the SPIDPTR register (for more details, see Table 3.50). By default, SPIDPTR points to the SBC interrupt status register (IRQSTAT).

Note: Because the burst mode is not allowed during an SPI write, the B bit value is “don’t care.” Keep this value as ‘0’ for future product compatibility.

In order to have a successful data write, the SPI frame length should be exactly 2 bytes, i.e., 16 clocks. If the SPI slave receives more than 16 clocks between the falling and rising CSN edges, the WRITE frame is ignored and no WRITE command is executed.

If the data is successfully shifted into the SBC, the WRITE command is executed on the rising edge of CSN. This allows the MCU to control precisely the moment when the data is written. For example, the drivers are enabled by writing in their control register (see sections 3.8, 3.9, or 3.11) and in this case the CSN rising edge marks the beginning of the motor operation.

3.7.3. SPI Pipelined Write/Read Access

In order to provide bidirectional data transfer, the SBC SPI slave provides functionality for performing a register data READ while the WRITE frame is transferred. This functionality is implemented as automatic indirect addressing of the SBC registers using the SPIDPTR register (Table 3.50) as the data pointer.

Register SPIDPTR points to the address of the register from which content will be automatically shifted out while the SBC receives the data WRITE byte. This feature allows pipelining of the ADC conversion and the data WRITE (see section 3.12.7).

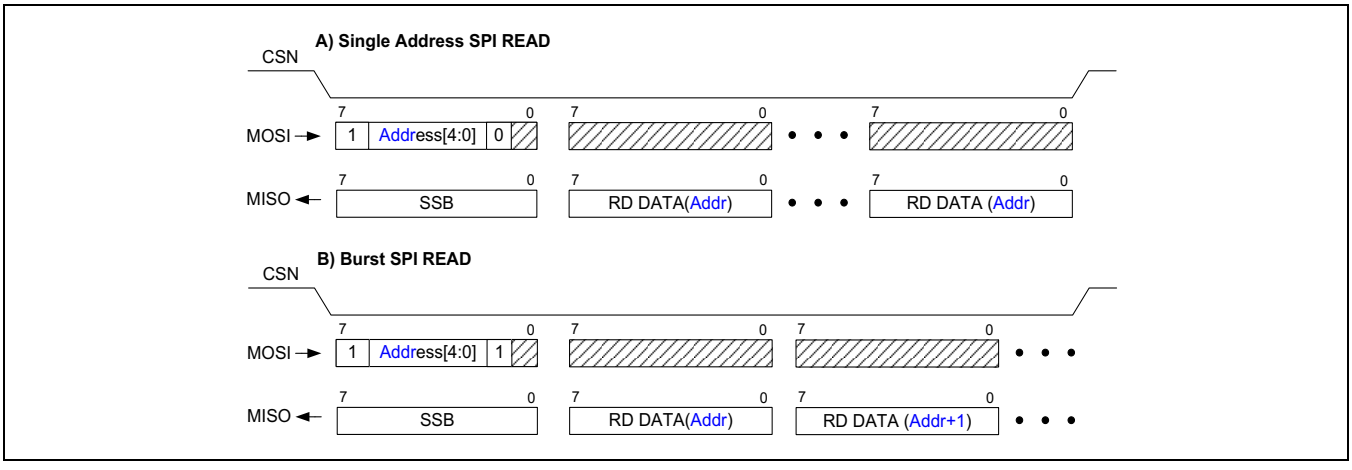
3.7.4. SPI Read Access

To read data from SBC registers, the MCU sends a frame as shown in Figure 3.10–B.

1) Command byte (CMD) with the R/$\overline{W}$ bit = 1, the 5-bit address of the register that is going to be read, and the B bit with a value depending on the READ access type:
    a. Single data read: B = 0
    b. Burst data read: B = 1
2) Data byte: The MCU sends dummy data bits (keep as 0) while the SBC shifts out the contents of the register pointed to by the address value in the command field.

The SPI read access allows the MCU to read more than one data byte. The SBC shifts out data as long as the CSN line is low and the SPI clock is running. If bit B = 0, the SBC shifts out all data from only one address, i.e. continuous reading of a single address (Figure 3.11-A). If bit B = 1, the SBC shifts out multiple consecutive data bytes and increments the address after sending each byte (Figure 3.11-B). In the case of a one-byte READ, the B bit value is “don’t care.”

Figure 3.11 SPI Single or Burst Access



3.7.5. SPI Slave Status Byte

The SPI slave status byte (SSB) contains important status bits that the SBC shifts out while the command SPI byte is shifted in. The purpose of each SSB bit is described in Table 3.12. Most of the SSB bits are copies of the SBC interrupt flags.

Table 3.12 SPI Slave Status Bits Descriptions (SSB)

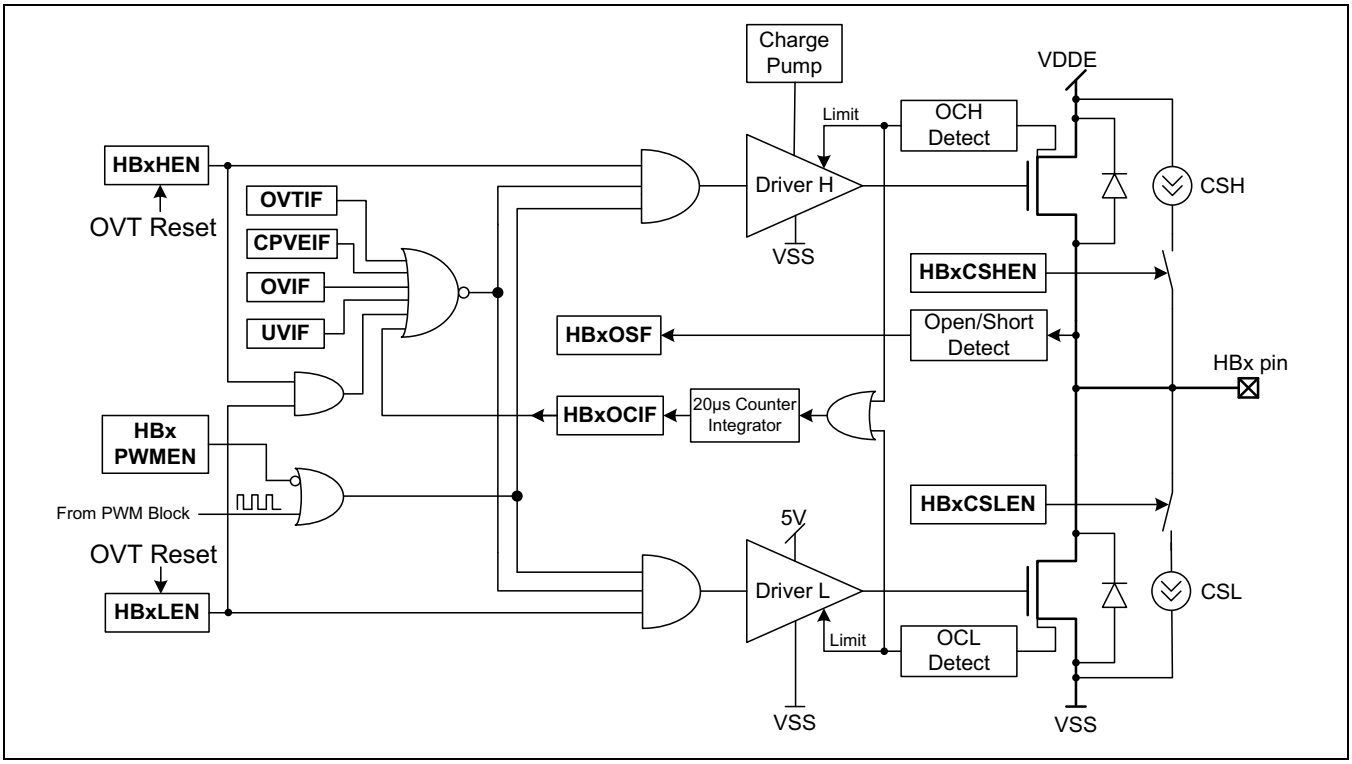
Bit	Name	Description
7	**ACCSIG**	The access signature that is currently configured. See section 3.14.2.
6	**ADCRDYIF**	ADC ready interrupt flag. See section 3.12.7.
5	**ADCCMPIF**	ADC comparator interrupt flag. See section 3.12.9.
4	**HBOCIF**	Over-current interrupt flag for half-bridge drivers. See section 3.8.
3	**HSOCIF**	Over-current interrupt flag for high-side drivers. See section 3.9.
2	**ECMOCIF**	ECM driver interrupt flag. See section 3.11.
1	**SEIF**	Supply error interrupt flag. See section 3.4.
0	**LINIF**	LIN interrupt flag. See section 3.13.

© 2019 Renesas Electronics Corporation

3.8. Half-Bridge Drivers

Within the SBC chip, there are 4 half-bridge (HB) drivers for HB1 to HB4. Figure 3.12 shows the block diagram, which is the same for each HB driver.

Figure 3.12 Structure of the Half-Bridge Drivers



Features of the HB Drivers:

- Separate enable bit for each MOSFET. If both enable bits are set, then both MOSFETs are off; this safety logic avoids the possibility of turning on both NMOS.
- Separate PWM enable bit for each driver.
- Each MOSFET has over-current detection (OCH and OCL for the high and low side drivers respectively) and protection circuitry. The over-current event automatically disables the driver (see section 3.8.2).
- If an over-voltage or under-voltage is detected at the VDDE pin the drivers are automatically disabled (see section 3.4).
- Each driver has diagnostic circuitry for detection of open or short conditions in the output load (see section 3.8.3).

© 2019 Renesas Electronics Corporation

3.8.1. Half-Bridge Drivers Control and Status Register

All half-bridge drivers are enabled and disabled using the HBDCTRL register. The bit mapping of HBDCTRL is given in Table 3.13, and the bit descriptions are in Table 3.14. The over-current and diagnostic status bits are allocated in register HBDSTAT, which is described in Table 3.15 and Table 3.16.

Table 3.13 HBDCTRL Register Bits Mapping

Name	HBDCTRL							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	HB4HEN	HB4LEN	HB3HEN	HB3LEN	HB2HEN	HB2LEN	HB1HEN	HB1LEN
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	**0x06**							
R/W = Read/Write								

Table 3.14 HBDCTRL Bits Description

Bit	Description
7,5,3,1	**HBxHEN:** [1), 2), 3), 4)] High-side MOSFET enable bit for half-bridge driver x, (x=4 to 1) 1 = High-side MOSFET enabled 0 = High-side MOSFET disabled
6,4,2,0	**HBxLEN:** [1), 2), 3), 4)] Low-side MOSFET enable bit for half-bridge driver x, (x=4 to 1) 1 = Low-side MOSFET enabled 0 = Low-side MOSFET disabled

1) When both enable bits of one driver are set or cleared (for example, HB1HEN = 1 and HB1LEN = 1 or HB1HEN = 0 and HB1LEN = 0), the driver is disabled.

2) Bits HBxHEN and HBxLEN are automatically cleared upon entry into SLEEP or STANDBY Mode.

3) These bits are automatically cleared if there is an over-temperature event.

4) These bits are gated if a failure event (OVT, OC, OV, UV, or CPVE) is detected, and a corresponding interrupt flag is set (see Figure 3.12).

© 2019 Renesas Electronics Corporation

Table 3.15 HBDSTAT Register Bits Mapping

Name	HBDSTAT							
Bit No	7	6	5	4	3	2	1	0
Bit name	HB4OSF	HB3OSF	HB2OSF	HB1OSF	HB4OCIF	HB3OCIF	HB2OCIF	HB1OCIF
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/Wc	R/Wc	R/Wc	R/Wc
Address	0x0A							
R/Wc = Read/Write 1 to clear, **R =** Read-only								

Table 3.16 HBDSTAT Register Bit Descriptions

Bit	Description
7:4	**HBxOSF:** Open/short status flag for half-bridge driver x (x = 4 to 1, for HB4 to HB1 respectively). This flag is not registered; i.e., its value stays as long as the diagnostic event takes place. The meaning of these bits depends on the settings of bits HBxCSLEN and HBxCSHEN of the HBDDIAG register and are given in Table 3.19 as explained in section 3.8.3.
3:0	**HBxOCIF:** Over-current interrupt flag for half-bridge driver x (x = 4 to 1, for HB4 to HB1 respectively). See section 3.8.2. 1 = There is an over-current in the output of driver x. The driver x is disabled as long as this bit is set. 0 = There is no over-current in the output of driver x.

3.8.2. Half-Bridge Over-Current Protection

In the case of an over-current event, the corresponding driver automatically limits the current in order to prevent damage in the MOSFET stage. If the over-current condition continues more than 20µs, the corresponding interrupt flag (HBxOCIF; Table 3.16) is set and the driver is automatically disabled. The MCU can re-enable the driver by clearing its interrupt flag. If the over-current condition still exists after 20µs, the driver will be disabled again.

The time of 20µs is counted by a counter functioning like an integrator. The counter works in increment or decrement mode determined by following conditions:

- The counter only increments per unit time in response to an over-current occurrence.
- The counter only decrements (to zero) per unit time in response to normal current operation.

Figure 3.13 illustrates the over-current function using three typical examples of over-current occurrences in an application:

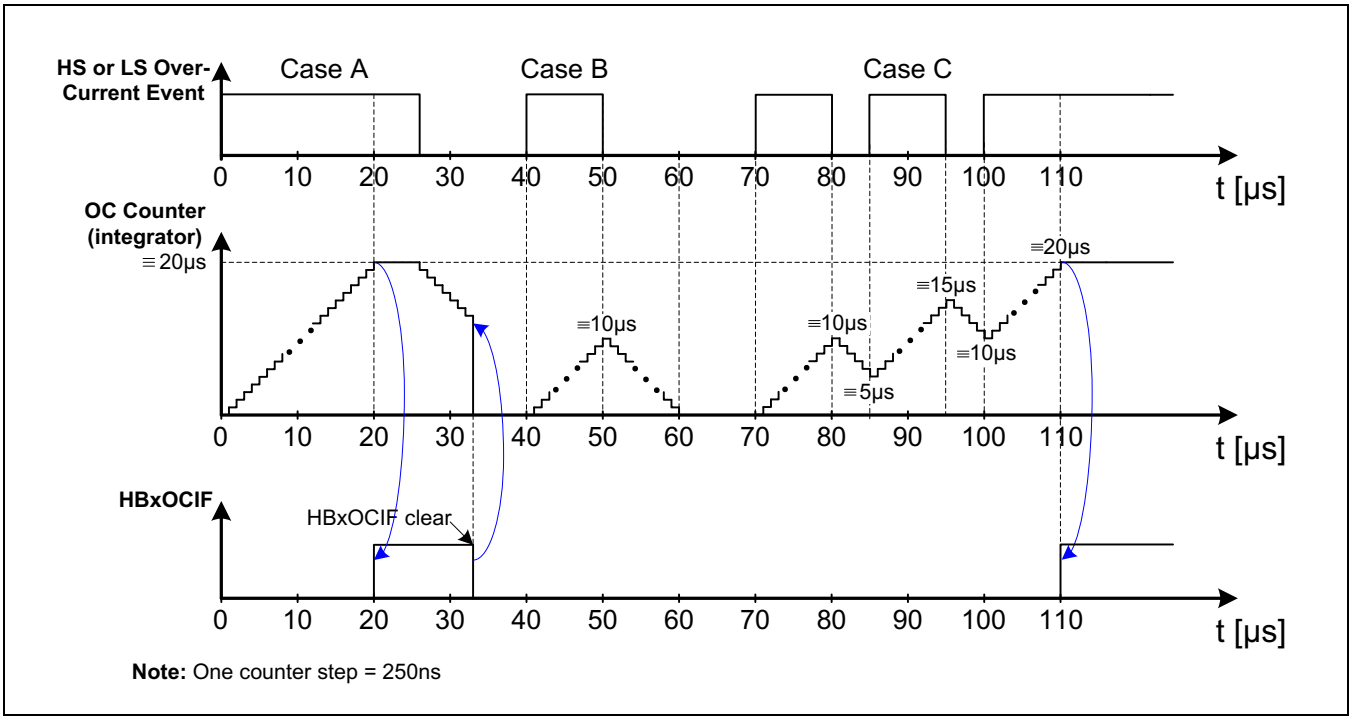
Case A: Operating with persistent over-current occurrence for more than 20µs. The counter increments constantly until expiration after 20µs, the interrupt flag is set, and the MOSFET is automatically disabled.

Case B: The over-current event occurs and then is resolved after 10µs. The counter will increment for 10µs and then decrement for 10µs until its value is back to zero.

Case C: There are three consecutive over-current events of 10µs duration with 5µs intervening pauses. In this case, the counter increments for 10µs, then decrements for 5µs, again increments for 10µs and so on. The process continues until the counter value reaches 20µs, the interrupt flag is set, and the MOSFET is disabled.

Case C shows the main idea of the counter integrator concept: consecutive over-current events with a duration shorter than 20µs are integrated over the time period and can set the over-current flag, which disables the driver. This guarantees no chip overheating in the case of frequent over-current events shorter than 20µs.

Figure 3.13 Operation Principle of Over-Current Counter Integrator



3.8.3. Half-Bridge Driver Output Diagnostic Check

Each half-bridge driver has dedicated diagnostic circuitry that is able to detect open/short failures in the output load. The HB driver diagnostic circuitry is controlled by the HBDDIAG register (Table 3.17 and Table 3.18), and the diagnostic flags are monitored by reading register HBDSTAT.

Table 3.19 gives the meaning of the corresponding HBxOSF flags in the HBDSTAT register with respect to the load configuration and the combination of the HBxCSHEN and HBxCSLEN bits of HBDDIAG register, which controls the diagnostic current sources/sinks.

Important note: When the diagnostic circuitry is enabled (one of the HBxCSHEN bits or the HBxCSLEN bits is set), then the corresponding over-current protection circuitry is not active. Therefore, enabling the driver when the diagnostic circuitry is enabled is not recommended!

Table 3.17 HBDDIAG Register Bits Mapping

Name	**HBDDIAG**							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	HB4CSHEN	HB4CSLEN	HB3CSHEN	HB3CSLEN	HB2CSHEN	HB2CSLEN	HB1CSHEN	HB1CSLEN
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	**0x07**							
R/W = Read/Write bit								

Table 3.18 HBDDIAG Register Bits Description

Bit	**Description**
7,5,3,1	**HBxCSHEN:** [1), 2)] High-side current source enable bit for driver x, (x = 4 to 1, for HB4 to HB1 respectively). 1 = The high-side current source of driver x is enabled. 0 = The high-side current source of driver x is disabled.
6,4,2,0	**HBxCSLEN:** [1), 2)] Low-side current source enable bit for driver x, (x = 4 to 1, for HB4 to HB1 respectively). 1 = The low-side current source of driver x is enabled. 0 = The low-side current source of driver x is disabled.

1) In order to keep current consumption as low as possible, the MCU software should disable the current sources before enabling the SLEEP or STANDBY Modes.

2) In the event of an over-temperature event, these current sources are automatically disabled.

Table 3.19 Half-Bridge Diagnostic Bit Descriptions

Note: See important notes at end of table.

HBxCSHEN [1]	HBxCSLEN [1]	HByLEN [1]	HBxOSF [1]	Description [2]
0	0	“don’t care”	0	Diagnostic disabled.
1	1	0	“don’t care”	Undefined result.
1	1	1	0	Not applicable.
Low-resistance load connected to VSS				
1	0	-	0	Normal load or short to VSS.
0	1	-	0	
1	0	-	1	Short to VDDE.
0	1	-	1	
1	0	-	1	Open load.
0	1	-	0	
1	0	-	0	Invalid.
0	1	-	1	
Low-resistance load connected to VDDE				
1	0	-	0	Short to VSS.
0	1	-	0	
1	0	-	1	Normal load or short to VDDE.
0	1	-	1	
1	0	-	1	Open load.
0	1	-	0	
1	0	-	0	Impossible.
0	1	-	1	
Load connected between two half-bridges [3]				
1	0	0	0	Short to VSS.
0	1	0	0	
1	0	0	1	Short to VDDE.
0	1	0	1	
1	0	0	1	Normal load.
0	1	0	0	
1	0	1	0	
0	1	1	0	
1	0	0	1	Open load.
0	1	0	0	
1	0	1	1	

© 2019 Renesas Electronics Corporation

HBxCSHEN [1]	HBxCSLEN [1]	HByLEN [1]	HBxOSF [1]	Description [2]
0	1	1	0	
1	0	0	0	Invalid.
0	1	0	1	

1) In the case of a load connected between two half-bridges (full-bridge configuration), “x” corresponds to one of the half-bridges (x = 1 to 4) and “y” corresponds to the other half-bridge (y = 1 to 4). In the case of a load connected between the output of the driver and one of the supply rails, the column “HByLEN” is not applicable. See Table 3.13 for more information about the HByLEN bits.

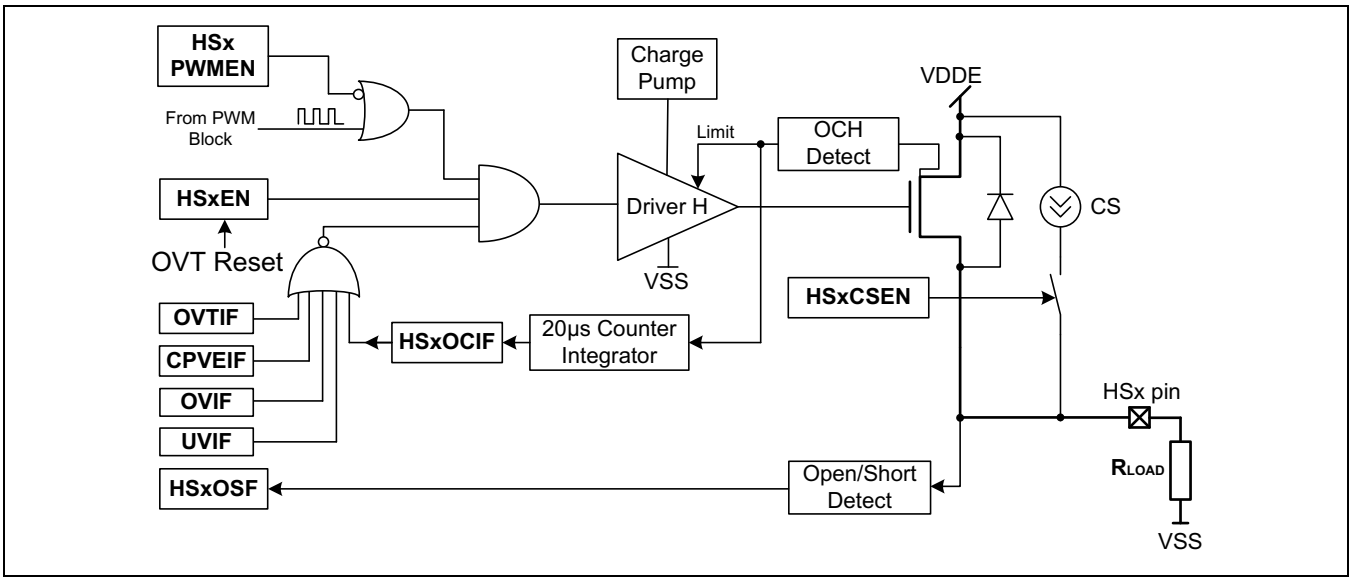
2) **Important:** The result of the diagnostic procedure is correct only if the drivers involved are switched-off during the diagnostic procedure and are turned on only when explicitly required according to this table.

3) The diagnostic procedure for full-bridge configuration includes enabling one of the low-side drivers, which makes a complete diagnostic of the load possible.

3.9. High-Side Drivers

Within the SBC, there are 4 high-side drivers for HS1 to HS4. Figure 3.14 shows the block diagram for the each driver. It is essentially the same as the half-bridge but without the low-side MOSFET.

Figure 3.14 Structure of High-Side Drivers



Features of High-Side Drivers:

- Separate enable bit for each driver.
- Separate PWM enable bit for each driver.
- Each NMOS has over-current detection and protection circuitry. The over-current event automatically disables the driver.
- If an over-voltage or under-voltage condition is detected at VDDE pin the drivers are automatically disabled.
- Each driver has diagnostic circuitry for detecting an open or short in the output load.

© 2019 Renesas Electronics Corporation

3.9.1. High-Side Control and Status Register

Table 3.20 HSDCTRL Register Bits Mapping

Name	HSDCTRL							
Bit No	7	6	5	4	3	2	1	0
Bit name	HS4CSEN	HS3CSEN	HS2CSEN	HS1CSEN	HS4EN	HS3EN	HS2EN	HS1EN
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	0x08							
R/W = Read/Write								

Table 3.21 HSDCTRL Register Bits Description

Bit	Description
7:4	**HSxCSEN:** [1), 2)] Current source enable bit for high-side driver x, (x = 4 to 1). 1 = The current source of high-side driver x is enabled. 0 = The current source of high-side driver x is disabled.
3:0	**HSxEN:** [2), 3), 4)] High-side driver enable. 1 = High-side driver x (x = 4 to 1) enabled. 0 = High-side driver x (x = 4 to 1) disabled. Note: Bits HSxEN are automatically cleared at SLEEP or STANDBY Mode entry.

1) To keep current consumption as low as possible, the MCU software should disable the current sources before enabling the SLEEP or STANDBY Modes.
2) These bits are automatically cleared in the event of an over-temperature event.
3) Bits HSxEN are automatically cleared at SLEEP or STANDBY Mode entry.
4) These bits are gated if failure event (OVT, OC, OV, UV, CPVE) is captured in interrupt flag (see Figure 3.14).

3.9.2. High-Side Over-Current Protection

The high-side over-current protection circuitry is identical to the protection for the half-bridge driver previously discussed in section 3.8.2.

3.9.3. High-Side Drivers Output Diagnostic Check

Each high-side driver has a dedicated diagnostic circuitry that is able to detect an open/short failure in the output load. The diagnostic functionality is controlled and monitored respectively by registers HSDCTRL (Table 3.20 and Table 3.21) and HSDSTAT (Table 3.22 and Table 3.23). Table 3.24 gives the meaning of the HSxOSF flags in the HSDSTAT register with respect to the value of bit HSxCSEN in the HSDCTRL register, which controls the corresponding diagnostic current source.

© 2019 Renesas Electronics Corporation

Important note: When the diagnostic circuitry is enabled (one of the HSxCSEN bits is set), then the corresponding over-current protection circuitry is not active. Therefore enabling the driver when the diagnostic circuitry is enabled is not recommended!

Table 3.22 HSDSTAT Register Bits Mapping

Name	HSDSTAT							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	HS4OSF	HS3OSF	HS2OSF	HS1OSF	HS4OCIF	HS3OCIF	HS2OCIF	HS1OCIF
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/Wc	R/Wc	R/Wc	R/Wc
Address	**0x0B**							
R/Wc = Read/Write 1 to clear; **R** = Read-only bit								

Table 3.23 HSDSTAT Register Bits Description

Bit	Description
7:4	**HSxOSF:** Open/short status flag of the high-side driver x (x = 4 to 1). This flag is not registered; i.e., its value remains as long as the diagnostic event takes place. Refer to Table 3.24 for the descriptions of the HSxOSF bits.
3:0	**HSxOCIF:** Over-current interrupt flag of the high-side driver x (x = 4 to 1). 1 = There is an over-current in the output of driver x. The driver x is disabled as long as this bit is set. 0 = There is no over-current in the output of driver x.

Table 3.24 High-Side Diagnostic Bits Description

HSxCSHEN	HSxOSF	Description [1)]
0	0	Normal load or short to VSS [2)]
1	0	
0	0	Open load
1	1	
0	1	Short to VDDE
1	1	
0	1	Not possible in normal application
1	0	

1) **Important:** The diagnostic result is correct only if the corresponding driver is switched off.
2) In the event of a short, an over-current will occur.

3.10. Pulse Width Modulation (PWM)

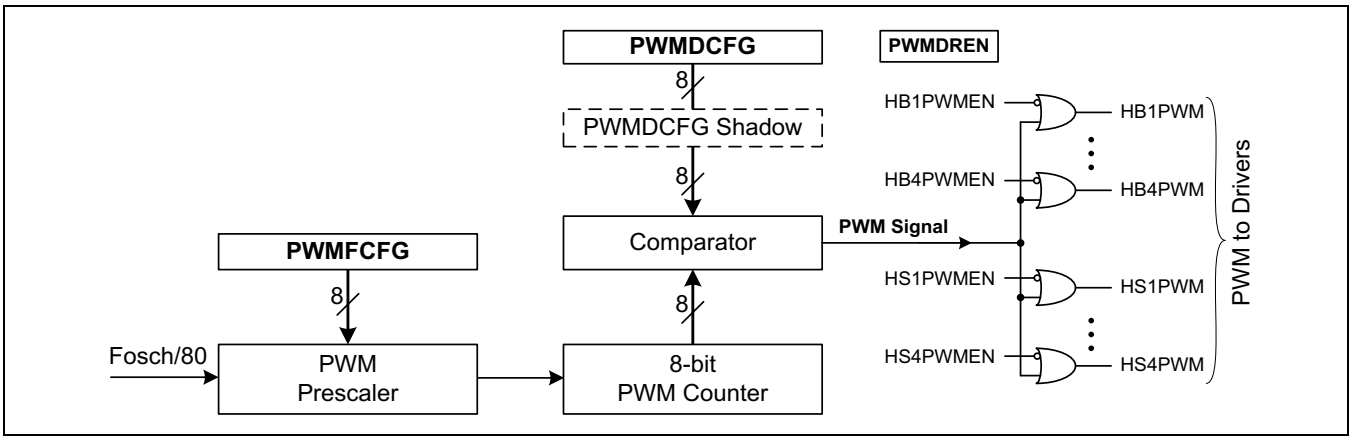
The ZAMC4100 SBC incorporates an 8-bit pulse width modulation module (PWM), which is designed to regulate discrete power levels of the half-bridge and high-side driver's output. The PWM module is a highly configurable solution by the MCU via SPI.

3.10.1. Overview

The PWM block supports the following features:

- PWM control for each half-bridge or high-side driver
- Two or more drivers can be simultaneously controlled by a PWM block
- When PWM control is not required, it can be disabled
- Maximum frequency: ~1kHz
- PWM frequency resolution: 8-bit
- PWM duty-cycle resolution: 8-bit

Figure 3.15 PWM Block Diagram



This module has three registers as summarized below:

- PWM frequency configuration register PWMFCFG (Table 3.50)
- PWM duty cycle configuration register PWMDCFG (Table 3.50)
- PWM driver control enable register PWMDREN (see Table 3.25)

On power-on-reset, the value of register PWMDCFG is zero and the PWM output is continuously low; i.e. the PWM is disabled. In order to start the PWM signal generation, the MCU should write to register PWMDCFG with a value different than zero. The PWMDCFG register can be written at any time, but internally the duty-cycle value is latched in a shadow register each time the PWM counter crosses zero. This guarantees a glitch-free signal at the PWM output by avoiding duty-cycle changes within the PWM period.

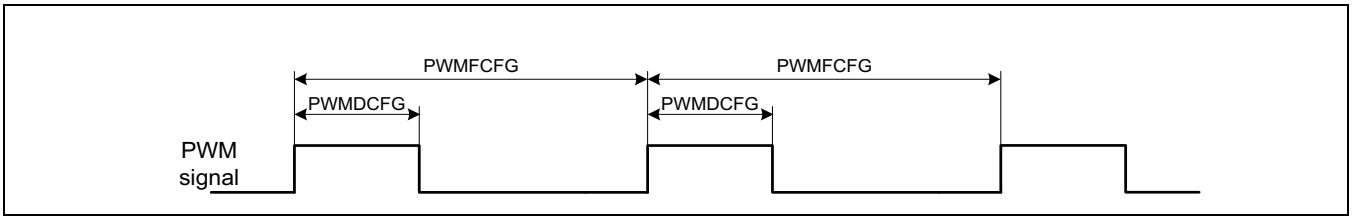
© 2019 Renesas Electronics Corporation

Register PWMDREN contains local bits for enabling the PWM control for each half-bridge or high-side driver. When all bits of PWMDREN are 0 (power-on-reset value), there is no driver using PWM control. In this case, the PWM module is automatically disabled even if PWMDCFG is not 0.

3.10.2. PWM Signal Generation

The PWM module is enabled when at least one bit of the PWMDREN register is set (see section 3.10.3). In that case, a signal with an adjustable frequency and duty-cycle is generated at the output of the PWM module (see Figure 3.16).

Figure 3.16 PWM Output Signal



The PWM frequency is configured via register PWMFCFG and is calculated using the formula:

$$PWMfreq = \frac{(Fosch / 80)}{256 \times [PWMFCFG + 1]}, [Hz] \rightarrow PWMFCFG \in [0:255]$$

The value of PWMFCFG determines the division ratio of the PWM prescaler (see Figure 3.15). The prescaler determines the speed of incrementing the PWM counter value from 0 to 255, i.e. the duration of the PWM period. Setting a minimum value for PWMFCFG provides a maximum PWM frequency of approximately 976Hz. Each time the counter crosses zero, the following procedure is executed:

1) The comparator (PWM) output is set to high (if PWMDCFG = 0, the output is not set)

2) The PWM duty-cycle value from register PWMDCFG is latched to the shadow register (see Figure 3.15).

The PWM output stays high as long as the counter value is less than the value configured in register PWMDCFG. When counter's value matches the value of PWMDCFG, the comparator (PWM) output is cleared. This is the end of the PWM duty cycle. The PWM duty cycle is calculated using the following formula:

$$PWMdc = \frac{[PWMDCFG]}{256} \times 100, [\%] \rightarrow PWMDCFG \in [0:255]$$

A duty-cycle of 50% is achieved by writing the value 128_{DEC} in register PWMDCFG. Writing 0 to PWMDCFG produces a continuous low level at the output of PWM module. The maximum PWMDCFG value 255_{DEC} produces a duty-cycle of 99.6%.

Note: In order to have a continuously enabled driver (i.e. duty-cycle of 100%), the MCU should clear the corresponding PWM enable of register PWMDREN.

3.10.3. Drivers PWM Control

The PWM signal is propagated to all half-bridge and high-side drivers. Each driver has a PWM enable bit that determines whether the driver's output is pulse-width modulated or not. The PWM enable bits for all drivers are mapped to register PWMDREN (see Table 3.25 and Table 3.26). If needed, the MCU can enable simultaneous PWM control for two or more drivers by setting more PWM enable bits in register PWMDREN.

Once the PWM is enabled (PWMDCFG ≠ 0), the write access to register PWMDREN is disabled. In order to enable the PWM control for another driver, the MCU must disable the PWM generation (write PWMDCFG = 00_{HEX}) and then modify the value of the PWMDREN register. For a successful PWMDREN change, the following sequence is recommended:

1) Disable the PWM by writing to PWMDCFG = 00_{HEX}
2) Wait for a time longer than one PWM period to ensure that register PWMDCFG is updated; i.e. PWM is disabled. This requirement comes from the ZAMC4100 PWM glitch-free feature (see section 3.10.1).
3) Write the new value in register PWMDREN
4) Enable the PWM again (PWMDCFG ≠ 0x00)

Table 3.25 PWMDREN Register Bits Mapping

Name	**PWMDREN**							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	HS4PWMEN	HS3PWMEN	HS2PWMEN	HS1PWMEN	HB4PWMEN	HB3PWMEN	HB2PWMEN	HB1PWMEN
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	**0x1B**							
R/W = Read/Write bit								

Table 3.26 PWMDREN Register Bits Description

Bit	**Description**
7:4	**HSxPWMEN:** [1), 2)] Enables PWM control for high-side drivers x (x = 1 to 4). 1 = The output of the corresponding high-side driver is pulse-width modulated. 0 = There is no pulse width modulation at the driver's output.
3:0	**HBxPWMEN:** [1), 2)] Enables PWM control for half-bridge drivers x (x = 1 to 4). 1 = The output of the corresponding half/bridge driver is pulse-width modulated. 0 = There is no pulse width modulation at the driver's output.
1) Bits HSxPWMEN and HBxPWMEN are automatically cleared at SLEEP or STANDBY Mode entry. 2) When all bits of the PWMDREN register are 0, the PWM module is disabled.	

3.10.4. PWM Initialization

This section describes the recommended PWM initialization flow which ensures no latency between the SPI driver enable and driver's output response. As an example, the initialization process using PWM control of the driver HS3 is illustrated. In order to have pulse-width modulation at the HS3 pin, the MCU should execute the following procedure (see Figure 3.17):

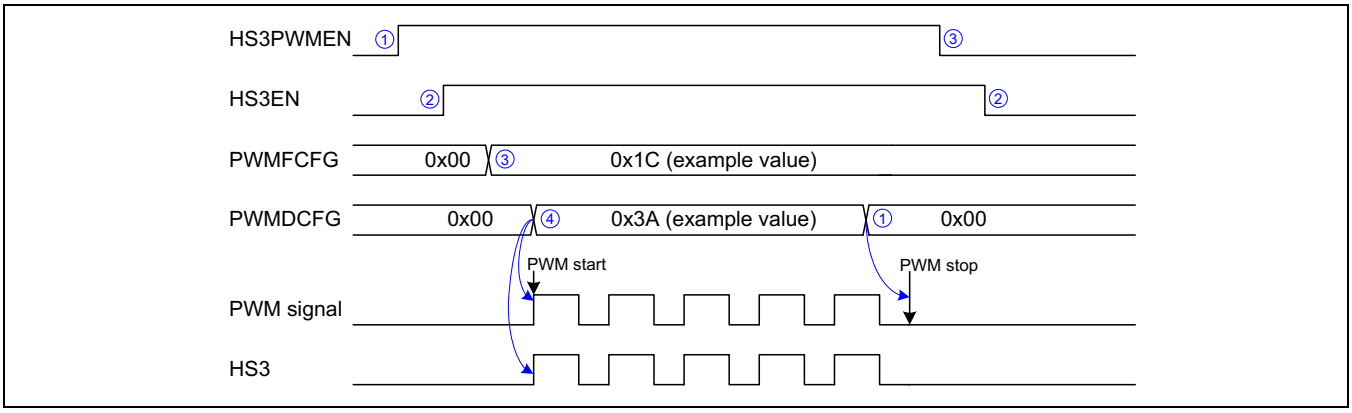
1) Enable PWM control for the HS3 driver by setting the HS3PWMEN bit of the PWMDREN register (see Table 3.25).
2) Enable driver HS3 by setting the HS3EN bit of the HSDCTRL register (Table 3.20).
3) Configure the PWM frequency by writing to register PWMFCFG (Table 3.50).
4) Configure the PWM duty cycle by writing in register PWMDCFG (Table 3.50) a value greater than zero.

Until step 4 is executed, the driver's output is disabled. After configuring the duty-cycle (in step 4), the PWM module starts operating; i.e., it starts modulating the HS3 output.

In order to disable the PWM at the driver's output, the MCU should execute the following (recommended) procedure (see Figure 3.17):

1) Write zero in register PWMDCFG in order to stop PWM generation. As shown in Figure 3.17, the PWM stops after the end of the current period. The value in the PWMFCFG register can be left unchanged.
2) Disable the driver by clearing bit HS3EN of register HSDCTRL.
3) Disable the PWM for driver HS3 by clearing the HS3PWMEN bit. If step 2 is omitted (bit HS3EN still set), the execution of this step will enable the driver.

Figure 3.17 Recommended PWM Initialization Procedure with HS3 as an Example

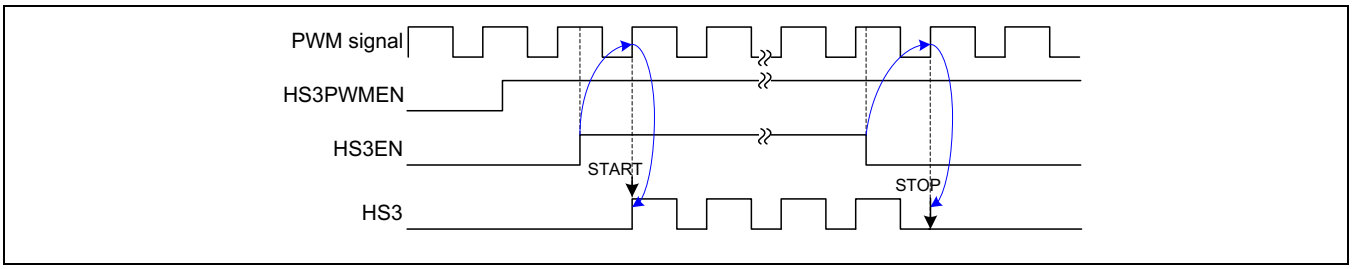


The initialization flow explained above should be considered as a recommendation for ensuring that the application has no latency between setting the enable bits and driver's output reaction.

The ZAMC4100 has internal synchronization logic that prevents glitches at each driver's output in case the driver's enable bit is set when the generation of PWM signal is already running. In this event, the driver enable will be synchronized with the PWM signal, which will cause a latency of one PWM period (worst case). See Figure 3.18 for an illustration of the synchronization effect.

© 2019 Renesas Electronics Corporation

Figure 3.18 Effect of Driver Enable When the PWM is Running



3.10.5. PWM Module in SLEEP and STANDBY Mode

When SLEEP or STANDBY Mode is activated, the PWM signal generation is automatically disabled by clearing the value of register PWMDCFG (i.e., zero the duty-cycle). The drivers' PWM enable bits in register PWMDREN are also cleared. If needed after wake-up, the MCU can reinitialize the PWM unit following the procedure explained in the previous section.

As a recommended practice, before entering the SLEEP or STANDBY Mode, the MCU should stop the PWM in order to ensure that the drivers will not be disabled during the PWM pulse generation phase; i.e. no glitches at the drivers' outputs.

3.11. EC Mirror Driver

The structure of the EC mirror (ECM) driver is shown in Figure 3.19. It is controlled via register ECMCTRL (see Table 3.27). Setting the ECMEN bit in this register enables the driver while the output voltage is controlled by a 6-bit DAC programmed by the ECMDAC bits.

The driver operates as the DAC voltage is continuously compared with the voltage measured at the EC_M pin by the error amplifier. Depending on the comparison result, it controls both transistors of the output stage to provide a current with a direction and value necessary for achieving the desired voltage across the electrochromatic mirror terminals. The reflectivity of the mirror is controlled precisely with respect to the desired level (set by bits ECMDAC[5:0] in the ECMCTRL register).

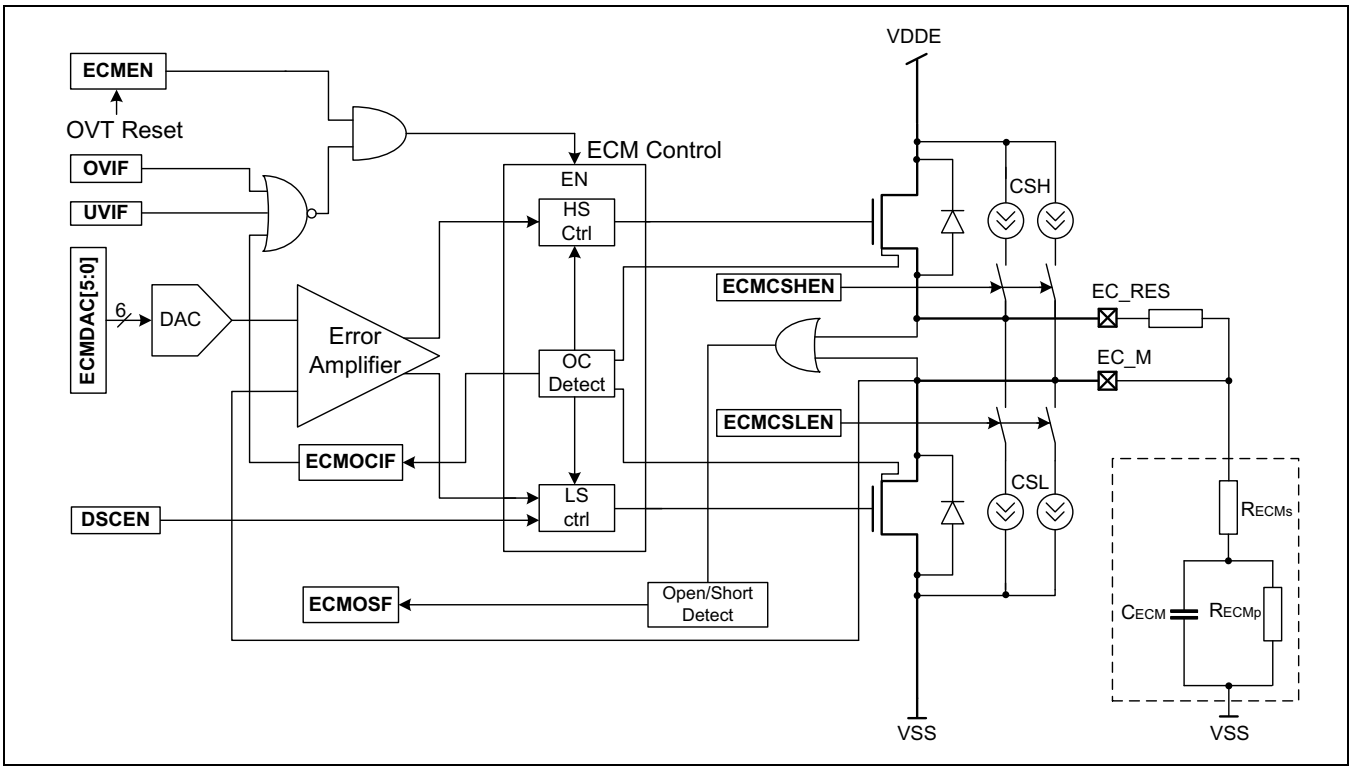
In order to limit the current through the load and to reduce the power dissipation of the driver inside the chip, an external resistor is connected in series between the high-side transistor of the output stage and the EC mirror. It must withstand power in the range of 0.5W to 1W depending on the nominal supply voltage, the static current consumption, and the expected dynamic current consumption.

For achieving maximum reflectivity of the mirror in the fastest possible way, it is necessary that the voltage across the electrochromatic material (C_{ECM}) to be reduced to zero. This is achieved by completely turning on the low-side transistor of the output stage of the ECM driver, which shorts the terminals of the mirror and provides a low resistance current path for discharging. If the equivalent resistance of that switch is low (see Table 1.9), the time necessary for that transition is determined mainly by the parameters of the EC mirror (C_{ECM}, R_{ECMs}, R_{ECMp}) and the initial voltage to which it has been charged. This feature is enabled by the dedicated bit DSCEN in the ECMCTRL register.

Like all other power drivers in the ZAMC4100, the ECM driver is automatically disabled in the event of an over/under voltage event or overheating of the system.

© 2019 Renesas Electronics Corporation

Figure 3.19 ECM Driver Structure



3.11.1. ECM Driver Current/Voltage Limitation and Protection against Short to GND or Vbat

The limitation of the current through the load to safe levels for all conditions is achieved by adding an external resistor between the EC_RES pin and the mirror terminal. Its value is chosen considering the characteristics of the EC mirror and the nominal supply voltage of the system (see Table 6.1 for recommended external components). In addition, the maximum voltage at the output of the driver is limited to the level specified in Table 1.9. The limitation of this voltage relies on the feedback connection between the mirror terminal and the EC_M pin. If this connection is missing, the voltage applied by the driver to its load could rise up to the supply of the system, so an important step before activating the driver is to perform an open/short diagnostic check (see 3.11.3).

The current through both transistors (high-side and low-side) of the output stage is sensed for protecting the driver against a short to Vbat or GND of both output pins (EC_RES and EC_M). In the event of such failures, if the current exceeds a specified threshold level (see Table 1.9), the output stage of the ECM driver is automatically disabled. This event generates an interrupt to the MCU and the ECMOCIF flag bit in the ECMDIAG register is set (the ECMOCIF bit in IRQSTAT register is set; see

Table 3.9). Recovering from that state to normal functionality requires that the ECMOCIF flag to be cleared by writing “1” to its position in the ECMDIAG register. The procedure for recovering from that state to normal functionality goes through the following steps:

As previously discussed, it is strongly recommended that the MCU performs an open/short diagnostic check before re-enabling the driver:

- If the result is “Normal load or short to VSS” (see section 3.11.3), then “Normal load” condition is assumed and the MCU should proceed with the driver enable. If enabling the driver generates an over-current event, then there is a “short to VSS” failure at the output.
- If the diagnostic (and eventual activating of the driver as described above) results in “short to VSS,” “Short to VDDE,” or “Open load,” the ECM driver should not be enabled until the failure is present.

ECM “Short to VSS” and “Short to VDDE” Failures

For ZAMC4100 failures for the ECM, “Short to VSS” and “Short to VDDE” must be considered differently on the module level compared to the ZAMC4100 level. On the module level, the short failures are defined for each point in the outside circuitry, including external connections, components and loads, and reflect their specifics. On the ZAMC4100 level, failure is defined inside the IC, also including the pins on the package.

This clarification is important because module-level failures are not always IC-level failures. For example in the case of a “Short to VSS” of the EC mirror terminal (module-level failure), the current through the output of the ECM driver is limited by the external resistor R1 (see the application diagram in Figure 6.1) to a value in the normal operation range for the ZAMC4100 (see parameters $I_{OC\_EC\_M}$ and $I_{OC\_EC\_RES}$ in Table 1.9). As a result, this particular event will not be detected by the internal over-current protection circuitry of the ECM driver and no over-current event will be detected and captured in the ECMOCIF interrupt flag (see section 3.11.3,

© 2019 Renesas Electronics Corporation

Table 3.9, and Table 3.1).

To cover all possible failure scenarios, the ZAMC4100 features different resources for detecting short failures on the module level and on the IC level. Failures on the IC level are detected using internal diagnostic current sources and over-current protection, while detecting failures on the module level also requires the dedicated ADC channel for direct measuring of the EC_M pin voltage (see Table 3.36). Because the EC_M pin is used for sensing the potential at the mirror terminal, it gives precise information for the overall status and current operating condition of the ECM functionality. Any failure on the IC or module level will result in an unexpected or inaccurate response at the EC_M potential with respect to the configuration in the ECMCTRL register (see section 3.11.2).

The ADC ECM voltage channel together with the “open/short diagnostic check” (see 3.11.3) provides all the necessary information for the MCU firmware to distinguish between different types of ECM short failures.

3.11.2. ECM Control Register

Table 3.27 ECMCTRL Register Bits Mapping

Name	**ECMCTRL**							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	ECMEN	DSCEN	ECMDAC[5]	ECMDAC[4]	ECMDAC[3]	ECMDAC[2]	ECMDAC[1]	ECMDAC[0]
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	0x09							
R/W = Read/Write bit								

Table 3.28 ECMCTRL Register Bits Description

Bit	**Description**
7	**ECMEN:** [1) 2) 3)] Electrochromatic mirror driver enable bit. 1 = ECM driver is enabled. 0 = ECM driver is disabled.
6	**DSCEN:** [1) 2)] Electrochromatic mirror discharge enable bit. 1 = ECM driver low-side MOSFET is ON; EC mirror discharge enabled. 0 = ECM driver low-side MOSFET is OFF; EC mirror discharge disabled.
5:0	**ECMDAC[5:0]:** [1) 2)] Electrochromatic control DAC bits. 111111 = ECM driver provides maximum output voltage (see Table 1.9). 000000 = ECM driver provides minimum output voltage (see Table 1.9).

1) These bits are automatically cleared on an over-temperature event.
2) These bits are automatically cleared upon SLEEP or STANDBY Mode entry.
3) This bit is gated in case a failure event (OC, OV, UV) is captured in an interrupt flag (see Figure 3.19).

3.11.3. ECM Driver Output Diagnostic Check

The ECM driver diagnostic circuitry is able to detect open/short failures in the driver output. The diagnostic functionality is controlled and monitored by register ECMDIAG (Table 3.29 and Table 3.30).

Table 3.31 gives the meaning of the ECMOSF bit in the ECMDIAG register with respect to the combination of ECMCSHEN and ECMCSLEN bits that enable the diagnostic current sources/sinks connected to the EC_M and AC_RES pins respectively.

Table 3.29 ECMDIAG Register Bits Mapping

Name	**ECMDIAG** [1]							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name					ECMCSHEN	ECMCSLEN	ECMOSF	ECMOCIF
Reset	U	U	U	U	0	0	0	0
Access	U	U	U	U	R/W	R/W	R	R/Wc
Address	0x0C							
R/W = Read/Write; **R =** Read-only bit; **U** = Unimplemented, read as '0'; **R/Wc =** Read/Write 1 to clear								
1) The ECMOCIF is physically implemented in register ECMDIAG and mapped to IRQSTAT (Table 3.10).								

Table 3.30 ECMDIAG Register Bits Description

Bit	Description
7:4	Unimplemented bits. Read as 0.
3	**ECMCSHEN**[(1)]**:** ECM high-side current source enable bit. 1 = High-side current source is enabled. 0 = High-side current source is disabled.
2	**ECMCSLEN**[(1)]**:** ECM low-side current source enable bit. 1 = Low-side current source is enabled. 0 = Low-side current source is disabled.
1	**ECMSOF:** Open/short status flag of the ECM driver. This flag is not registered; i.e., its value remains as long as the diagnostic event takes place (see Table 3.31).

Bit	Description
0	**ECMOCIF** [(2)]**:** Over-current interrupt flag of the ECM driver. 1 = Over-current event has been detected at the ECM output, and the driver has been disabled. 0 = There is no over-current at ECM output.

1) To keep current consumption as low as possible, the MCU software should disable the current sources before enabling SLEEP or STANDBY Modes.
2) This bit is also mapped to IRQSTAT register.

Table 3.31 ECMOSF Diagnostic Bit Meaning

ECMCSHEN	ECMCSLEN	ECMOSF [1)]	Description
0	0	x	Diagnostic disabled
1	1	x	Result not defined
1	0	0	Normal load or short to VSS [2)]
0	1	0	
1	0	1	Short to VDDE
0	1	1	
1	0	1	Open load
0	1	0	
1	0	0	Unused
0	1	1	

1) **Important:** The diagnostic result is correct only if the driver is disabled.
2) In the event of a short to VSS, the output stage of the driver will be automatically disabled if the current exceeds the threshold specified in Table 1.9. If the current is limited to levels below that threshold due to specifics of the EC mirror glass used, distinguishing between the two possible states, which are “Normal load” or “Short to VSS,” can be done by an additional ADC channel and software procedure for measuring the potential at EC_M pin (for more details, see section 3.11.1).

3.12. ZAMC4100 Analog to Digital Converter

The ZAMC4100 features a 10-bit successive approximation ADC with a front-end multiplexer for digitizing the multi-channel inputs, including four external ports.

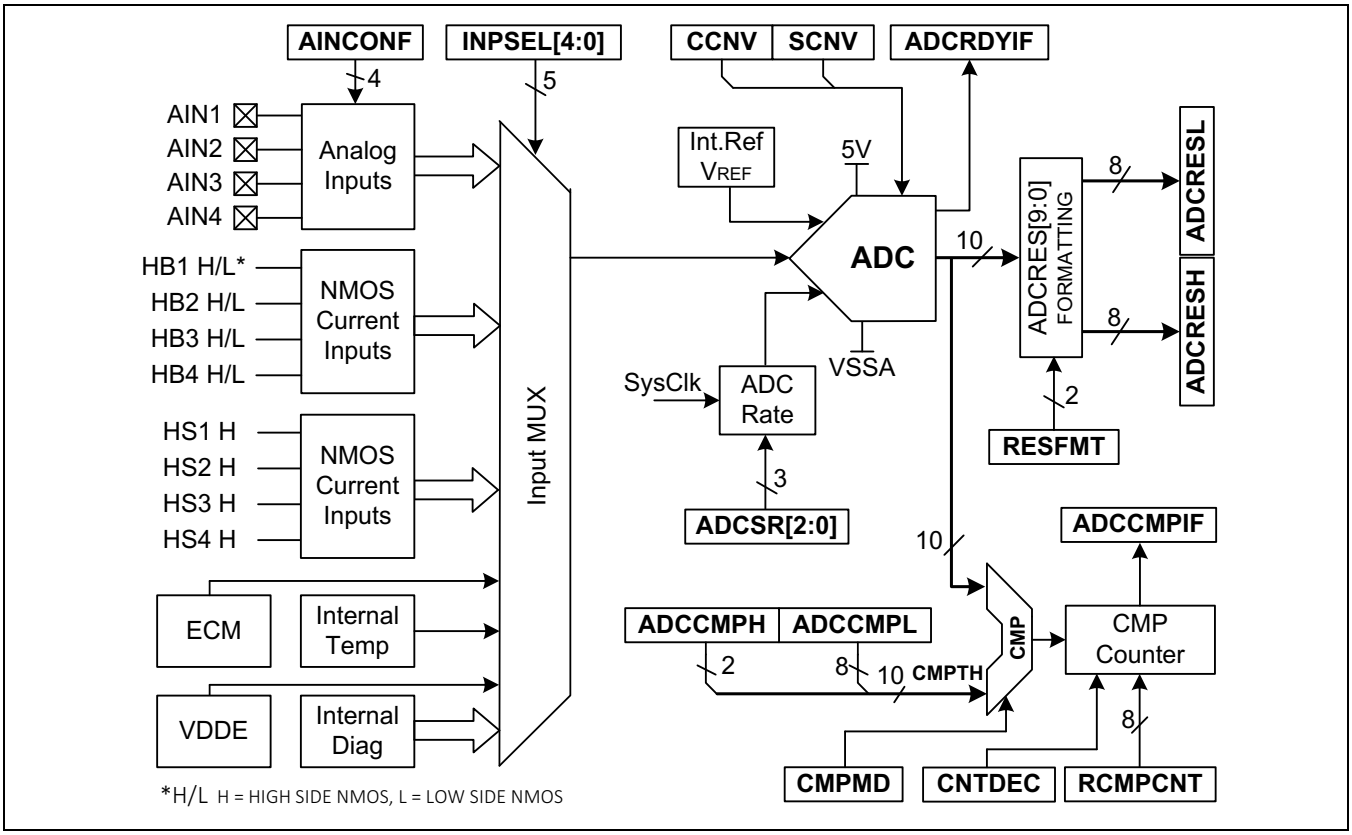
3.12.1. ADC Overview

The ZAMC4100 ADC block has the following key features:

- SAR (successive approximation) conversion, 10-bit resolution, single or continuous conversion
- High conversion speed (see Table 1.15)
- Multi-channel front-end multiplexer with access to various input signals as shown in the Figure 3.20; 4 external channel ports with versatile options
- Internal voltage and current references
- Configurable result format
- Result comparator with counter
- Interrupt support for various type of events

Figure 3.20 illustrates the block diagram of the ADC module with the result comparator at the bottom of the diagram.

Figure 3.20 ADC Functional Diagram



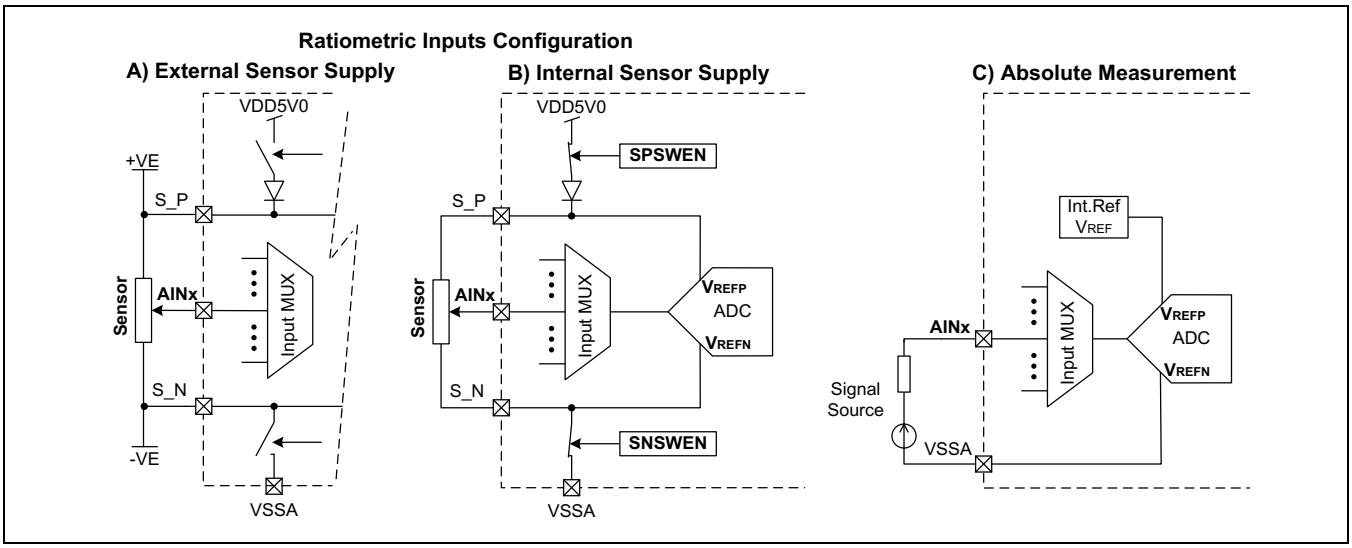
This module is highly versatile due to a comprehensive register structure as summarized below:

- ADC External Port Configuration Register — AINICONF
- ADC Multiplexer and Control Register — ADCCTRL
- ADC Configuration Register — ADCCONF
- ADC Status Register with Interrupts Flags — ADCSTAT
- ADC Compare and Counter Registers — RESCMPH, RESSCMPL, RCMPCNT
- ADC Digitized and Formatted Results — ADCRESH and ADCRESL

3.12.2. ADC External Inputs Configuration

The external analog inputs AIN1-AIN4 of the ADC can be configured for measurement in ratiometric or absolute mode or for temperature measurement using internal current sources. All AINx inputs are configured via settings in the AINCONF register (see Table 3.32 and Table 3.33).

Figure 3.21 Ratiometric versus Absolute Measurement Inputs Configuration



Ratiometric Measurement on AIN1-AIN4

By default, the analog input pins are configured for ratiometric measurements with the external supply source as shown in the Figure 3.21-A. The ADC reference inputs are connected to the *S_P* or *S_N* pins, which “sense” the sensor supply voltage and ensure accurate ratiometric conversion results.

The supply switches SPSWEN and SNSWEN are controlled by the AINCONF register. This register will determine the usage of the external or internal supply source for the sensor and the ADC reference as shown in Figure 3.21-A and Figure 3.21-B. In general, when the external sensor supply source is used, the measurement can provide more accurate results because the voltage drop across the connections to the S_P and S_N pins is negligible due to the very low current through them.

If the signed ADC result mode is used, the sign is determined as positive if the measured voltage is in the upper half of the voltage difference between *S_P* and *S_N* pins and negative if in the lower half.

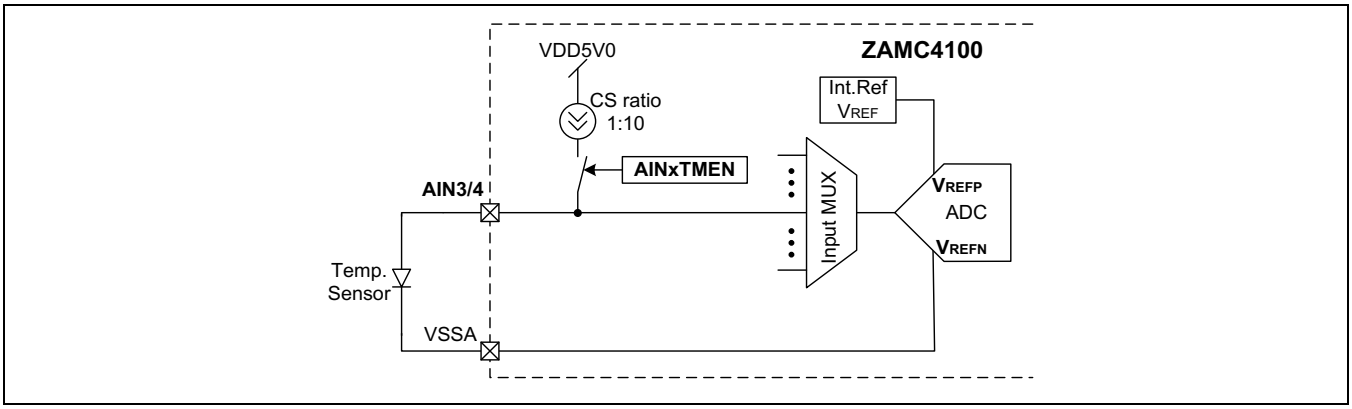
Absolute Measurement on AIN1-AIN4

Absolute measurement as shown in Figure 3.21-C can be achieved by setting the AINxMD bits in the AINCONF register (where x = channel 1 to 4). During the absolute measurement, the ADC uses its internal reference and the sensor absolute voltage is measured at the corresponding AINx pin.

Diode Temperature Measurement on AIN3, AIN4

If the application requires external temperature measurements using a diode-based sensor, it can be connected to input AIN3 or AIN4 (Figure 3.22). Both inputs have an internal current source that is enabled with bits AIN3TMEN and AIN4TMEN in register AINCONF (Table 3.32). The current sources are internally controlled by the ADC in order to provide two currents with a fixed ratio 1:10. This ratio is used for the calculation of the temperature based on the measured results.

Figure 3.22 Using AIN3 and AIN4 for External Temperature Measurement



For more information about the temperature measurement including the LM135 and thermistors, refer to the *ZAMC4100 Application Note – ADC.*

Table 3.32 AINCONF Register Bits Mapping

Name	AINCONF							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name	SPSWEN	SNSWEN	AIN4TMEN	AIN3TMEN	AIN4MD	AIN3MD	AIN2MD	AIN1MD
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	0x10							

R/W = Read/Write bit

Note: When the ADC conversion is running, the AINCONF register is accessed as read-only.

Table 3.33 AINCONF Register Bits Description

Bit	Description
7	**SPSWEN:** Bit for connecting the internal 5.0V supply to the S_P pin. 1 = Pin S_P is connected to the internal supply. 0 = Pin S_P is connected to an external source.
6	**SNSWEN:** Bit for connecting the VSSA to pin S_N. 1 = Pin S_N is connected to VSSA. 0 = Pin S_N is connected to an external source.
5:4	**AINxTMEN:** Enable external temperature measurement at pins AIN3, AIN4. 1 = Pin AINx (x=3, 4) is configured for temperature measurement. 0 = Pin AINx (x=3, 4) is used as a general-purpose pin. **Note**: When AINxTMEN = 1, the bits AINxMD are “don’t care.”
3:0	**AINxMD:** Analog inputs absolute or ratiometric configuration bit. 1 = Pin AINx (x=1 to 4) is configured for absolute measurement. 0 = Pin AINx (x=1 to 4) is configured for ratiometric measurement.

3.12.3. ADC Input MUX Control

The ADC input multiplexer is controlled via the INPSEL[4:0] bits in the ADCCTRL register (see Table 3.34). The settings for combinations of INPSEL bits are described in Table 3.36 and illustrated in Figure 3.20Figure 3.21. After setting the INPSEL for a corresponding channel, the ADC control circuitry automatically processes the following operations:

1) It configures the analog circuitry to provide required reference voltage.
2) When the conversion is started, it runs the conversion algorithm depending on the type of selected input.

Bits SCNV and CCNV of register ADCCTRL are used for starting a single or continuous ADC conversion. Supported ADC conversion modes and conversion enable bits are discussed in section 3.12.7.

Note: When an ADC conversion is running, the WRITE access for the INPSEL bits is blocked.

Table 3.34 ADCCTRL Register Bits Mapping

Name	ADCCTRL							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name		CCNV	SCNV	INPSEL[4]	INPSEL[3]	INPSEL[2]	INPSEL[1]	INPSEL[0]
Reset	U	0	0	0	0	0	0	0
Access	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	**0x12**							

R/W = Read/Write bit; U = Unimplemented, read as '0.'

Note: When the ADC conversion is running, bits INPSEL[4:0] are accessed as read-only.

Table 3.35 ADCCTRL Register Bits Description

Bit	Description
7	Unimplemented bit. Read as 0.
6	**CCNV:** Continuous Conversion Mode. 1 = ADC performs Continuous Conversion Mode. 0 = ADC idle.
5	**SCNV:** Single Conversion Mode. 1 = ADC performs Single Conversion Mode. Cleared automatically at completion of ADC conversion. 0 = ADC idle.
4:0	**INPSEL:** Input MUX Select Channels. See Table 3.36.

Note: When CCNV and SCNV are both set, the ADC performs in the Continuous Conversion Mode.

© 2019 Renesas Electronics Corporation

Table 3.36 Input MUX Control Bits Description

INPSEL[4:0] value	Channel Number [DEC]	Selected Channel Name	Notes
Application mode ADC channels			
00000	0	AIN1	The type of measurement depends on the value of the AICONF register.
00001	1	AIN2	
00010	2	AIN3	
00011	3	AIN4	
00100	4	HB1H	Current measurement mode. Measures the current through the high-side/low-side transistor of the half-bridge drivers 1-4.
00101	5	HB1L	
00110	6	HB2H	
00111	7	HB2L	
01000	8	HB3H	
01001	9	HB3L	
01010	10	HB4H	
01011	11	HB4L	
01100	12	HS1	Current measurement mode. Measures the current through high-side drivers 1-4.
01101	13	HS2	
01110	14	HS3	
01111	15	HS4	
10000	16	VDDE	Supply voltage at VDDE pin.
10001	17	INTTMP	Internal chip temperature.
10010	18	ECM	Measures EC_M pin voltage.
[10011 to 11000]	19-24	-	Reserved.
Diagnostic mode ADC channels (see section 5.2)			
11001	25	Internal reference	Ratiometric ADC diagnostic
11010	26	Internal reference	Ratiometric ADC diagnostic
11011	27	Internal reference	Absolute voltage measurement
11100	28	ADC VREF	ADC reference voltage diagnostic
11101	29	REG2v5 regulator voltage	Internal VDD2V5 supply diagnostic
11110	30	REG5v0 regulator voltage	Internal VDD5V0 supply diagnostic
11111	31	Current multiplexer	Absolute current measurement diagnostic

© 2019 Renesas Electronics Corporation

3.12.4. ADC Voltage References and Result Calculation

The ZAMC4100 ADC features different reference sources depending on the configured measurement mode. The reference voltage is automatically selected depending on the INPSEL setting and the analog inputs configuration (AINCONF register; Table 3.32). These settings also configure the result interpretation as described below.

Note: In the following descriptions of the measurement options, the coefficients $CSR_{\_X}$, I_{REF}, V_{REF} and V_{t27} can vary from chip to chip. If more precise calculations are needed their factory-trimmed values can be obtained from the MCU FLASH memory (refer to section 4.2.3); otherwise the default values from Table 1.8, Table 1.11, and Table 1.12 are used.

Ratiometric measurement

In the ratiometric measurement mode, the reference voltage for the ADC is the voltage between the *S_P* and *S_N* sensor pins. The result of the ADC conversion is calculated as the *RATIO* of the voltage between the *S_P*, *S_N* pins and the voltage at input AINx:

$$RESCODE = \frac{RATIO * \left(2^{10} - 1\right)}{100\%}$$

Depending on the application requirements, the result format can be configured as signed or unsigned. For example, the signed ratiometric result can be used for position adjusting while the unsigned result format is convenient for level measurement.

Absolute voltage measurement

The absolute voltage measurement mode is used for analog inputs measurement, for system supply measurement (the voltage at VDDE pin), and for the EC_M pin diagnostic measurement (see Table 3.36). In this case, the reference voltage for the ADC is provided by the internal reference source (V_{REF}) with the value that is specified in Table 1.12. The formula used for result calculation is

$$RESCODE = \frac{AINx[V] * 2^{10}}{V_{REF}[V]}$$

For VDDE measurements, the equation is:

$$RESCODE = \frac{VDDE[V] * 2^{10}}{8 * V_{REF}[V]}$$

Absolute current measurement

This measurement is for internal channels associated with half-bridge and high-side drivers. The equation below represents a current measurement in RESCODE (unsigned) after internal translation using two constants:

$CSR_{\_X}$ = Current Sense Ratio Coefficient (specified for each high-side and half bridge driver in Table 1.8)

I_{REF} = Calibrated internal reference current as specified in Table 1.12 (typically 10µA)

$$RESCODE = \frac{I_{IN}[\mu A] * 2^{10}}{CSR_{\_X} * I_{REF}[\mu A]}$$

Where I_{IN} = Input current up to full-scale range equal to ($CSR_{\_X} * I_{REF}$)

© 2019 Renesas Electronics Corporation

External temperature measurement

For a basic understanding of external temperature measurement for the diode-based sensor, the equation below provides a temperature measurement result (unsigned) within RESCODE (other additional factors apply in final applications; contact IDT for more details using the contact information on page 155):

$$RESCODE = 32 * T[K] * \frac{k * \ln(10)}{q} * \frac{2^{10}}{V_{REF}[V]}$$

$$RESCODE = \frac{T[K]}{0.1538 * V_{REF}[V]}$$

Where: q = elementary charge ($1.60217646 \times 10^{-19}$)

k = Boltzmann constant ($1.3806503 \times 10^{-23}$)

Internal temperature measurement

The formula for temperature calculation in correspondence with the conversion unsigned result is

$$T[°C] = \left(\frac{V_{REF}[V] * 10^3}{3072 * S_{VT}[mV/°C]} (512 + RESCODE) \right) - \left(\frac{V_{t27}[V] * 10^3}{S_{VT}[mV/°C]} \right) + 27$$

Where the parameter S_{VT} is the voltage/temperature slope of the internal temperature sensor and V_{t27} is the output voltage at 27°C (both specified in Table 1.11).

3.12.5. ADC Configuration and ADC Status Register

ZAMC4100 ADC configurations define the conversion rate, result format (sign and justification) and continuous conversion mode in the ADCCONF register. The mapping of bits in this register is described in Table 3.37 and the purpose of the bits is described in the following sections.

The interrupt flags of the ADC module are allocated in the ADCSTAT register described in Table 3.39.

Table 3.37 ADCCONF Register Bits Mapping

Name	ADCCONF							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name			RESJSTF	RESSIGN	CCNVMD	ADCSR[2]	ADCSR[1]	ADCSR[0]
Reset	U	U	0	0	0	0	0	0
Access	U	U	R/W	R/W	R/W	R/W	R/W	R/W
Address	0x11							

R/W = Read/Write bit; U = Unimplemented, read as '0.'
Note: When the ADC conversion is running, the ADCCONF register is accessed as read-only.

Table 3.38 ADCCONF Register Bits Description

Bit	Description
7:6	Unimplemented bits. Read as 0.
5	**RESJSTF:** Result register justification format. Details provided in section 3.12.8. 1 = Right-justified result in ADCRESx. 0 = Left-justified result in ADCRESx.
4	**RESSIGN:** Result register in signed or unsigned format. 1 = Signed value (RESCODE = -512 to +511 full-scale range). 0 = Unsigned value (RESCODE = 0 to 1023 full-scale range).
3	**CCNVMD:** Continuous conversion versus ADC result register read mode. 1 = After ADC sample, halt next ADC sample operation until ADCRESx register is read. 0 = After ADC sample, proceed to next ADC sample operation regardless of whether the ADCRESx register has been read or not. This means that the ADCRESx register will be overwritten by the next conversion result.
2:0	**ADCSR:** Sample rate configuration (see Table 3.41).

Table 3.39 ADCSTAT Register Bits Mapping

Name	ADCSTAT							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name							ADCCMPIF	ADCRDYIF
Reset	U	U	U	U	U	U	0	0
Access	U	U	U	U	U	U	R/Wc	R/Wc
Address	**0x0D**							

R/Wc = Read/Write 1 to clear, U = Unimplemented, read as '0'

Note: When the comparator is enabled (bit CMPEN = '1'), the ADCRDYIF flag bit is automatically masked. As a result, the global ADC interrupt (see Figure 3.8) is not generated at the end of each conversion cycle; it is generated only when a valid comparator event is detected.

Table 3.40 ADCSTAT Register Bits Description

Bit	Description
7:2	Unimplemented bits. Read as 0.
1	**ADCCMPIF:** ADC comparator counter interrupt flag. 1 = Compare counter (CMP) matches the value programmed in reference RCMPCNT register (Table 3.50). 0 = There is no result matching detected by comparator counter.
0	**ADCRDYIF:** ADC ready interrupt flag. 1 = Current ADC conversion has finished and ADCRESx (Table 3.50) contains results ready to be read by the MCU. 0 = ADC conversion has not finished and/or the result has not been read from the ADCRESX register. **Note:** If CMPEN is set, this flag is masked; see Table 3.42.

3.12.6. ADC Sample Rate

The ADC sampling rate is calculated as the maximum sampling rate (250Ksps) divided by the value configured in bits ADCSR[2:0] of register ADCCONF (Table 3.37). Table 3.41 gives the definitions of the ADCSR[2:0] bits. t_{CCNV} is the ADC continuous conversion phase. For more information on this and other ADC timing parameters, refer to section 1.3.2.

The time between two consecutive conversion results is calculated as the sum of the following:

- ADC sample time (see Table 1.15 and Figure 1.1): configurable parameter (via ADCSR[2:0] bits) for all modes except the external temperature conversion for which it is fixed.
- ADC conversion time (see Table 1.15 and Figure 1.1): fixed parameter which depends only on the OSCH clock accuracy.
- ADC idle time (see Table 1.15 and Figure 1.1): configurable parameter (via ADCSR[2:0] bits) which determines the sample rate for the external temperature conversion.

Each ADC start conversion command (single or continuous) counts a 10µs period needed for proper settling of the ADC input stage.

Table 3.41 ADC Sample-Rate Configuration

ADCSR[2:0]	Sample Rate [1), 2)]	t_{CCNV}
000	SRmax	4µs
001	SRmax/2	8µs
010	SRmax/4	16µs
011	**SRmax/8**	32µs
100	SRmax/16	64µs
101	SRmax/32	128µs
110	SRmax/64	256µs
111	SRmax/128	512µs

1) The value of SRmax is specified in Table 1.15. The t_{CCNV} (ADCSR=0) is calculated based on 1/SRmax = 1/250Ksps.

2) The maximum possible sample rate for the external temperature conversion is **SRMAX/8**. For ADCSR[2:0] values in the range from 000 to 011, the external temperature conversion is running at SRmax/8

3.12.7. ADC Single and Continuous Conversion

Single or continuous conversion operation is defined by the SCNV and CCNV bits respectively, in the ADCCTRL register (Table 3.34). If this bit is set, the ADC will automatically start the conversion process. It sets the interrupt flag ADCRDYIF in the ADCSTAT register (Table 3.39) when the conversion process is completed with the results (after the formatting process) stored in registers ADCRESH and ADCRESL (Table 3.50). This interrupt flag is cleared after reading the result registers. If both of the SCNV and CCNV bits are set, the ADC operates in continuous conversion mode.

The output of the ADC goes to the results formatting block, where there are four options for result representation as detailed in section 3.12.8.

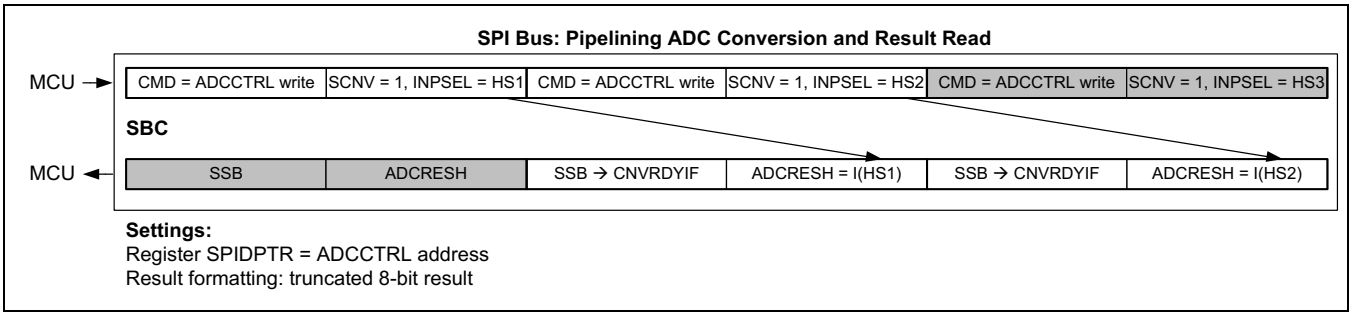
© 2019 Renesas Electronics Corporation

Single ADC Conversion and SPI Pipeline Concept

The MCU starts a single ADC conversion by setting bit SCNV=1. When the conversion finishes, bit SCNV is automatically cleared and the interrupt flag ADCRDYIF is set.

Since the SCNV and INPSEL bits share a register (ADCCTRL), the MCU starts a single conversion on only the selected channel with one SPI WRITE access. Using the pipelined SPI WRITE/READ feature of the ZAMC4100 SBC, the MCU can pipeline the channel's measurement and results monitoring. Figure 3.23 shows an example for SPI communication when the MCU is measuring the current of the high-side drivers. See Table 3.50 for details for the SPIDPRT register.

Figure 3.23 Pipelined Measurement and Results Reading for the Driver Current



ADC Continuous Conversion

With the CCNV bit set, the ADC operates in continuous conversion mode at the sampling rate determined by the ADCSR[2:0] bits. After each conversion with formatted data in the results register, the ADC generates the interrupt flag ADCRDYIF (Table 3.39). The CCNVMD bit provides two options affecting the way the result register is handled by the MCU:

Bit CCNVMD = 0 The ADC continuous conversion is running and the results registers ADCRESH and ADCRESL (Table 3.50) contain the latest conversion value. If the MCU does not read the value, it is overwritten with the next one.

Bit CCNVMD = 1 When the result is available, the ADC continuous conversion is halted until the MCU reads the result value. Reading the conversion result clears the ADCRDYIF flag and the ADC automatically continues with the next conversion. This mode allows the MCU to control the ADC sample rate by controlling the time between two SPI READ accesses.

Note: Before enabling continuous conversion with CCNVMD = 1, the MCU should clear the ADCRDYIF bit (by reading the result from previous conversions) in order to start the new ADC conversion.

3.12.8. ADC Result Formatting Options

While the SPI bus only handles 8-bit data transfers, the ADC result of 10 bits is split into two registers: ADCRESH and ADCRESL. Prior to this, the ADC digitized data are formatted in accordance with the settings for the bits RESJSTF and RESSIGN of register ADCCONF (Table 3.37). Bit RESJSTF configures left or right result justification as explained in the following sections.

Left-Justified ADC Results

This setting applies when bit RESJSTF = 0. The digitized data of bits [9:2] go to ADCRESH, while the remaining bits [1:0] go to ADCRESL as shown in Figure 3.24. For correct SPI extraction of the result, the ADRESH is read first, where it clears the interrupt flag ADCRDYIF as well as updating the value ADCRESL. This guarantees protection of the results from the next conversion.

Right-Justified ADC Results

This setting applies when bit RESJSTF = 1. The digitized data of bits [7:0] go to ADCRESL, while the remaining bits [9:8] go to ADCRESH as shown in Figure 3.24. For correct SPI extraction of the result, the ADRESL is read first, where it clears the interrupt flag ADCRDYIF as well as updating the value ADCRESH. This guarantees protection of results from the next conversion.

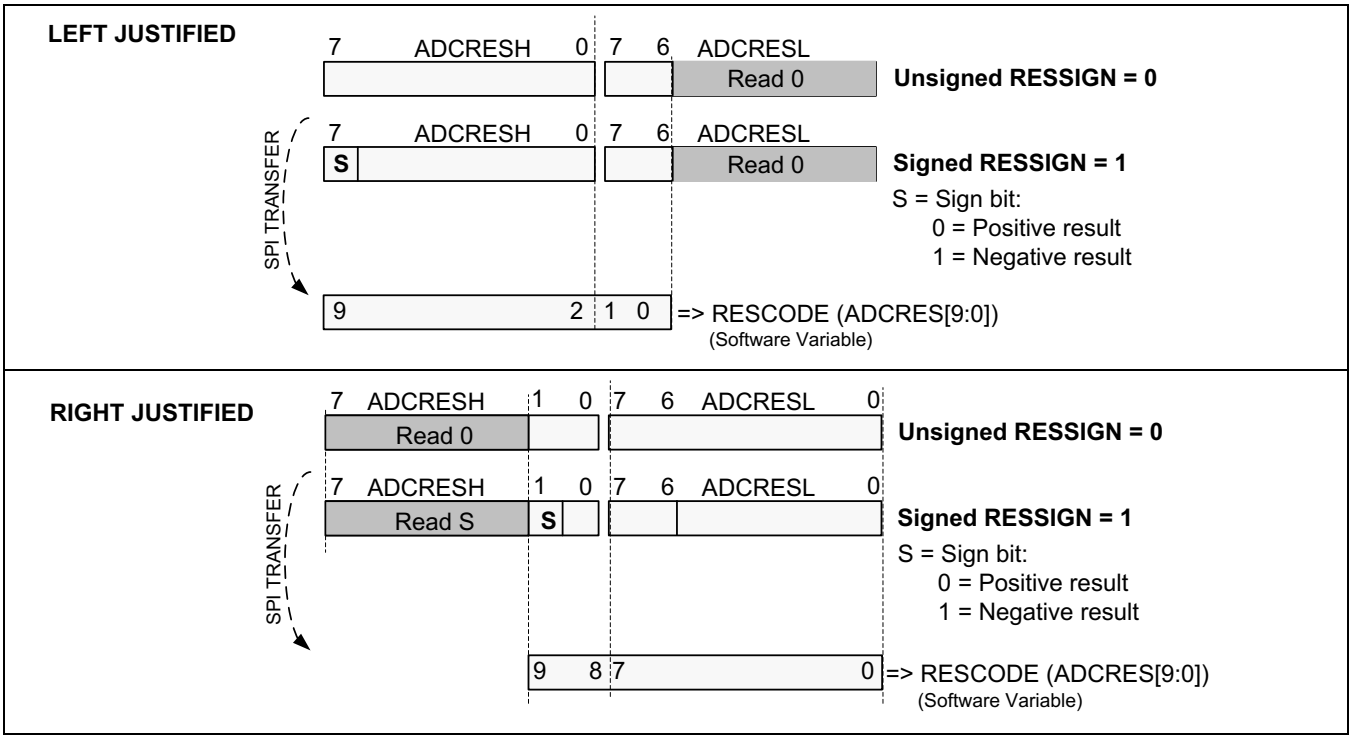
Signed or Unsigned ADC Results

Bit RESSIGN configures the signed or unsigned result format when the ADC is performing ratiometric conversion. When RESSIGN = 0, the ADC result is represented as unsigned value. When RESSIGN = 1, the ADC result is represented as signed value.

Note: If the ADC is performing absolute or temperature conversion, the result is unsigned regardless of the value of the RESSIGN bit.

Note: For the right-justified signed results format, the unused bits of ADCRESH have the value of the sign bit.

Figure 3.24 ADC Result Formatting Options



3.12.9. ADC Comparator and Counter

The ZAMC4100 features a 10-bit result comparator that allows continuous ADC running and *signed* result comparison within the SBC. After each conversion, the ADC comparator checks the result for being less than or greater than a pre-configured value. If the result of the comparison is TRUE, the comparator counter is incremented. If the result is FALSE, the counter value is decremented or cleared depending on the value of the CNTDEC bit in the ADCCMPH register (Table 3.42). When the comparator counter value reaches the value programmed in the RCMPCNT register (Table 3.50), a comparator interrupt is generated and indicated in the ADCCMPIF flag in the ADCSTAT register (Table 3.39).

Using the ADC comparator together with the counter allows filtering out the short-term deviations in the measured analog values (currents, voltages, positions, etc.).

Note: When the comparator is enabled (bit CMPEN = '1'), the ADCRDYIF flag is automatically masked. In this way, the global ADC interrupt (see Figure 3.8) is not always generated at the end of every conversion cycle, only when a valid comparator event is detected.

Signed or Unsigned Comparison

Whether the results are signed or unsigned in ADC comparator mode is determined automatically depending on the type of the ADC conversion and the value of the RESSIGN bit (see section 3.12.8). When the ADC performs ratiometric conversion, the ADC comparison runs in signed mode if bit RESSIGN = 1 and in unsigned mode if RESSIGN = 0. For absolute or temperature ADC conversion, the comparator performs unsigned results comparison.

ADCCMPH and ADCCMPL Registers

These registers are used for configuring the comparator mode and threshold value. The bits of both registers are described in Table 3.42. The reset values for registers ADCCMPH and ADCCMPL are 00_{HEX}.

Note: When the ADC conversion is running, WRITE access to registers ADCCMPH and ADCCMPL is disabled.

Table 3.42 ADCCMPH and ADCCMPL Bits Description

Bit	Description
Register ADCCMPH	
7	**CMPEN:** Comparator enable bit. 1 = The comparator is enabled. 0 = The comparator is disabled.
6	**CMPMD:** Comparison mode bit. 1 = The comparison returns TRUE if the ADC result is less than the programmed value. 0 = The comparison returns TRUE if the ADC result is greater than or equal to the programmed value.
5	**CNTDEC:** Determines the comparator's counter decrement behavior. 1 = When the comparison returns FALSE, the counter value is decremented by one. 0 = When the comparator returns FALSE, the counter value is cleared.
4:2	Unimplemented bits. Read as 0.
1:0	**CMPTH[9:8]**: Bits 9:8 of the ADC comparator threshold value.
Register ADCCMPL	
7:0	**Bits 7:0** of the ADC comparator threshold value.

Register RCMPCNT

This register contains the number of the consecutive comparator matches before interrupt ADCCMPIF is generated. If the value of RCMPCNT (Table 3.50) is 00_{HEX} (default value), the counter is bypassed and the interrupt flag ADCCMPIF (Table 3.39) is set on the first match. When the conversion is running, this register is accessed as read-only.

3.12.10. External ADC Analog Inputs Open/Short Diagnostics and Protection

ZAMC4100 provides open/short diagnostics for the analog input pins (AINx) and the sensor supply pin S_P as well. Pins S_P and AIN1,2 also have short-to-battery protection. The internal protection and diagnostic structure of the S_P and AINx pins is shown in Figure 3.25.

The diagnostic functionality for all pins (S_P, AINx) is controlled by register AINDIAG described in Table 3.43 and Table 3.44. Table 3.45 gives the meaning of the corresponding open/short flag from the AINDIAG register with respect to the combination of AINCSHEN and AINCSLEN bits that enable the diagnostic current sinks/sources connected to the analog input pins.

The protection against a short to Vbat on the S_P and AIN1,2 pins relies on Zener-like structure inside the ZAMC4100 that limits the voltage propagated to the internal circuitries to a safe level. The limitation of the current through the S_P pin in the event of such a failure is achieved with an internal resistor connected in series (see Figure 3.25-A. For the AIN1,2 pins this resistor is external (R2 and R3 on the application diagram as shown in Figure 6.1).

The protection against a short to GND on the S_P pin is achieved by limitation of the load current. Its value is specified in Table 1.4 as parameter I_{SP_OC}.

Note: Enabled diagnostic current sources during ADC conversion might result in inaccurate results. Disable all AINx and S_P pin diagnostics before starting an ADC conversion.

Figure 3.25 ADC Analog Inputs Protection and Open/Short Detection Circuitry

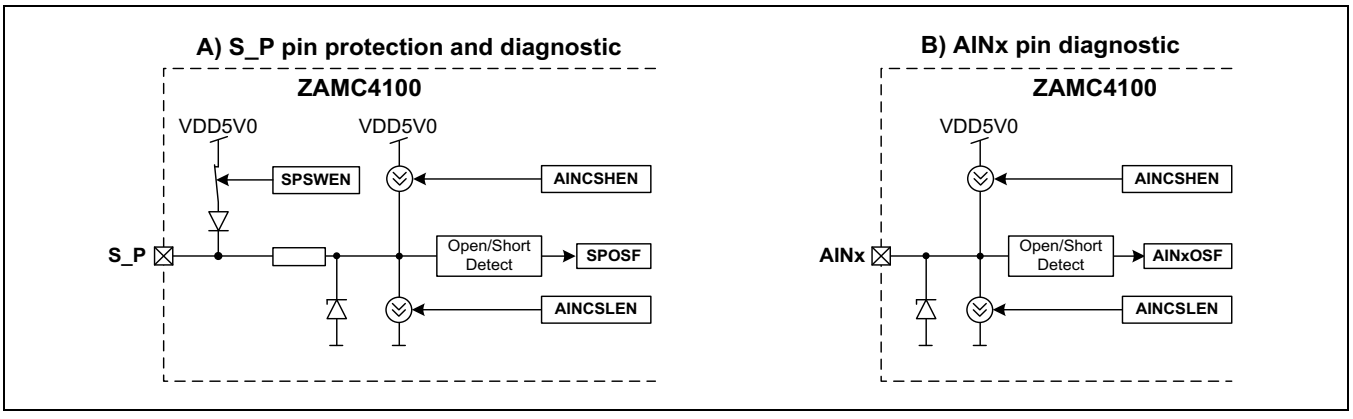


Table 3.43 AINDIAG Register Bits Mapping

Name	AINDIAG							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name		SPOSF	AIN4OSF	AIN3OSF	AIN2OSF	AIN1OSF	AINCSHEN	AINCSLEN
Reset	U	0	0	0	0	0	0	0
Access	U	R	R	R	R	R	R/W	R/W
Address	**0x16**							
R/W = Read/Write bit; R = Read-only bit; U = Unimplemented, read as '0'								

Table 3.44 AINDIAG Register Bits Description

Bit	Description
6	**SPOSF:** Open/short status flag of positive sensor supply pin (*S_P*). This flag is not registered; i.e. its value stays as long as the diagnostic event takes place. Refer to Table 3.45 for SPOSF bit description.
5:2	**AINxOSF:** Open/short status flag of the ADC external inputs AINx (x = 1 to 4). This flag is not registered; i.e. its value stays as long as the diagnostic event takes place. Refer to Table 3.45 for the interpretation of the AINxOSF bits.
1	**AINCSHEN**: [1), 2)] High-side current source enable bit for inputs AINx (x = 1 to 4) and S_P. 1 = The high-side current source is enabled. 0 = The high-side current source is disabled.
0	**AINCSLEN**: [1), 2)] Low-side current source enable bit for inputs AINx (x = 1 to 4) and S_P. 1 = The low-side current source is enabled. 0 = The low-side current source is disabled.

1) In order to keep current consumption as low as possible, the MCU software should disable the current sources before enabling the SLEEP or STANDBY Modes.

2) Make sure this bit is '0' when ADC conversion is enabled.

Table 3.45 External ADC Inputs Diagnostic Bits Description

AINCSHEN	AINCSLEN	SPOSF or AINxOSF	Description [1)]
0	0	x	Diagnostic disabled
1	1	x	Undefined result
0	1	0	Normal sensor or short to VSSA
1	0	0	
0	1	0	Open sensor connection
1	0	1	
0	1	1	Short to sensor supply
1	0	1	
0	1	1	Not possible in normal application
1	0	0	

1) The descriptions given are based on use of the external connections shown on the application diagram (see Figure 6.1). If another configuration of external connections is used, the meaning of the SPOSF and AINxOSF bits will be different.

© 2019 Renesas Electronics Corporation

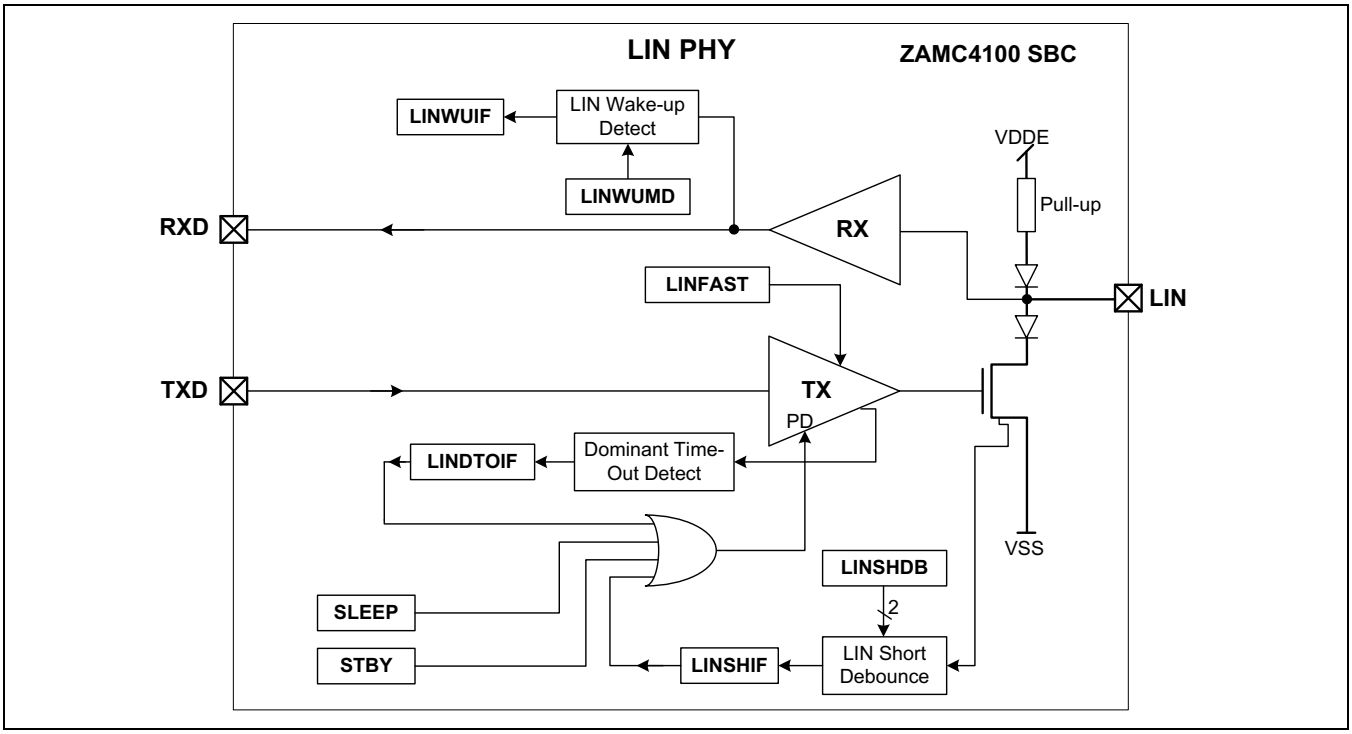
3.13. LIN Physical Layer (LIN PHY)

3.13.1. Overview

The block diagram of the LIN PHY layer is as shown in Figure 3.26. It is situated within the SBC chip and is designed to be conformant to the *LIN Specification Rev. 2.1*. The LIN PHY has following functionality:

- LIN bus wake-up detection
- LIN TXD dominant time-out detection
- LIN short detection
- LIN PHY power control

Figure 3.26 LIN PHY Block Diagram



This module has two registers:

- LIN control register LINCTRL (see Table 3.46)
- LIN status register LINSTAT (see Table 3.48)

3.13.2. LIN PHY Operation

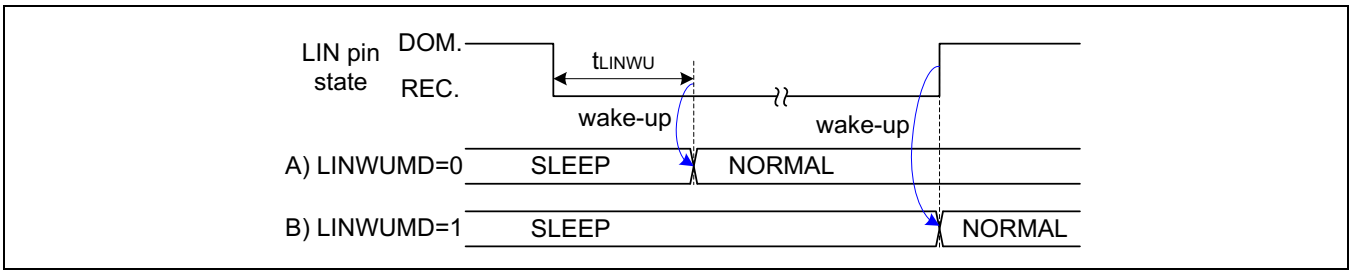
The status of the LIN PHY is checked by reading its interrupt flags mapped in the LINSTAT register. When an interrupt flag is set, the MCU can clear it by “dummy” writing ‘1’ in the corresponding LINSTAT register bit.

LIN Wake-Up Detection

When the system is in SLEEP or STANDBY Mode, the LIN transmitter is powered-down and only the receiver is functional. The last is used for detecting the LIN bus wake-up event.

The ZAMC4100 supports two wake-up scenarios depending on the value of the LINWUMD bit (see Table 3.46). If LINWUMD = 0 (see Figure 3.27-A), a wake-up is detected when there is a falling edge at the LIN pin followed by a dominant bus level longer than time t_{LINWU}. If bit LINWUMD = 1 (Figure 3.27-B), a wake-up is detected when there is falling edge at the LIN pin, followed by a dominant bus level longer than t_{LINWU} and a rising edge at the LIN pin. The value of the parameter t_{LINWU} is defined in Table 1.7.

Figure 3.27 LIN Wake-up Detection Depending on the Value of the LINWUMD Bit



When a valid wake-up is detected, the interrupt flag LINWUIF is set to high and the LIN transmitter is enabled. The status of this flag can be checked by reading the LINSTAT register (Table 3.48).

LIN Baud Rate

The LIN PHY supports a fast mode which provides a higher baud-rate communication than is specified in the standard (see Table 1.7). The LIN fast mode is enabled by setting high the LINFAST bit of register LINCTRL.

The LIN PHY has two self-diagnostics protections: LIN short and LIN dominant time-out.

© 2019 Renesas Electronics Corporation

LIN Short Protection

If there is a valid LIN short event, after the debounce time configured in the LINCTRL register (Table 3.46), the LIN short interrupt is generated, and the LIN transmitter is disabled. This interrupt is indicated in the LINSHIF flag bit in the LINSTAT register.

The ZAMC4100 detects a LIN short event if it is transmitting frames to a master device and the following conditions are both met concurrently:

1) The current through the output stage of the LIN transmitter exceeds the value specified by the parameter I_{OC_LIN} (see Table 1.7).
2) The MCU transmits the dominant state, but the LIN bus remains in the recessive state.

LIN Dominant Time-Out Protection

If the dominant state at the LIN transmitter's output continues more than 11ms, the logic detects this as a dominant time-out event and consequently disables the LIN transmitter. The dominant time-out detection generates an interrupt that is captured in the LINDTOIF flag of register LINSTAT. When LINDTOIF = HIGH, the LIN transmitter is automatically powered-down. In this way, the dominant state caused by the ZAMC4100 is isolated; i.e., the LIN bus is released and the other devices can continue to communicate.

Important Notes:

1. ZAMC4100 detects a LIN dominant time-out ONLY if it is caused by the device itself; i.e., due to an internal error in the ZAMC4100 LIN transmitter path. If the LIN bus is forced into dominant state externally (for more than 11ms), this is not a valid dominant time-out for the ZAMC4100 because it is not caused by its transmitter stage. This prevents detection of a dominant time-out that is caused by another device on the LIN bus, i.e. not by the ZAMC4100.
2. For robustness, the ZAMC4100 dominant time-out detection is implemented using both oscillators (OSCL and OSCH), which ensures that this type of failure event will be detected even if one of the oscillators has stopped.
3. The LIN dominant time-out detection feature is permanently enabled and cannot be disabled by software.

Table 3.46 LINCTRL Registers Bit Mapping

Name	LINCTRL							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name					LINWUMD	LINSHDB[1]	LINSHDB[0]	LINFAST
Reset	U	U	U	U	0	0	0	0
Access	U	U	U	U	R/W	R/W	R/W	R/W
Address	**0x17**							
R/W = Read/Write bit **U** = Unimplemented, read as '0'								

Table 3.47 LINCTRL Registers Bits Description

Bit	Description
3	**LINWUMD:** LIN wake-up detection mode (see Figure 3.27). 1 = LIN wake-up is detected if there is a falling edge at the LIN pin followed by a dominant state longer than t_{LINWU} and a rising edge at the LIN pin. 0 = LIN wake-up is detected if there is a falling edge at the LIN pin followed a by dominant state > t_{LINWU}.
2:1	**LINSHDB:** LIN short de-bounce time configuration. 00 = LIN short is detected if it is longer than 96 cycles of OSCH/5 clock. 01 = LIN short is detected if it is longer than 128 cycles of OSCH/5 clock. 10 = LIN short is detected if it is longer than 192 cycles of OSCH/5 clock. 11 = LIN short is detected if it is longer than 255 cycles of OSCH/5 clock.
0	**LINFAST:** LIN fast mode enable bit. 1 = Fast mode is enabled. 0 = Fast mode is disabled.

Table 3.48 LINSTAT Register Bits Mapping

Name	LINSTAT							
Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit name						LINSHIF	LINDTOIF	LINWUIF
Reset	U	U	U	U	U	0	0	0
Access	U	U	U	U	U	R/W	R/W	R/W
Address	**0x18**							
R/W = Read/Write ‘1’ bit; **U** = Unimplemented, read as ‘0’								

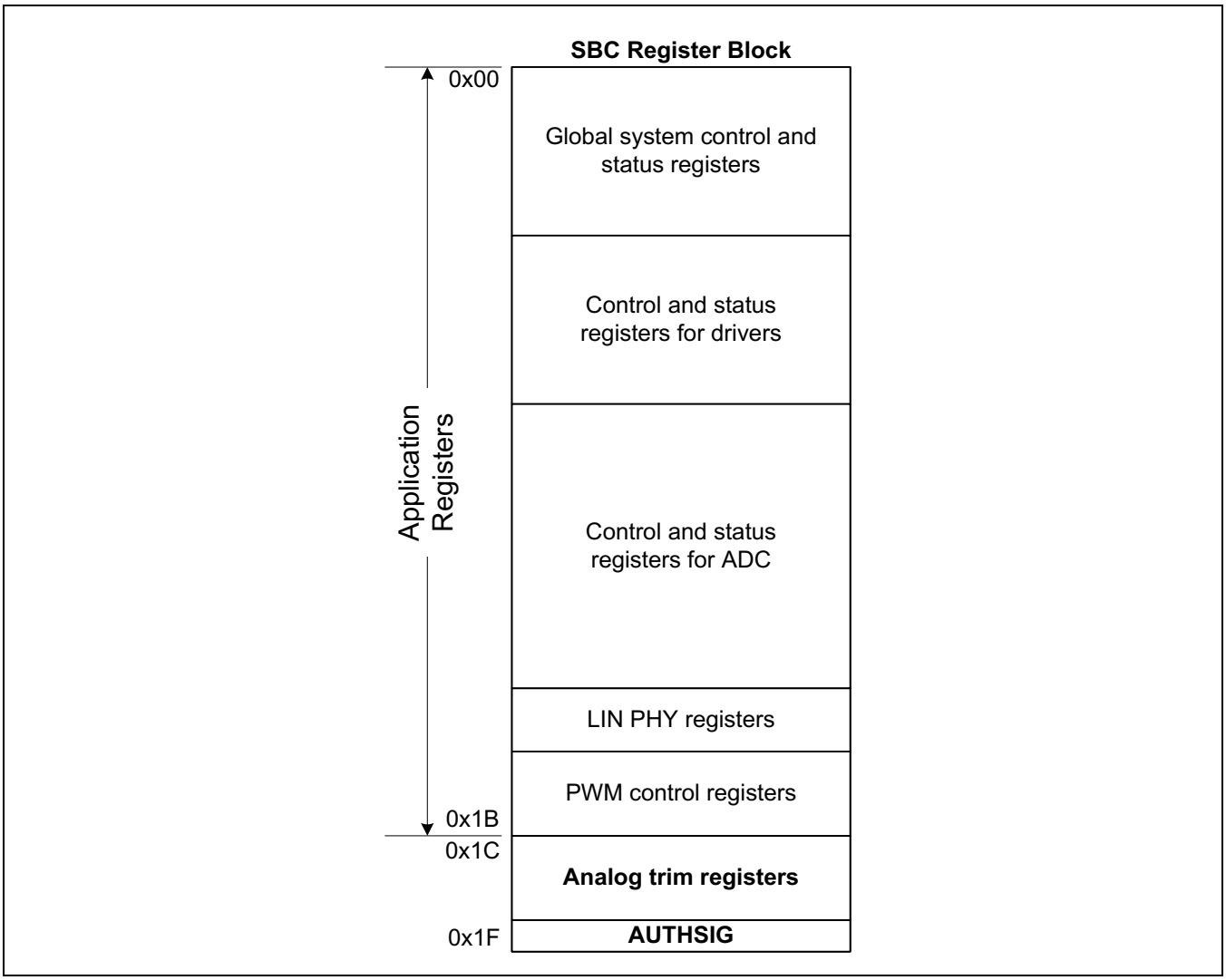
Table 3.49 LINSTAT Register Bits Mapping

Bit	Description
2	**LINSHIF:** LIN short interrupt flag. 1 = LIN short has been detected, and the transmitter is disabled. 0 = LIN short has not been detected.
1	**LINDTOIF:** LIN dominant time-out interrupt flag. 1 = Dominant time-out has been detected, and the transmitter is disabled. 0 = Dominant time-out has not been detected.
0	**LINWUIF:** LIN wake-up interrupt flag. 1 = Wake-up event has been detected on the LIN bus; i.e., the transition from SLEEP to NORMAL Mode is caused by the LIN bus. 0 = Wake-up event has not been detected on the LIN bus; i.e. any transition from SLEEP to NORMAL Mode has been caused by the WDT (see section 3.5.4).
Note: All LIN interrupt flags are cleared by writing ‘1’ in the corresponding LINSTAT bit.	

3.14. SBC Registers

All SBC registers are organized as one register file that is read/write accessible via the SPI. The SBC register file organization is shown in Figure 3.28.

Figure 3.28 SBC Register File Organization



The SBC register address space is 5-bits and is divided into two partitions: application registers and analog trim registers. The application registers are READ/WRITE accessible and contain system and periphery control registers.

The analog trim registers are WRITE-protected and can only be authorized by writing the correct code in AUTHSIG (see section 3.14.2).

© 2019 Renesas Electronics Corporation

Note: When the TRIM signature is set, the WRITE access to the application registers is forbidden. After the trim registers are written, the MCU should set the APPLICATION signature (see section 3.14.2) in order to have regular access to the application registers.

3.14.1. SBC Registers Address Map

Table 3.50 SBC Registers Address Map

Address [HEX]	Register Name	Reset Value, [HEX]	Description
Global control and status registers			
00	**RSTSTAT**	01	Reset status
01	**IRQSTAT**	00	Interrupt status
02	**IRQCTR**	00	Interrupt control
03	**SMDCTRL**	0C	System mode control
04	**WDTCONF**	11	Watchdog timer configuration
05	**SPIDPTR**	01	SPI data pointer
Drivers' control and status registers			
06	**HBDCTRL**	00	Half-bridge driver control
07	**HBDDIAG**	00	Half-bridge driver diagnostics enable
08	**HSDCTRL**	00	High-side driver control
09	**ECMCTRL**	00	Electrochromatic driver control
0A	**HBDSTAT**	00	Half-bridge driver status
0B	**HSDSTAT**	00	High-side driver diagnostics and status
0C	**ECMDIAG**	00	ECM driver diagnostics
ADC control and status registers			
0D	**ADCSTAT**	00	ADC status
0E	**ADCRESH**	00	ADC result high byte
0F	**ADCRESL**	00	ADC result low byte
10	**AINCONF**	00	Analog input pins configuration
11	**ADCCONF**	00	ADC configuration
12	**ADCCTRL**	00	ADC control
13	**ADCCMPH**	00	ADC result comparator threshold high byte
14	**ADCCMPL**	00	ADC result comparator threshold low byte
15	**RCMPCNT**	00	ADC comparator counter
16	**AINDIAG**	00	Analog inputs diagnostic
LIN PHY control and status registers			
17	**LINCTRL**	00	LIN control
18	**LINSTAT**	00	LIN PHY status

© 2019 Renesas Electronics Corporation

Address [HEX]	Register Name	Reset Value, [HEX]	Description
PWM control registers			
19	**PWMDCFG**	00	PWM duty cycle configuration
1A	**PWMFCFG**	00	PWM frequency configuration
1B	**PWMDREN**	00	PWM driver enable
Analog trim registers			
1C	**OSCHTRM**	00	OSCH frequency trimming
1D	**OSCLTRM**	00	OSCL frequency trimming
1E	**CREFTRM**	10	Reference current trimming
1F	**AUTHSIG**	00	Authentication signature write

3.14.2. Registers Write Access Signatures

There are two signatures pertaining to WRITE access for registers:

- APPLICATION signature: enables the WRITE access in the application registers.
- TRIM signature: enables the WRITE access in analog trim registers.

Both signatures are mutually exclusive. This means that if the application registers are WRITE accessible, the analog trim registers are READ-only and vice versa. The purpose of this logic is to avoid unintentional overwriting of the trim registers by the MCU and losing the correct calibration values.

The SBC indicates which signature is set with the ACCSIG bit, which is allocated in the slave status byte (see section 3.7.5):

- Bit **ACCSIG = 0:** APPLICATION access signature is set.
- Bit **ACCSIG = 1:** TRIM access signature is set.

AUTHSIG register

The AUTHSIG register is allocated at address $1F_{HEX}$ in the SBC register file. It is not physically implemented. Writing the correct values in this register will set the corresponding access signature. Reading the register will return a zero value.

TRIM signature setting

In order to set the TRIM signature, the MCU must consecutively write the AUTHSIG register (address $1F_{HEX}$) with the values AD_{HEX} and $B6_{HEX}$. For successful signature setting, both values should be written as two consecutive SPI WRITE accesses. If between the two writes, the MCU executes READ or WRITE access for another address, then no TRIM signature will be set.

APPLICATION signature setting

Once the TRIM signature is set, the MCU can return to having application registers access by writing the value 00_{HEX} in the AUTHSIG register.

© 2019 Renesas Electronics Corporation

3.15. SBC Trimming

All SBC analog circuitry trimming values are stored in the INFO page 0 of the MCU FLASH memory (Figure 4.1) during the ZAMC4100 production test. The MCU firmware has READ-only access to this partition; i.e., no trim data corruption is possible due to firmware malfunction.

This procedure must be implemented as part of the MCU firmware and executed during system initialization. When executed, the trimming procedure reads the OSCH, OSCL and Reference Current trim values from the FLASH INFO page and transfers them to the corresponding SBC trim registers.

ZAMC4100 trimming procedure

Trimming data is located in the INFO page 0 at offset 00000040_{HEX}. Trimming data is programmed in one word and has the following structure:

Figure 3.29 Structure of SBC Trimming Data

Byte 3	Byte 2	Byte 1	Byte 0
Not Used	CREF	OSCH	OSCL

The trimming procedure is given below. It must be executed as soon as the device is powered:

1. Disable the watchdog timer.
2. Write AD_{HEX} to the AUTHSIG register of the SBC.
3. Write $B6_{HEX}$ to the AUTHSIG register of the SBC.
4. Write OSCH to the OSCHTRM register of the SBC.
5. Write OSCL to the OSCLTRM register of the SBC.
6. Write CREF to the CREFTRM register of the SBC.
7. Write 00_{HEX} to AUTHSIG.
8. Setup and enable the watchdog timer. This step is optional.

This procedure prepares the MCU clock and current sources to be at their optimal value. It must be executed immediately after each MCU reset.

© 2019 Renesas Electronics Corporation

4 ZAMC4100 MCU Functional Description

4.1. Overview

The ZAMC4100 features a 32-bit ARM® Cortex™-M0 MCU with following peripherals:

- 32KB FLASH program memory
- 2KB SRAM data memory
- SPI master
- 32-bit timer
- 8 configurable GPIOs
- LIN UART

Note: Since the ZAMC4100 MCU is based on common Cortex M0 core architecture the description below excludes this topic. Refer to the ARM® CORTEX™-M0 Document Kit for MCU core details.

© 2019 Renesas Electronics Corporation

4.2. Memory Organization

The MCU contains two memory blocks: FLASH (program) memory and SRAM (data) memory. Together with the peripheral and system registers, their addresses are mapped as shown in Table 4.1. By default, the flash is mirrored to address 0x0. Software can change this and mirror the SRAM to address 0x0 instead by writing to the register field MEMSWAP in register SYS_MEMPORTCFG (see Table 4.6).

Table 4.1 MCU Memory Map

Address	Description
0xFFFF_FFFF	Reserved
0xE010_0000	
0xE00F_FFFF	Private Peripheral Bus; see the ARM® CORTEX™-M0 documentation for details
0xE000_0000	
0xDFFF_FFFF	Reserved
0x4000_1C00	
0x4000_1BFF	(SPI, SW-LIN)
0x4000_1800	
0x4000_17FF	GPIO
0x4000_1400	
0x4000_13FF	32-Bit Timer
0x4000_1000	
0x4000_0FFF	FLASH Info Page
0x4000_0C00	
0x4000_0BFF	FLASH Controller
0x4000_0800	
0x4000_07FF	Reserved
0x4000_0400	
0x4000_03FF	System Management Unit
0x4000_0000	
0x3FFF_FFFF	Reserved
0x2000_0800	
0x2000_07FF	SRAM Bank 2KB
0x2000_0000	
0x1FFF_FFFF	Reserved
0x1000_8000	
0x1000_7FFF	FLASH Bank 32KB
0x1000_0000	
0x0FFF_FFFF	Reserved
0x0000_0000	

© 2019 Renesas Electronics Corporation

4.2.1. Accessing Invalid Memory Location

A default slave replies with an error response if unused parts of the memory space are accessed. For more details, refer to the “AMBA 3 AHB-Lite Protocol Specification” section of the ARM® CORTEX™-M0 Document Kit.

4.2.2. FLASH Memory

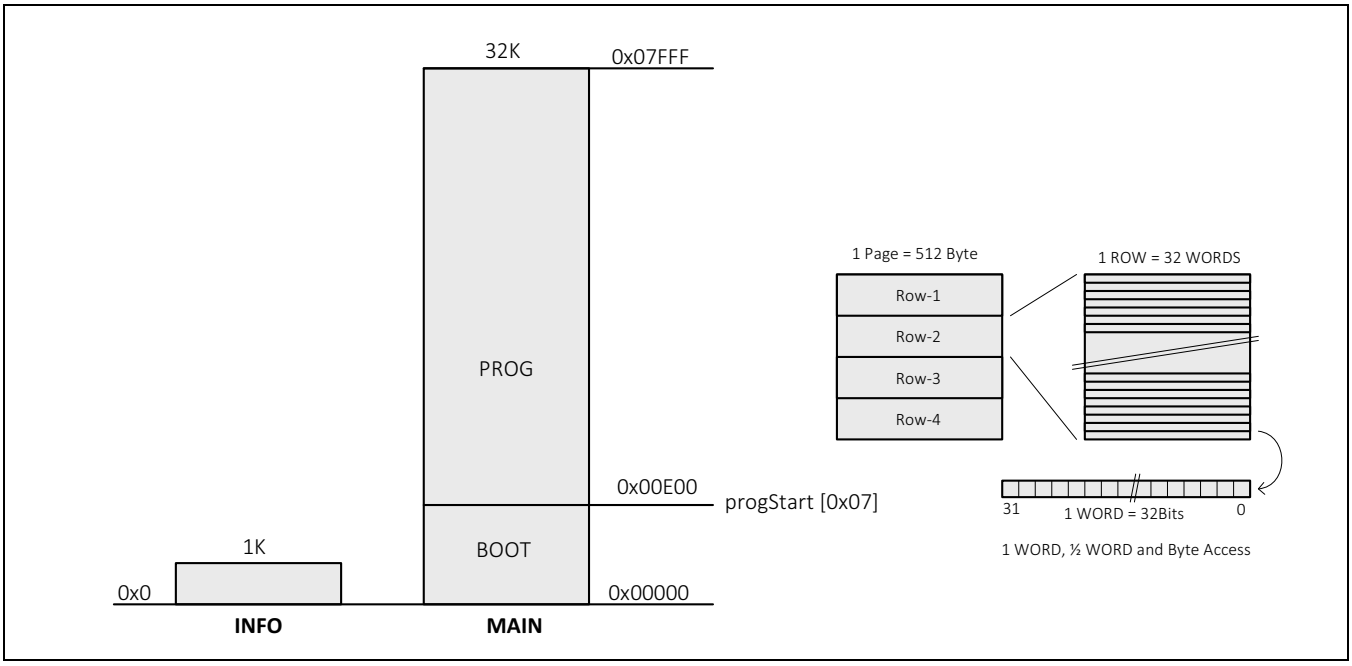
FLASH memory implements three regions (see Figure 4.1):

- INFO Contains READ-only parameter data set by manufacturing as well as memory protection management.
- BOOT Optionally used by a bootloader. The size of this area is programmable.
- PROG Program code memory, up to 32KB.

Each FLASH memory region is built with several *pages* of 512 bytes each. Each page is built with 4 *rows* and each row contains of 32 *words* (1 word = 32 bits). One flash page is the smallest block that can be erased.

As the FLASH memory has some dedicated timings for the control signals when erasing (all or part of) the FLASH and when writing data to the FLASH, a flash controller is used to support all mandatory operations (READ, WRITE, ERASE) to be performed on the different locations and to guarantee the correct timings for write and erase operations. Additionally, it is checked if the different operations are allowed to be performed depending on the memory protection scheme.

Figure 4.1 Structure of FLASH Memory and Page Details



4.2.2.1. FLASH Single Bits Error Correction (ECC)

Each word is protected by ECC logic with a hamming distance of 4, which enables the system to correct a single bit error and to detect two-bit errors within a word. The correct ECC code bits are automatically appended on each WRITE access to the FLASH. When a single error within a word is detected during a READ access, it is automatically corrected.

The occurrence of bit errors is signaled via dedicated status bits in registers FC_STAT_DATA (Table 4.38) and FC_STAT_PROG (Table 4.37) inside the FLASH controller. The status bits distinguish between an erased FLASH word (PROGALL1 flag), the detection and correction of a single-bit error (PROG1ERR flag), and the detection of more than one bit error (PROG2ERR flag). The condition of more than one bit error is not correctable. The two status register sets are distinguished by the type of FLASH access:

FC_STAT_PROG status bits are used when errors occur during an instruction fetch

- An instruction fetch to an erased memory location (PROGALL1 flag set) or the detection of more than one bit error within a word will also assert a non-maskable interrupt (NMI) as the program is corrupted.
- The detection and correction of a single-bit error within a word is signaled via a normal interrupt (ARM® interrupt line 0).

FC_STAT_DATA status bits are used when errors occur during a load operation

- Loads from an erased memory location as well as the detection (and correction) of errors within a word are signaled via a normal interrupt (ARM® interrupt line 0).

4.2.2.2. FLASH Main Area

The MAIN area of the FLASH is physically located within the address range 0x1000_0000 to 0x1000_7FFF. It can also be mirrored to address 0x0000_0000 by clearing the MEMSWAP bit in the SYS_MEMPORTCFG register (see Table 4.6), which is a default setting.

The FLASH MAIN area is split into two sections: BOOT and PROG. Each section is built with multiple FLASH blocks of 512 bytes. The BOOT and PROG sections must contain at least one FLASH block. This partition is only needed for writes when memory protection is active and also for some erase commands.

The MAIN area can be READ with byte, half-word and word size. It can always be read by the ARM® processor but reads via the JTAG interface are blocked when memory protection is active. The different sections have no influence on READ accesses.

4.2.2.3. FLASH Controller

The flash controller handles all accesses to the different flash locations (INFO pages; MAIN area). It provides different types of commands for modifying the flash contents, guarantees all required timings for the different accesses to the FLASH, and checks the ECC code.

The FLASH controller contains a set of registers that are needed for command execution (see Table 4.33) and observing their status. These registers are also used to enable the different interrupt sources to drive the interrupt line. Additionally, there are two non-maskable interrupt sources that are connected to the NMI of the ARM® core. Information for FLASH controller registers is given in section 4.10.6, and FLASH command timing is defined in Table 1.18.

© 2019 Renesas Electronics Corporation

4.2.3. INFO Area

The INFO area is mapped into the system address space between 0x4000_0C00 and 0x4000_0FFF. No memory location in the INFO pages can be directly written. The INFO area contains two pages of 512 bytes each:

- INFO page 0 (lower half of addresses) contains traceability information and trimming values for SBC analog circuitry (see section 3.15). This page is only readable and cannot be accessed indirectly by the flash controller (no WRITE, no ERASE). The INFO page 0 base address is 0x4000_0C00.
- INFO page 1 (upper half of addresses) is a mirror copy of INFO page 0 for redundancy of INFO page data content.

Table 4.2 INFO Page 0 Structure

Offset	Size	Description
0x00000000 to 0x0000003C		Traceability data
0x00000040	1 word	SBC Trim Data
0x00000044 to 0x000000C0		Reserved
0x000000C4	1 word	Exact value for I_{REF} (see Table 1.12) Note: The number stored represents I_{REF} in nA.
0x000000C8	1 word	Exact value for V_{REF} (see Table 1.12) Note: The number stored represents V_{REF} in mV.
0x000000CC	1 word	Exact value for V_{t27} (see Table 1.11) Note: The number stored represents V_{t27} in mV.
0x000000D0	1 word	Exact current sense ratio for HS1 (see Table 1.8)
0x000000D4	1 word	Exact current sense ratio for HS2 (see Table 1.8)
0x000000D8	1 word	Exact current sense ratio for HS3 (see Table 1.8)
0x000000DC	1 word	Exact current sense ratio for HS4 (see Table 1.8)
0x000000E0	1 word	Exact current sense ratio for HB1 high-side (see Table 1.8)
0x000000E4	1 word	Exact current sense ratio for HB1 low-side (see Table 1.8)
0x000000E8	1 word	Exact current sense ratio for HB2 high-side (see Table 1.8)
0x000000EC	1 word	Exact current sense ratio for HB2 low-side (see Table 1.8)
0x000000F0	1 word	Exact current sense ratio for HB3 high-side (see Table 1.8)
0x000000F4	1 word	Exact current sense ratio for HB3 low-side (see Table 1.8)
0x000000F8	1 word	Exact current sense ratio for HB4 high-side (see Table 1.8)
0x000000FC	1 word	Exact current sense ratio for HB4 low-side (see Table 1.8)

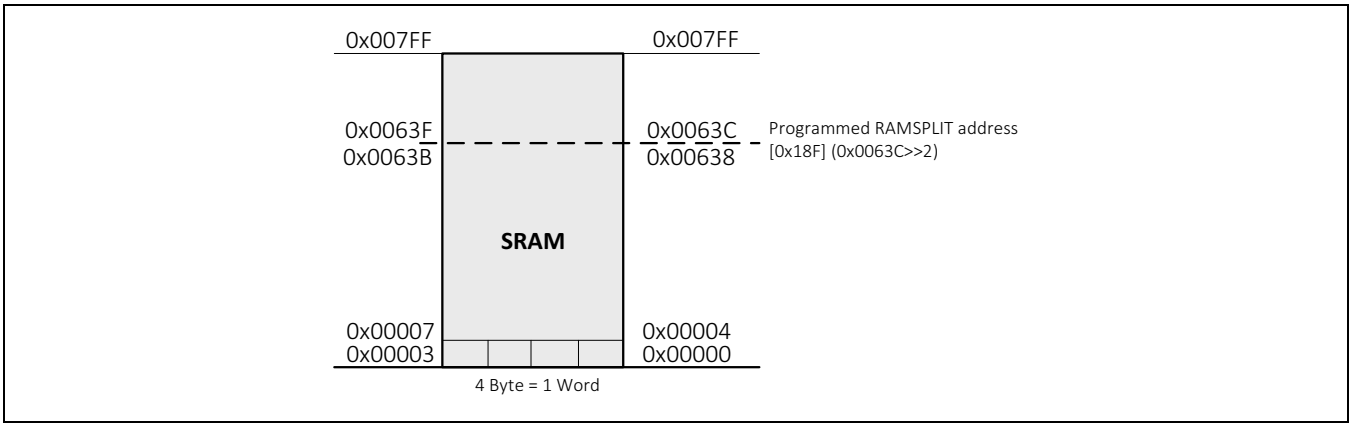
4.2.4. RAM Memory 2kB

The SRAM is physically located at addresses 0x2000_0000 to 0x2000_07FF. It can be accessed (READ and WRITE) with byte, half-word, and word size. The SRAM can also be mirrored to address 0x0000_0000 by setting the MEMSWAP bit in register SYS_MEMPORTCFG (see Table 4.6).

© 2019 Renesas Electronics Corporation

The entire RAM is always accessible by the ARM® processor, but there are restrictions for RAM access via the JTAG interface. From the perspective of the JTAG interface, the RAM is split into two sections with a boundary configurable via the RAMSPLIT bit field in the SYS_MEMINFO register (see Table 4.7).

Figure 4.2 Example of RAMSPLIT Address Configuration



The RAMSPLIT address is word-aligned and stored in the FLASH. The upper section starting with address RAMSPLIT can always be accessed via JTAG. The lower section can only be accessed via JTAG when no memory protection is active. The stack used by software must be placed into the lower section to avoid unauthorized access, which could result in potentially damaging changes.

Notes:

1. Always place the software stack into the lower section of the RAM. The lower section ends one word address before the address RAMSPLIT.
2. The RAMSPLIT address can always be determined from register SYS_MEMINFO (see Table 4.7). The last two address bits are not included in the read value as they are always 0.

4.3. MCU Clock and Reset Sources

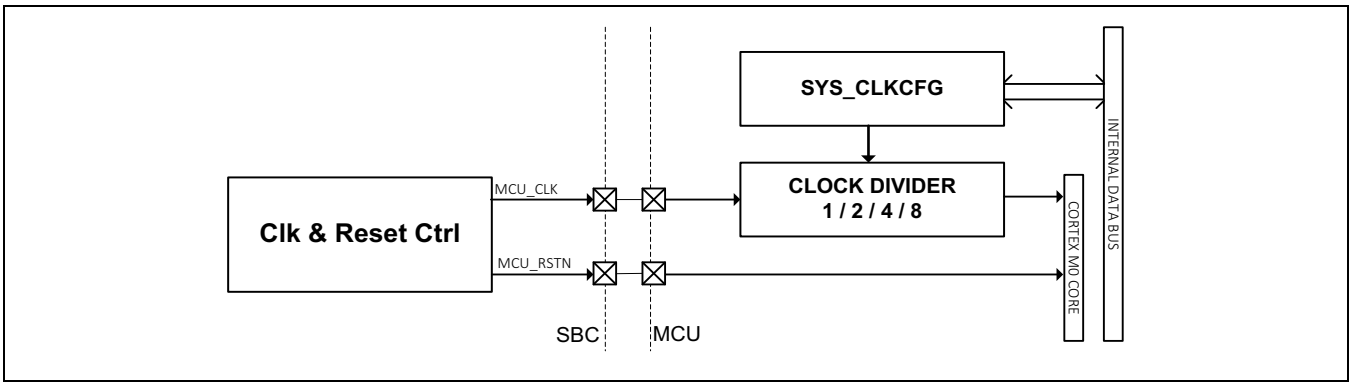
The RESET and clock operation are managed by the SBC as detailed in section 3.2. The main clock frequency is set to 20MHz for the MCU core.

As shown in Figure 4.3, the MCU_CLK from the SBC go to the clock divider with 1:1 as the default divider. The CLKDIV bit field in the SYS_CLKCFG register (Table 4.5) can be configured for other settings for the divider so that the system runs at 20MHz, 10MHz, 5MHz, or 2.5MHz. It is not advisable to change the clock division setting during any of these operations:

- FLASH programming
- LIN operations, unless the LIN module is put into the safe state, which is achieved by setting the STOPRX bit and clearing the ENTOCNT bit in the Z1_LINCFG register (Table 4.25).
- Active operations associated with the 10msec reference of the SYST_CALIB register of the ARM® core. This must be disabled before making adjustments for the clock division.

The SBC has full control in configuring the SLEEP and NORMAL operation modes. In STANDBY Mode, the ARM® Cortex™-M0 core is internally put into the SLEEP Mode by suspending the processor clock. Refer to the ARM® Cortex™-M0 Document Kit for more details. Refer to section 3.5 for more information about the SLEEP and NORMAL Modes.

Figure 4.3 RESET and CLOCK between SBC and MCU

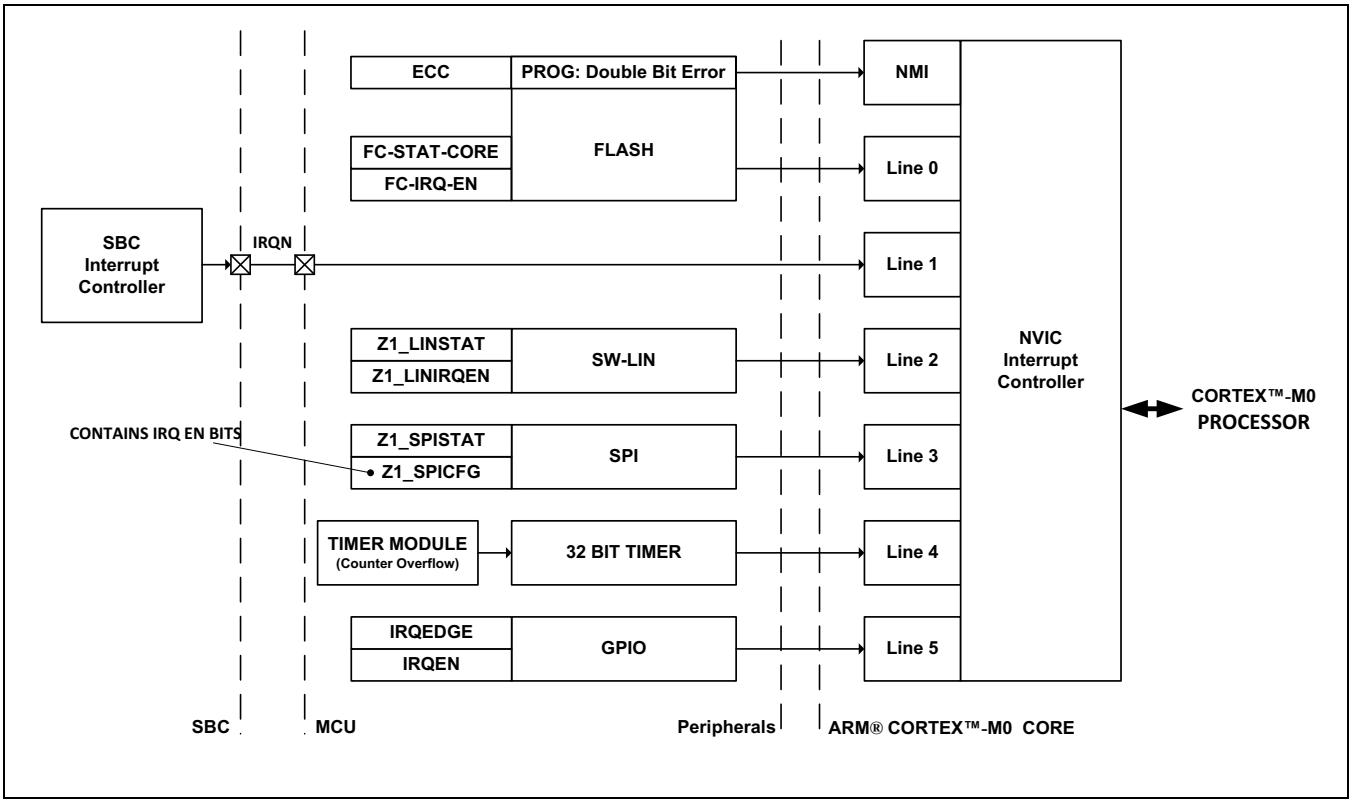


4.4. MCU INTERRUPTS

4.4.1. Interrupts Organization

Figure 4.4 shows the peripheral interrupts with the related control and status registers. Each peripheral has a unique IRQ line that is attached to dedicated interrupt lines 0 to 5, which go to the nested vector interrupt controller (NVIC) module within the ARM® CORTEX™-M0 processor. For more information about the NVIC, refer to the NVIC chapter of the ARM® CORTEX™-M0 Document Kit.

Figure 4.4 MCU Interrupt Logic Organization



Note: The Systick has an NVIC number of -1 which is within the Cortex™-M0 core. For more details, refer to the ARM® CORTEX™-M0 Document Kit.

4.4.2. Interrupts Configuration

As defined in Table 4.3, the MCU peripheral interrupts can have one of two types of interrupt input responses: level-sensitive or pulse (edge) triggered:

Table 4.3 MCU Peripheral Interrupts

Interrupt Line	MCU Peripheral Interrupt	Type
Interrupt line 0	Flash controller interrupt	Level-sensitive
Interrupt line 1	External interrupt (from SBC)	Level-sensitive
Interrupt line 2	SW-LIN interrupt	Level-sensitive
Interrupt line 3	SPI interrupt	Level-sensitive
Interrupt line 4	32-bit timer interrupt	Pulse
Interrupt line 5	GPIO interrupt	Level-sensitive

The effect of the pulse and level type is explained in the ARM™ Cortex™-M0 Documentation Kit in the NVIC section.

4.5. GPIO Module

4.5.1. Overview

There are 8 GPIO pins (GPIO00 to GPIO07) implemented in the MCU. Each GPIO pin can be individually configured to operate as an input or output. When configured as an output, the value driven out of the GPIO pin can be directly written or controlled via a set-clear register (see section 4.5.3).

GPIO pin configurations:

- Output mode: enabled by clearing a corresponding bit in registers GPIO_DIR (Table 4.9) and GPIO_OUT (Table 4.11); e.g., bit 0 = GPIO00 configuration, bit 1 = GPIO01, etc.
- Schmitt trigger input mode with internal pull-down resistors: enabled by setting a corresponding bit in registers GPIO_DIR and GPIO_IN (Table 4.10)
- Generation of interrupt events on a selectable edge
- Trigger source for the 32-bit timer

By default all GPIOs are set in *input* mode. Each register in a GPIO module can be accessed with a byte, half-word and word size.

4.5.2. GPIO Input Mode (default)

Assigned bits in register GPIO_DIR must be cleared for the input operation for a specific GPIO pin (bit numbers correspond to GPIO numbers). The value for each pin can be read by reading register GPIO_IN. The values from any GPIO pins that are configured as outputs or which have other functionality should be ignored.

© 2019 Renesas Electronics Corporation

4.5.3. GPIO Output Mode

Assigned bits in the GPIO_DIR register must be set for output operation for a specific GPIO pin (bit numbers correspond to GPIO numbers). The value can be written into the GPIO_OUT register, which defines the GPIO output level. If a single GPIO bit is to be modified, a more direct way to configure the bit is via the GPIO_SETCLR register (see Table 4.12), which can be used instead of performing the read-modify-write operation. Corresponding bits in the SETOUT[7:0] bit field are for setting the bit to 1 and corresponding bits in the CLROUT[23:16] bit field are for clearing the bit.

Note: Write the initial value to the GPIO_OUT register before defining GPIO_DIR for the output.

4.5.4. GPIO as Timer Trigger Operation

Each GPIO pin can be used as an external trigger source for the 32-bit timer (see section 4.6.1). To enable the trigger functionality, the GPIO pin must be configured as an input and the trigger functionality must be enabled via the GPIO_TRIGEN register (Table 4.14).

4.5.5. GPIO Interrupt Functionality

Each GPIO pin can be used as an external, edge-sensitive interrupt source. To enable the interrupt functionality, the GPIO pin must be configured as an input and the interrupt functionality must be enabled via register GPIO_IRQEN (Table 4.14). Additionally, the GPIO_IRQEDGE register (Table 4.15) can be used to select whether a rising or falling edge on the GPIO pin activates the interrupt.

All GPIO pins enabled as an external interrupt source drive a single interrupt line connected to the ARM® interrupt 5. The user can determine which GPIO pin caused the interrupt by reading register GPIO_IRQSTAT (Table 4.13). All interrupt status bits are cleared when reading register GPIO_IRQSTAT.

Note: As the synchronization flip-flops are only continuously clocked when the trigger or interrupt functionality is enabled for the corresponding GPIO pin, it is possible that an unwanted interrupt occurs when enabling the interrupt functionality. To avoid this, the following sequence must be guaranteed by software:

- Enable the GPIO trigger functionality and select the desired interrupt edge to be used.
- Enable the GPIO interrupt functionality at least three cycles after enabling as a trigger.
- Disable the GPIO trigger functionality.

4.5.6. GPIO Registers

Refer to section 4.10.2 for GPIO register details.

4.6. 32-Bit TIMER Module

The timer provides event-counting on the rising clock edge with a 32-bit resolution. When enabled (EN bit = 1 in register T32_CTRL; see Table 4.17), the timer counts clock events in timer mode or counts events from a selectable external trigger signal in counter mode as described in the next two sections. The external trigger can be configured to operate on rising/falling edges or on a low/high level via the MODEPN and MODELE bits. Additionally, the MODESR bit can be used to select whether the timer/counter stops when it overflows or continues its operation.

© 2019 Renesas Electronics Corporation

4.6.1. Timer Mode (MODETC = 0)

In timer mode, the counter register is incremented in each clock cycle. When the counter reaches 0xFFFF_FFFF, the reload value is copied into the counter register and it sets both the OVERFLOW flag bit and the interrupt line. The OVERFLOW bit (see Table 4.17) and interrupt line are set high for one clock cycle.

When reload mode is enabled (bit MODESR = 0 in the T32_CTRL register), the counter continues counting. Otherwise the counter stops. The two other control bits, MODELE and MODEPN in the T32_CTRL register, have no meaning in this mode.

4.6.2. Counter Mode (MODETC = 1)

The counter register is incremented in each clock cycle when the trigger is active. When the counter has a value of 0xFFFF_FFFF and the trigger is active, the reload value is copied into the counter register and the OVERFLOW bit and the interrupt line are set high for one clock cycle. When the reload mode is enabled (MODESR = 0 in the T32_CTRL register), the counter continues counting. Otherwise the counter stops.

The two control bits MODELE and MODEPN are used to configure the trigger as shown in Table 4.4.

Table 4.4 Configuration of Timer Trigger Behavior

MODELE	MODEPN	Sensitivity	Behavior
0	0	Falling Edge	Trigger is active when trigger input has a falling edge between clock cycles.
0	1	Rising Edge	Trigger is active when trigger input has a rising edge between clock cycles.
1	0	LOW	Trigger is active when trigger input is low.
1	1	HIGH	Trigger is active when trigger input is high.

4.6.3. TIMER Module Interrupt

The timer has an interrupt line that is active-high and set high for a single clock cycle whenever the counter overflows. The interrupt line is connected to the ARM® interrupt 4.

4.6.4. TIMER Module Registers

The TIMER module uses the following registers:

- T32_CTRL TIMER control and configuration register (Table 4.17)
- T32_TRIGSEL Trigger select register (Table 4.18)
- T32_CNT Register containing current TIMER value (Table 4.19)
- T32_REL Register containing (Table 4.20)

For detailed register address mapping and bit descriptions, refer to section 4.10.3.

4.7. SysTick

The Cortex™-M0 core has an integrated SysTick module. For configuration information, refer to the ARM® Cortex™-M0 Documentation Kit.

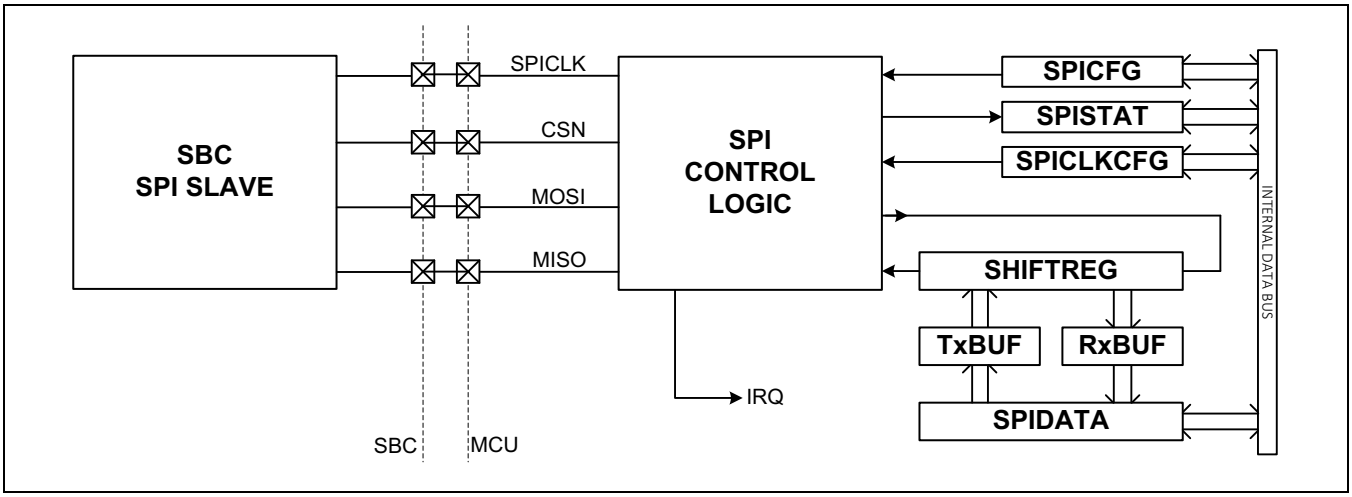
© 2019 Renesas Electronics Corporation

4.8. Master SPI Module

4.8.1. Overview

The integrated SPI is a pure four-wire master interface that is used for communication with the SBC. As shown in Figure 4.5, the three lines SPICLK, MOSI, and MISO are fully controlled by hardware while the chip select CSN line is controlled by software in accordance with the setup time requirement as per SPI bus timing (for specifications, see section 1.3.3).

Figure 4.5 Master SPI Block Diagram



4.8.2. Setup Requirements for the Slave SPI of SBC

For compatibility with the SBC SPI SLAVE, the following settings must be made in the MCU SPI MASTER:

- Clock phase = 1 and clock polarity = 1. This is the default configuration (after MCU reset) implemented via the CPHA and CPOL bits of the Z1_SPICLKCFG register respectively (Table 4.23).
- SPI clock frequency calculated by the formula: $F_{SPICLK} = \frac{F_{OSCH}}{2 * (CDIV + 1)}$

  Where CDIV is the clock divider value bit field in the Z1_SPICLKCFG register.

Note: For the SPI to operate within its valid speed range, the value programmed in CDIV must not be less than 2.

4.8.3. SPI Data Transfer Process

To start a transfer, the SPI SLAVE module must be enabled by setting the SPIEN bit in the Z1_SPICFG register (Table 4.21) and then setting the clock period via the CDIV bit field in the Z1_SPICLKCFG register (Table 4.23). The software then sets the SSN bit low, which enables the SPI SLAVE within the SBC. In addition, any necessary interrupt source must be enabled. Note that the TX buffer (TxBUF) illustrated in Figure 4.5 is empty until the data transfer has been started.

Upon expiration of the specified SSN set-up time period (see Table 1.16), the first byte to be transmitted is set up by writing a data byte into the SPIDATA register. This will clear the TXEMPTY flag bit in the Z1_SPISTAT register (Table 4.24). In the next system clock cycle, this byte is transferred into the shift register and this causes the TXEMPTY and BUSY flag bits to be set in the Z1_SPISTAT register. This allows the user to place a second byte into the TX buffer. The first byte is shifted out on the MOSI line and the SPI clock is generated.

Simultaneously, the MISO line is sampled and shifted in. Normally, the MISO line is sampled in the middle of a transmitted bit. While the MOSI line changes its value at the same time as the SPI clock, there is a delay regarding the MISO line as first the clock must be driven out of the chip into the connected SLAVE and then the data must be driven back from the connected SLAVE. To relax the timing, especially for fast SPI clocks, the RX data can be sampled at the end of a transmitted bit if configured via the SAMPLEPOS bit in the Z1_SPICFG register.

When the complete byte has been shifted in and the RX buffer is empty, the byte is stored into the RX buffer at the byte boundary and the RXFULL flag bit is set in the Z1_SPICFG register, signaling the end of the byte transfer.

If the RX buffer is already full and the byte inside the RX buffer is not read in the same cycle, the byte currently received is rejected (lost) and the RXOF flag bit is set in the Z1_SPISTAT register (Table 4.24).

Note: Because the SPI module operates in a full-duplex mode, a dummy byte must be placed into the TX buffer if a byte must be read from the SLAVE without any WRITE to the SLAVE.

4.8.4. Continuous SPI Data Streaming

When a new byte is written into TX buffer before the end of an active byte transfer, the transfer of the new byte starts immediately after the actual transfer. This means that the BUSY flag bit in the Z1_SPISTAT register stays active at the end of the first transmitted byte.

4.8.5. Abrupt SPI Discontinuity

Since it can be possible that the software disables the SPI while a transfer is in progress (not recommended), a byte could be present in the TX buffer indicated by a low value of the TXEMPTY flag bit in the Z1_SPISTAT register. This byte can be removed from the TX buffer by writing a 1 to the CLRTXBUF bit of the Z1_SPISTAT register as it would be transmitted when SPI is enabled again.

4.8.6. Interrupts and Status Flags in the Z1_SPISTAT Register

There are five status flags available in the Z1_SPISTAT register. Only four of them can be enabled to drive the interrupt lines as summarized below and in Table 4.24. The software is required to deal with interrupt events for proper SPI operation. The BUSY flag is used to reflect the operation state of the SPI module.

© 2019 Renesas Electronics Corporation

RXOVERFLOW: This bit is set by hardware when it is not able to store a received byte into the RX buffer (RX buffer is already full). It is cleared by a software READ access to the status register. To prevent losing any information, the set condition has higher priority than the clear condition. This bit is not set when the byte in the RX buffer is read in the same system clock cycle when the next received byte will be stored.

RXFULL: This bit is set by hardware when a received byte is stored into the RX buffer and cleared when the byte is read by the software. To prevent losing any information, the set condition has higher priority than the clear condition. This situation occurs when the byte in the RX buffer is read in the same system clock cycle when the next received byte will be stored.

TXEMPTY: This flag is active on default. It is cleared when software writes a byte to the TX buffer and is set by hardware when it moves this byte into the shift register. As it might be possible that both actions happen in the same system clock cycle, the clear condition has the higher priority.

WRCOLLISION: This flag is set when the software writes a byte into the TX buffer while the TX buffer is not empty and its contents have not been moved into the shift register in the same system clock cycle. The byte that the software intends to write is rejected to avoid loss of data. It is cleared by a software READ access to the status register.

4.8.7. Syncing the Data Transfer Rate via a Software Pause

When the software is not able to read the RX byte before the next byte is received (RXOVERFLOW bit = 1 in the Z1_SPISTAT register), the timing can be relaxed by not writing the second TX byte until the first RXFULL interrupt in the Z1_SPISTAT register. Instead the second TX byte can be written after this interrupt. This guarantees that no RX overflow can occur but introduces some delay between two consecutive bytes as the module waits for the next byte transfer until data is present.

4.8.8. Master SPI Registers

The following registers are used by the SPI MASTER module:

- Z1_SPICFG SPI configuration register
- Z1_SPIDATA READ/WRITE access to this register accesses the SPI RX data buffer and the TX data buffer respectively
- Z1_SPICLKCFG SPI clock configuration
- Z1_SPISTAT Collects SPI status flags

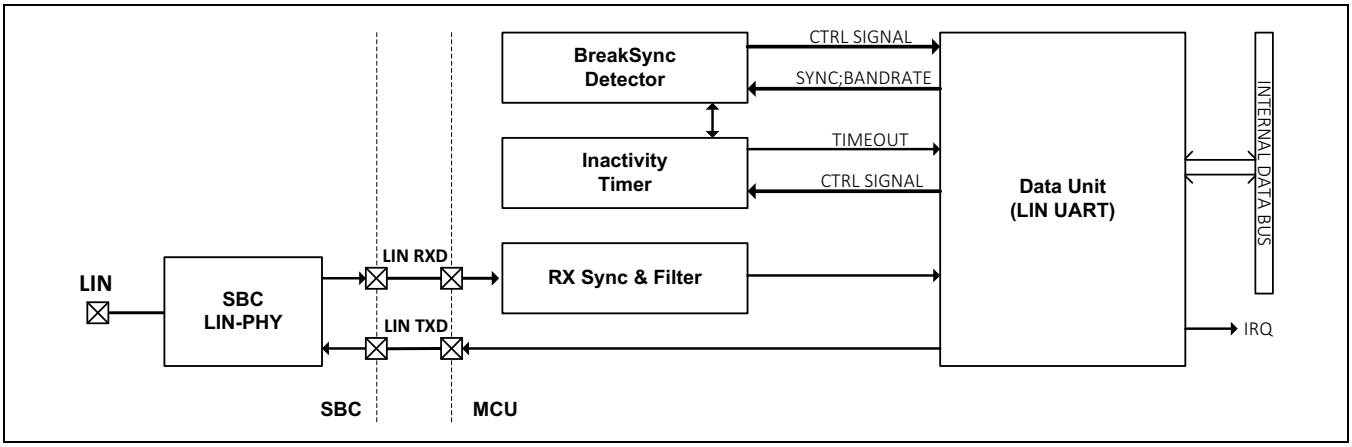
For detailed register address mapping and bits description, refer to section 4.10.4.

© 2019 Renesas Electronics Corporation

4.9. SW-LIN Module

The SW-LIN provides the logic for communication on the LIN bus via the LIN PHY within the SBC. It is compliant with *LIN Specifications Rev. 2.1* and operates as a LIN SLAVE only. Most of the protocol must be handled in software.

Figure 4.6 SW-LIN Block Diagram



The hardware consists of a BREAK/SYNC field detector for bus synchronization and baud rate detection, an inactivity timer, a data unit similar to a UART for communication, and a small block to synchronize and filter the incoming data line.

The BREAK/SYNC field detector is used to detect any occurrence of a BREAK field and a SYNC field as described in the LIN standard on the bus. It generates a sync strobe at the rising edge of RXD at the beginning of the STOP bit and updates the baud rate as needed. The LIN standard defines the baud rate as being between 1kBaud and 20kBaud; however the SW-LIN is able to operate at even higher baud rates. The maximum baud rate depends on the clock divider value and whether or not the LIN is operating in fast mode. In slow mode, the maximum baud rate is 30kBaud independent of the clock divider value. In fast mode, the maximum baud rate is 75kBaud for a clock divider value of 3, 150kBaud for a clock divider value of 2, and 200kBaud for others.

The inactivity timer observes the RXD line and generates an interrupt when the LIN bus is inactive for more than 4 seconds as required by the LIN standard.

4.9.1. The Inactivity Timer

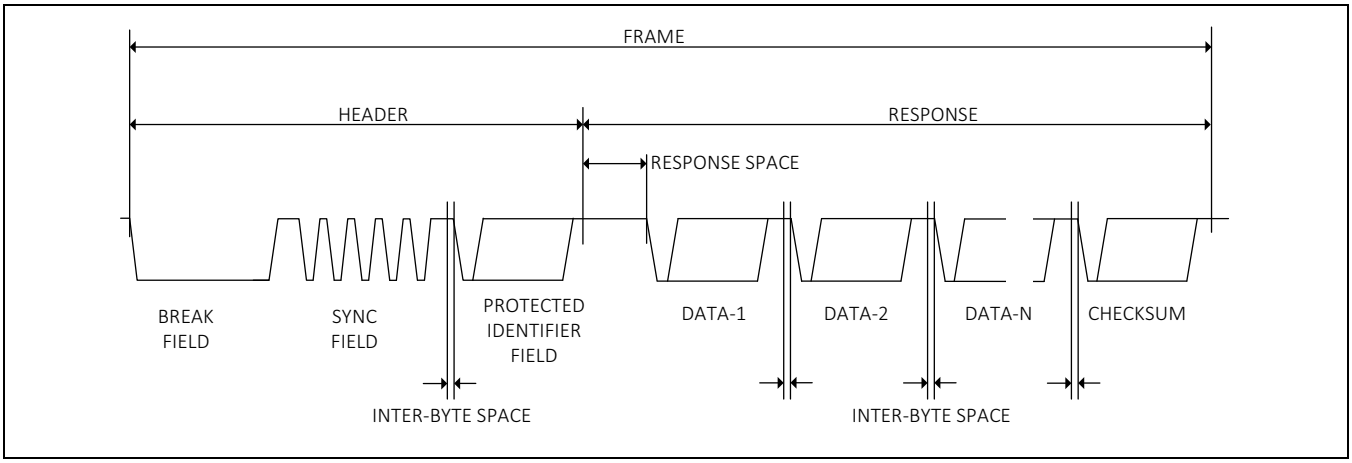
The SW-LIN contains an inactivity timer. This module is required as the LIN standard requires that a LIN SLAVE goes to sleep after more than 4s but less than 10s of inactivity on the bus. As bus inactivity means that there is no change on the bus regardless of whether the bus is high or low, a timer is implemented which is reset on each transition on the bus (falling or rising edge of the RXD line) and which is incremented when the timer has not expired. When this timer expires, an interrupt is generated so that the software can disable the SW-LIN.

The LIN standard describes two ways of going to sleep. In addition to the inactivity timeout, the LIN SLAVES must go to sleep when the corresponding SLEEP command has been received. If the LIN master in the full application guarantees that a SLEEP command is always sent if the bus is not required anymore and a timeout can never occur. The software can disable the inactivity timer to save power.

4.9.2. The BREAK/SYNC Field Detector

As described in the LIN standard, each transaction on the bus is initiated by the LIN master sending the header of a LIN frame. This header consists of the BREAK field, the SYNC field, and the PID field. The first two fields are used by the slave to determine the baud rate used for the rest of the transaction while the PID is used to determine the type of the transaction (receive response; transmit response; do nothing). To enable the slave to receive the PID, the baud rate must be detected before the START bit of the PID field. As a new BREAK and SYNC field can occur at any time, an active transaction is interrupted when those fields are detected.

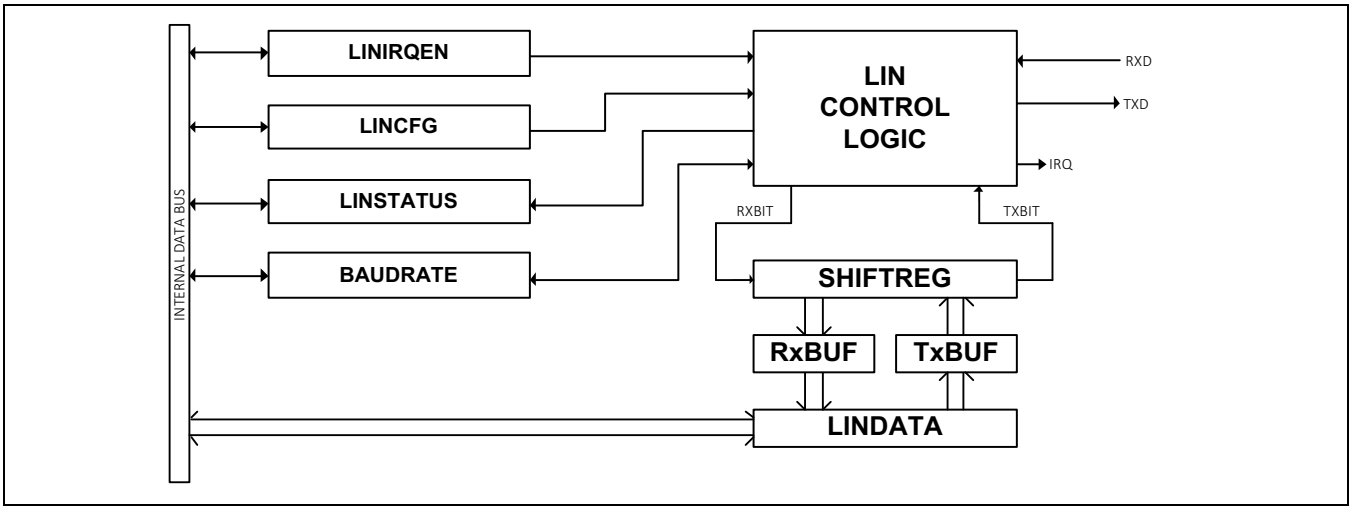
Figure 4.7 Structure of LIN Frame



4.9.3. SW-LIN Data Unit

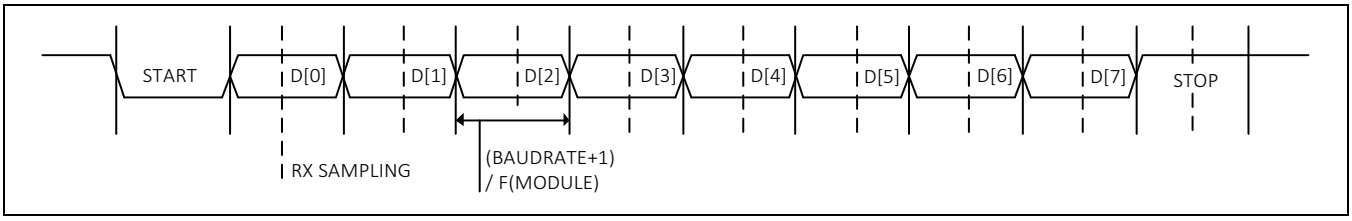
The data unit handles the reception and transmission of bytes (PID, DATA, and CRC as described in the LIN standard). It contains all registers accessible by the user and generates the interrupt (Figure 4.8).

Figure 4.8 Block Diagram of the LIN Data Unit



Although the fields to be received and transmitted have the same structure as a UART, the control logic is different as the TX and RX channel are not independent. Each transfer consists of 8 data bits enclosed by a leading START bit and a trailing STOP bit. On reception, the receiver synchronizes on the falling edge of the RXD line (START) and samples the incoming data stream in the middle of each data bit. Additionally it checks that the STOP bit is high. For transmission, the module itself generates the data stream, but it also checks that it can receive the bits sent on its RXD line.

Figure 4.9 Frame Format of Each LIN Field (PID, DATA, Checksum) and RX Sample Position



4.9.4. Description of the Receive Operation

The receive path is directly controlled by the LIN bus. Whenever a valid BREAK and SYNC field is detected by the BREAK/SYNC field detector, the baud rate registers are updated, the receiver is enabled (RXEN bit is set to 1 in the Z1_LINCFG register; see Table 4.25) but placed into an inactive state (RXACTIVE bit is set to 0 in Z1_LINCFG), and the RX control logic is set appropriately. Additionally, the SYNCDET interrupt flag in the LINSTAT register (Table 4.26) is set, which must be cleared by software. If the receiver was already active on reception of the sync strobe, the active transfer is discarded as it was receiving a new SYNC field.

After being enabled in inactive state, the bus is observed for a START condition (falling edge on the RXD line). When a START condition is detected, the receiver is placed into the active state (RXACTIVE bit is set to 1 in the Z1_LINCFG). The data is sampled into the shift register in the middle of each data bit. When the complete byte has been shifted in, the receiver is placed back into its inactive state in the middle of the STOP bit. When the STOP bit is high (as it should be), the received byte is placed into the RX buffer if this buffer is empty and then the buffer is marked as full (RXFULL bit is set to 1 in LINSTAT). Otherwise the actual received byte is rejected and the RX overflow flag is set (RXOVERFLOW bit is set to 1 in LINSTAT). If the STOP bit is low (wrong baud rate detection or another BREAK and SYNC field is send by the master), no data is placed into the RX buffer. Instead the receiver is disabled (RXEN bit is set to 0 in Z1_LINCFG) and a new BREAK/SYNC field is needed to restart the receiver.

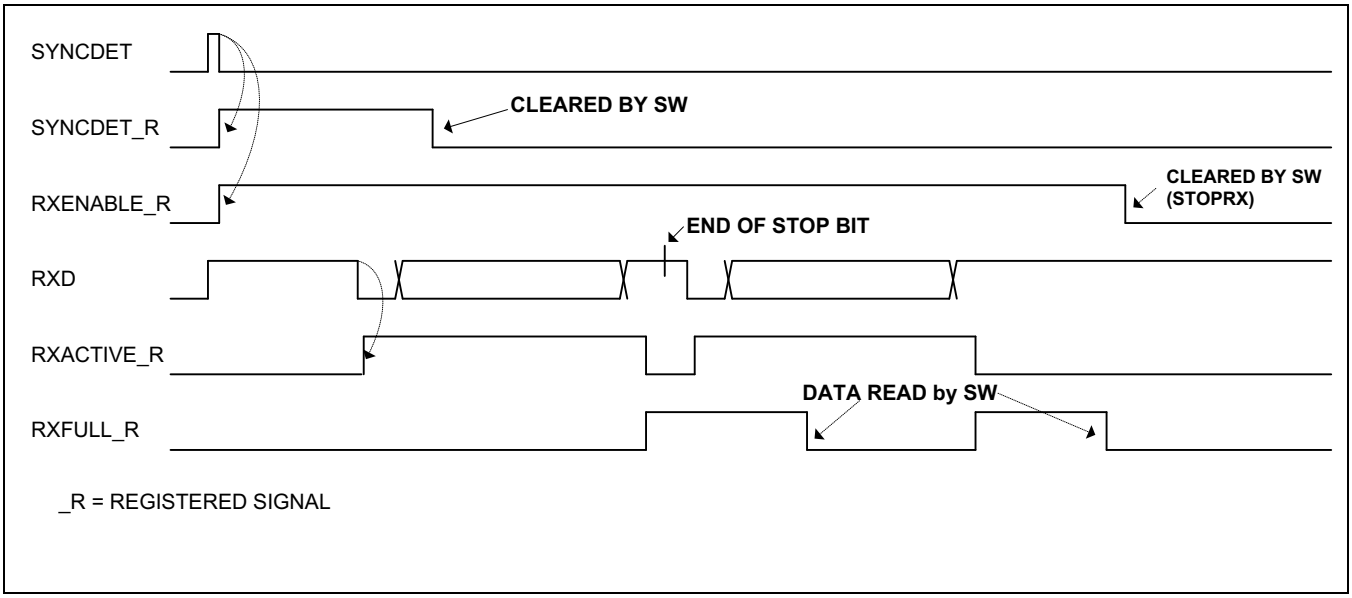
After the first byte (PID) was received, the receiver remains enabled but inactive as a DATA byte can be sent via the bus by the LIN master or another slave. In parallel, triggered by the RXFULL flag bit in the LINSTAT register, the software must read the received byte (PID) and must determine how to proceed. If the PID signals that the SW-LIN will receive data bytes, no more actions need to be taken as the receiver remains enabled. The software only has to wait for the next RXFULL interrupt bit for the successful reception of the first data byte. If the PID indicates that this module is not part of the following transfer or if the last byte (checksum) has been received, software should stop the receiver. This is done by writing 1 to the STOPRX bit in Z1_LINCFG. This register is a strobe register (cleared after one clock cycle) and is used to clear both the RXEN bit and RXACTIVE flag in Z1_LINCFG, as well as to place the RX control logic into a safe state.

© 2019 Renesas Electronics Corporation

The receiver is re-enabled again after reception of the next synchronization strobe. If the PID indicates that this SW-LIN must transmit data bytes, there is no need to disable the receiver by writing 1 to the STOPRX bit in Z1_LINCFG although it is allowed. The receiver is also disabled (and the transmitter is started) when the software writes a byte to be transmitted into the TX buffer.

There is also the capability to completely disable the receiver so that it is not even re-enabled by an incoming BREAK and SYNC field. This can be done by writing 1 to the DISABLERX bit in Z1_LINCFG. However, disabling the receiver completely must only be done when the clock divider value will be changed, the system clock will be stopped, or for debugging purposes when the LIN can operate as a transmitter only. In the first two cases, disabling the receiver is required to avoid any malfunction. In addition to the receiver being disabled, the inactive timer must also be switched off via the ENTOCNT bit in Z1_LINCFG.

Figure 4.10 RX Control and Status Signal Waveforms



4.9.5. Description of Transmit Operation

A transmission is started by writing the data to be transmitted into the TX buffer. It is in the responsibility of the software to start the transmitter only when needed (correct PID was received) taking the LIN protocol into account. When data is written into the TX buffer, the receiver is disabled (RXEN and RXACTIVE bits are set to 0 in the Z1_LINCFG register; see Table 4.25) and the transmitter is started (TXACTIVE bit is set to 1 in LINCFG). The TXEMPTY flag bit in the Z1_LINSTAT register (Table 4.26) is cleared on write access to the TX buffer. It is immediately set again in the next cycle as the data to be transmitted is copied into the shift register allowing the software to write the next byte to be transmitted into the TX buffer. If software attempts to write into the TX buffer while it already contains data (TXEMPTY bit is 0 in Z1_LINSTAT), the written byte is rejected and the WRCOLL bit in LINSTAT is set. This flag will be cleared on a READ access to the status register.

© 2019 Renesas Electronics Corporation

In a successful transmission, the transmitter first sends a 1 (STOP bit is placed at the beginning so that software does not have to take care about the length of the previously received STOP bit), then sends a 0 (START bit) and then the data byte is appended. At the end of the data byte transmission, the bus is released. If further data is present, the transmitter remains active and continues transmission. If the TX buffer is empty, the transmitter is de-activated (TXACTIVE is set to 0 in Z1_LINCFG).

Because several slaves might attempt to send data on the bus, it is possible that a conflict could occur on the bus (first error condition). This can only be detected when the module sends a 1 (recessive value) but receives a 0 (dominant value). There are three checks implemented for this:

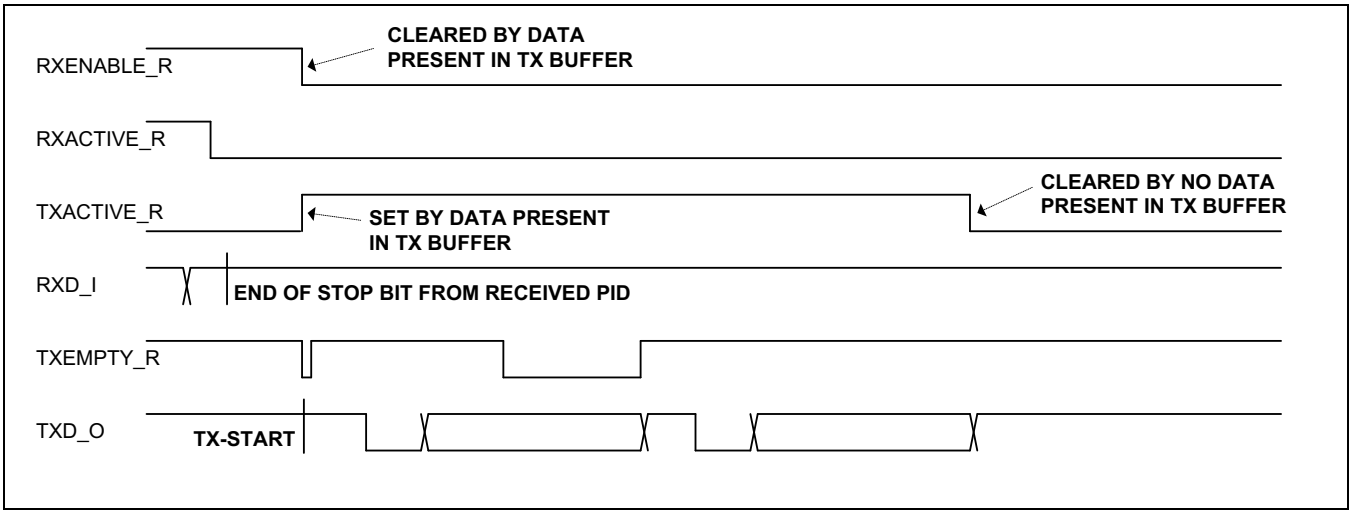
- The receiver is already activated when the transmitter will be started
- A falling edge is detected on the bus while sending the STOP bit at the beginning
- A 0 is received during the data byte while transmitting a 1

In all three cases, the transmitter is stopped (TXACTIVE bit is set to 0 in Z1_LINCFG) and the TX buffer is cleared (TXEMPTY bit is set in LINSTAT) to prevent invalid data from remaining in the buffer. The CONFLICT flag in the LINSTAT register is also set.

There is a second error condition that might occur. For some error conditions (see section 3.13), the LIN PHY can protect the bus by disabling the transmitter. This can be detected when sending a 0 but receiving a 1. This condition is checked at the end of each transmitted bit. When this error condition is detected, the transmitter is stopped (TXACTIVE bit is set to 0 in LINSTAT) and the TX buffer is cleared (TXEMPTY is set in LINSTAT) to prevent invalid data remaining in the buffer. Additionally the TXOFF flag bit in the LINSTAT register is set.

It is also possible that the master generates a BREAK and SYNC field while the SW-LIN is transmitting and that no bus conflict occurs. Therefore the transmitter is also de-activated when a sync strobe is detected.

Figure 4.11 Waveforms of the TX and RX Control and Status Signals



4.9.6. General Remarks for SW-LIN Usage

Here is a short summary of how to handle the SW-LIN:

- At power-up, the receiver is not completely disabled, but the inactivity timer is disabled.
- When the inactivity timer is needed, it has to be enabled.
- When the system clock divider will be changed, both the receiver and the inactivity timer must be disabled by writing 0x2 or 0x6 to the Z1_LINCFG register.
- When the system clock will be switched off, both the receiver and the inactivity timer must be disabled by writing 0x2 or 0x6 to the Z1_LINCFG register.
- It is in the responsibility of the software that switching to transmit happens in accordance with the LIN protocol.
- If there is no need to receive the data following the PID, software should stop the receiver by writing 0x1 or 0x5 to the Z1_LINCFG register.
- If data will be transmitted after reception of the PID, software can also stop the receiver by writing 1_{HEX} or 5_{HEX} to the Z1_LINCFG register. However, the receiver is also stopped by writing data to be transmitted into the TX buffer.
- The baud rate must not be written as it is automatically set by the BREAK/SYNC field detector. The only exception is in debugging mode, when the receiver is fully disabled and the SW-LIN will operate as a TX UART.
- The protocol is handled in software. Although the receiver is stopped and is waiting for a new sync strobe when the STOP bit is zero (this situation is possible when the MASTER changes the baud rate), it is also possible that a SYNC field is detected as a correct byte when the master has increased the baud rate and the receiver samples the data line for the STOP bit when the RXD line is high; e.g., during transmission of the SYNC field. Therefore software must ignore all received bytes (but clear the buffer) after the checksum has been received until a new synchronization has occurred (interrupt SYNCDET). To avoid this, software can also stop the receiver by writing 1 to the STOPRX bit in Z1_LINCFG. Then the receiver does not receive anything before a new synchronization has occurred.

4.9.7. SW-LIN Registers

The following registers are used by the SW-LIN module:

- Z1_LINCFG SW-LIN configuration register (Table 4.25)
- Z1_LINDATA READ/WRITE access to this register accesses the LIN RX data buffer and TX data buffer respectively (Table 4.27)
- Z1_LINIRQEN Interrupt enable register (Table 4.28)
- Z1_LINSTAT Interrupt status register (Table 4.26)
- Z1_LINBAUDHIGH:Z1_LINBAUDLOW: Baud rate value for LIN interface (Table 4.30 and Table 4.29)

For detailed register address mapping and bits description, refer to section 4.10.5.

© 2019 Renesas Electronics Corporation

4.10. MCU Registers Description

4.10.1. System Registers

The SMU registers are mapped into the system address space between 0x4000_0000 and 0x4000_03FF. Unused addresses must not be accessed.

Table 4.5 Register [0x4000_0000] SYS_CLKCFG

SYS_CLKCFG		ADD: 0x4000_0000			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
1:0	CLKDIV	R / W	R	0	Clock divider value: 0: Incoming clock is divided by 1 1: Incoming clock is divided by 2 2: Incoming clock is divided by 4 3: Incoming clock is divided by 8
6 : 2	---	R	---	0	Unused; always read as 0.
7	---	R/W	R	0	Reserved.*
31 : 8	---	R	---	0	Unused; always read as 0.

Table 4.6 Register [0x4000_0004] SYS_MEMPORTCFG

SYS_MEMPORTCFG		ADD: 0x4000_0004			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	PPNOD	R / W	R	0	Output configuration bits. It can be individually selected for each GPIO whether it functions as an open-drain (set to 0) or as a push-pull (set to 1).
15 : 8	N/A	N/A	N/A	0	Reserved.
17 : 16	N/A	N/A	N/A	0	Reserved, do not write to these bits.
20 : 18	N/A	N/A	N/A	0	Reserved, do not write to these bits.
23 : 21	N/A	N/A	N/A	0	Reserved, do not write to these bits.
29 : 24	---	R	---	0	Unused; always read as 0.
30	LINTEST	R / W	R	0	Configuration bit for LIN test. When set, RXD and TXD lines are directly connected to the GPIO.
31	MEMSWAP	R / W	R	0	Memory swap bit. It can be selected whether the FLASH MAIN area (set to 0) or the RAM (set to 1) will be mirrored to system address 0x0.

© 2019 Renesas Electronics Corporation

Table 4.7 Register [0x4000_0008] SYS_MEMINFO

SYS_MEMINFO		ADD: 0x4000_0008			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	PROGSTART	R	R / W	0xFF	This register represents the first page where the program section inside the FLASH starts. This value is read from FLASH during the power-up phase and is updated by some flash commands. Note: To determine the correct FLASH address offset, seven "0" must be appended.
15 : 8	LOGSTART	R	R / W	0xFF	This register represents the first page where the log section inside the FLASH starts. This value is read from the FLASH during the power-up phase and is updated by some FLASH commands. Note: To determine the correct FLASH address offset, seven "0" must be appended.
26 : 16	RAMSPLIT	R	R / W	0x7FF	This register represents the first RAM word that is accessible by JTAG although the memory is locked. This value is read from FLASH during the power-up phase and is updated by some FLASH commands. Note: To determine the correct RAM address offset, two "0" must be appended.
27	---	R	---	0	Unused; always read as 0.
31 : 28	PROTINFO	R	R / W	0xF	This register represents the memory protection scheme. This value is read from FLASH during the power-up phase and is updated by some FLASH commands. [0]: key-based lock [1]: permanent lock [3:2]: number of failed unlock attempts

Table 4.8 Register [0x4000_000C] SYS_RSTSTAT

SYS_RSTSTAT		ADD: 0x4000_000C			
Bit	**NAME**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	EXTRST	RC	---	1	This bit reflects if an external reset has occurred. It is cleared when the register is read.
1	SYSRSTREQ	RC	W	0	This bit reflects if a reset forced by a system reset request has occurred. It is cleared when the register is read.
2	LOCKUPRST	RC	W	0	This bit reflects if a reset was forced by a detected lockup when this reset was enabled. It is cleared when the register is read.
3	JTAGRST	RC	W	0	This bit reflects if a reset that was forced by a JTAG reset request has occurred. It is cleared when the register is read.
6 : 4	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

SYS_RSTSTAT		ADD: 0x4000_000C			
Bit	**NAME**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7	ENLOCKUP	R	R	0	This bit reflects whether a lockup from the ARM® core is allowed to reset the system (set to 1) or not (set to 0).
7 : 0	SETENLOCKUP	W	---	0	To enable the lockup reset, 0xC9 must be written to bits 7:0. It can be disabled by writing another value. This bit is reset by all four reset sources (extRst, sysRstReq, lockupRst, jtagRst).
15:8	---	R	---	0	Unused; always read as 0.
23 : 16	JTAGRSTREQ	W	R	0	To generate a reset via the JTAG interface, 0x3A must be written to bits 23:16. These bits cannot be written by the ARM® core.
31 : 24	---	R	---	0	Unused; always read as 0.

4.10.2. GPIO Registers

The GPIO registers are mapped into the system address space between 0x4000_1400 and 0x4000_17FF. Unused addresses must not be accessed.

Table 4.9 Register [0x4000_1400] GPIO_DIR

GPIO_DIR		ADD: 0x4000_1400			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	GPIODIR	R / W	R	0	1: GPIO pin is switched as output direction. 0: GPIO pin is switched as input direction.
15 : 8	---	N/A	N/A	0	Reserved.
31 : 16	---	R	---	0	Unused; always read as 0.

Table 4.10 Register [0x4000_1404] GPIO_IN

GPIO_IN		ADD:0x4000_1404			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	GPIOIN	R	R / W	0	Synchronized input value.
15 : 8	---	N/A	N/A	0	Reserved.
31 : 8	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

Table 4.11 Register [0x4000_1408] GPIO_OUT

GPIO_OUT		ADD:0x4000_1408			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	GPIOOUT	R / W	R	0	Value to be driven out of each GPIO.
15 : 8	---	N/A	N/A	0	Reserved.
31 : 16	---	R	---	0	Unused; always read as 0.

Table 4.12 Register [0x4000_140C] GPIO_SETCLR

GPIO_SETCLR		ADD:0x4000_140C			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	SETOUT	R / W	R	0	Writing a 1 to the corresponding bit will set the GPIO output to 1. Lower priority than clear. Always read as 0.
15 : 8	---	N/A	N/A	0	Reserved.
23 : 16	CLROUT	R / W	R	0	Writing a 1 to the corresponding bit will set the GPIO output to 0. Higher priority than set. Always read as 0.
31 : 24	---	N/A	N/A	0	Reserved.

Table 4.13 Register [0x4000_1410] GPIO_IRQSTAT

GPIO_IRQSTAT		ADD:0x4000_1410			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	IRQSAT	RC	R/W	0	This register reflects the interrupt status of each GPIO that is enabled as interrupt. Reading this register will clear the interrupt status bit.
15 : 8	---	N/A	N/A	0	Reserved.
31 : 16	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

Table 4.14 Register [0x4000_1414] GPIO_IRQEN

GPIO_IRQEN		ADD:0x4000_1414			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	IRQEN	R / W	R	0	When set to 1, the corresponding interrupt is allowed to drive the interrupt line when the appropriate edge occurs.
15 : 8	---	N/A	N/A	0	Reserved.
31 : 16	---	R	---	0	Unused; always read as 0.

Table 4.15 Register [0x4000_1418] GPIO_IRQEDGE

GPIO_IRQEDGE		ADD:0x4000_1418			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	IRQEDGE	R / W	R	0	0: rising edge on the corresponding GPIO triggers the IRQ line 1: falling edge on the corresponding GPIO triggers the IRQ line
15 : 8	---	N/A	N/A	0	Reserved.
31 : 16	---	R	---	0	Unused; always read as 0.

Table 4.16 Register [0x4000_141C] GPIO_TRIGEN

GPIO_TRIGEN		ADD:0x4000_141C			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	TRIGEN	R / W	R	0	When set to 1, the corresponding GPIO drives its trigger line.
15 : 8	---	N/A	N/A	0	Reserved.
31 : 16	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

4.10.3. 32 Bit Timer Registers

The 32-bit timer registers are mapped into the system address space between 0x4000_1000 and 0x4000_13FF. Unused addresses must not be accessed.

Table 4.17 Register [0x4000_1000] T32_CTRL

T32_CTRL		ADD:0x4000_1000			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	EN	R / W	R / W	0	Enable bit for timer. This bit is cleared by hardware if an overflow occurs and the module is operating in single shot mode.
1	MODETC	R / W	R	0	Select between timer and counter mode: 0: timer mode 1: counter mode
2	MODESR	R / W	R	0	Select between reload and single shot mode: 0: reload mode; at overflow, reload value is copied into counter register and counter continues 1: single shot mode; at overflow, reload value is copied into counter register and counter stops
3	MODELE	R / W	R	0	Select between level or edge sensitive trigger; counter mode only: 0: trigger is used as level sensitive 1: trigger is used as edge sensitive
4	MODEPN	R / W	R	0	Selects between rising or falling edge active trigger (MODELE = 1) or high or low level (MODELE = 0); counter mode only 0: Trigger on falling edge/low level 1: Trigger on rising edge/high level
5	OVERFLOW	R	R / W	0	Overflow flag (strobe); set for a single cycle when counter overflows. This bit also drives the interrupt line.
31 : 6	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

Table 4.18 Register [0x4000_1004] T32_TRIGSEL

T32_TRIGSEL		ADD:0x4000_1004			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
4 : 0	TRIGSEL	R / W	R	0	Select signal for the trigger source: 0x00: no trigger source 0x01: GPIO00 is used as trigger source 0x02: GPIO01 is used as trigger source … 0x07: GPIO07 is used as trigger source 0x08 to 0x1F: no trigger source
31 : 5	---	R	---	0	Unused; always read as 0.

Table 4.19 Register [0x4000_1008] T32_CNT

T32_CNT		ADD:0x4000_1008			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
31 : 0	COUNTER	R / W	R / W	0	Timer value; this register can be written directly regardless of whether timer is enabled or not. It is set to the reload value when the reload value is written.

Table 4.20 Register [0x4000_100C] T32_REL

T32_REL		ADD:0x4000_100C			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
31 : 0	RELOADVAL	R / W	R	0	Timer-reload value; when the timer (counter) overflows, the reload value is copied into the counter register. In reload mode, the timer continues; however it stops if reload mode is not enabled.

© 2019 Renesas Electronics Corporation

4.10.4. SPI Registers

The registers of the SPI module are mapped into the system address space between 0x4000_1820 and 0x4000_1BFF. Unused addresses must not be accessed.

Table 4.21 Register [0x4000_1820] Z1_SPICFG

Z1_SPICFG		ADD:0x4000_1820			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	RXOVERFLOW	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
1	RXFULL	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
2	TXEMPTY	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
3	WRCOLL	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
4	---	R	---	0	Unused; always read as 0.
5	SAMPLEPOS	R / W	R	0	This bit selects whether data on MISO will be sampled at the sampling edge (set to 0) or at shift edge (set to 1). Note: Change this bit only when the module is disabled (SPIEN = 0) or when no transfer is in progress.
6	SSN	R / W	R / W	1	This bit directly controls the SSN line.
7	SPIEN	R / W	R / W	0	Enable for SPI module.
31 : 8	---	R	---	0	Unused; always read as 0.

Table 4.22 Register [0x4000_1824] Z1_SPIDATA

Z1_SPIDATA		ADD:0x4000_1824			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	SPIDATA	R / W	R / W	0	When writing a byte to this register, the value is stored in the TxBuffer. Additionally a WRITE access to this register clears the TXEMPTY flag in the Z1_SPISTAT register (Table 4.24). When reading this register, the contents of the RxBuffer is returned. Additionally a READ access to this register clears the RXFULL flag in Z1_SPISTAT. Note: When writing to this register when TxBuffer is full, the TxBuffer keeps its contents and the written byte is rejected. This is signaled by the WRCOLL flag in Z1_SPISTAT.
31 : 8	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

Table 4.23 Register [0x4000_1828] Z1_SPICLKCFG

Z1_ SPICLKCFG		ADD:0x4000_1828			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	CPOL	R / W	R	1	Clock polarity; the content of this bit directly affects the idle state of the clock. Note: Change this bit only when the module is disabled (SPIEN=0 in register Z1_SPICFG; see Table 4.21).
1	CPHA	R / W	R	1	Clock phase; data is centered on the first (set to 0) or to the second (set to 1) clock edge. Note: Change this bit only when the module is disabled (SPIEN=0).
7 : 2	CDIV	R / W	R	1	Clock divider value; SPI clock period is 2∗(CDIV+1) times the system clock. Note: Change this bit only when the module is disabled (SPIEN=0) or when no transfer is in progress.
31 : 8	---	R	---	0	Unused; always read as 0.

Table 4.24 Register [0x4000_182C] Z1_SPISTAT

Z1_ SPISTAT		ADD:0x4000_182C			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	RXOF	RC	W	0	This bit signals that an RX overflow has occurred. Note: This bit is cleared when the status is read. Note: The received byte causing the overflow is rejected; the previous received byte is kept in the RX buffer.
1	RXFULL	R	R / W	0	This bit reflects the status of the RX buffer. It is set when a new byte is transferred into the RX buffer. Note: This bit is cleared when SPIDATA is read (Table 4.22).
2	TXEMPTY	R	R / W	1	This bit reflects the status of the TX buffer. It is set when a byte is transferred from the TX buffer into the shift register. Note: This bit is cleared when SPIDATA is written.
3	WRCOLL	RC	W	0	This bit is set when SPIDATA is written while TX buffer is already full. Note: This bit is cleared when the status is read.
4	BUSY	R	R / W	0	This bit reflects the status of the SPI module.
6 : 5	---	R	---	0	Unused; always read as 0.
7	CLRTXBUF	W1C	R	0	Writing a 1 to this bit clears the TX buffer. Note: Write only when SPI is disabled; always read as 0.
31 : 8	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

4.10.5. SW-LIN Registers

The registers of the SW-LIN module are mapped into the system address space between 0x4000_1800 and 0x4000_181F. Unused addresses must not be accessed.

Table 4.25 Register [0x4000_1800] Z1_LINCFG

Z1_LINCFG		ADD:0x4000_1800			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	STOPRX	R / W	R	0	This bit is a strobe register to stop the receiver. When writing 1 to this bit, the state machine of the data reception unit is placed into its default state (RXEN and RXACTIVE are cleared) and waits for a new sync strobe. This can be used by software to reject incoming data after the PID field has been evaluated as a non-used PID.
1	DISABLERX	R / W	R	0	When set to 1, this bit completely disables the BREAK/SYNC field detector, the RX data path, and the RX filter. It must be set when the system clock will be switched off, when the clock divider will be changed, when LIN will go to sleep but the MCU will continue operation, or for debugging purposes.
2	FASTMODE	R / W	R	0	This bit distinguishes between slow (0) and fast (1) mode. In slow mode, the detected baud rate is between 1kBaud and 30kBaud. In fast mode, the detected baud rate is up to 200KBaud for CLKDIV values 0, 1, and 2 and up to 100KBaud for CLKDIV value of 3 (see Table 4.5).
3	ENTOCNT	R / W	R	0	This bit enables the timeout counter for bus inactivity. The LIN protocol requires that the LIN controller must go to sleep when either a SLEEP command was received or after more than 4s of inactivity. If the overall system guarantees that LIN communication is always stopped with a SLEEP command, there is no need to activate the timeout counter.
4	RXEN	R	R / W	0	This bit reflects the status of the receiver. It is set when a BREAK/SYNC field is detected and it is cleared when software stops the receiver (STOPRX = 1) or when a transmission is started.
5	RXACTIVE	R	R / W	0	This bit reflects the status of the receiver. It is set when a START condition is detected on the bus (falling edge of RXD line after synchronization). It is cleared when the software stops the receiver (STOPRX = 1), when a new sync strobe occurs, in the middle of a received STOP bit, or when a transmission is started.
6	TXACTIVE	R	R / W	0	This bit reflects the status of the transmitter. It is set if data is present in the TX buffer to be transmitted. It is cleared if a sync strobe or a bus conflict (sending 1 but receiving 0) is detected, if the transmitter is forced to 1 by protection logic (sending 0 but receiving 1), or when no more data to be transmitted is present in the TX buffer at the end of an active transmission.

© 2019 Renesas Electronics Corporation

Z1_LINCFG		ADD:0x4000_1800			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
31 : 7	---	R	---	0	Unused; always read as 0.

Table 4.26 Register [0x4000_1804] Z1_LINSTAT

Z1_LINSTAT		ADD:0x4000_1804			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	SYNCDET	RC	R / W	0	This bit is set by hardware when a BREAK/SYNC field has been detected by the break-/sync-field detector. It is cleared when software reads this register.
1	RXFULL	R	R / W	0	This bit is set by hardware when a received byte is placed into the RX buffer. It is cleared when the software reads the data out of the RX buffer (read from LINDATA;Table 4.27).
2	TXEMPTY	R	R / W	1	This bit is set when a byte to be transmitted is read out of the TX buffer to be shifted out on the LIN bus. It is cleared when the software puts a new byte into the TX buffer (write to LINDATA).
3	CONFLICT	RC	R / W	0	This bit is set by hardware when it has detected a bus conflict during an active transmission. This can only occur when sending a 1 but receiving a 0. It is cleared when software reads this register.
4	RXOVERFLOW	RC	R / W	0	This bit is set by hardware if the receiver attempts to place a received byte into the RX buffer while this buffer is already full. The new received byte is rejected and lost. This bit is cleared when software reads this register.
5	WRCOLL	RC	R / W	0	This bit is set by hardware if software tries to write a byte into the TX buffer while this buffer is already full. The new written byte is rejected. This bit is cleared when the software reads this register.
6	TXOFF	RC	R / W	0	This bit is set by hardware if it is transmitting a 0 but receives a 1. As 0 is the dominant level on the bus; this can only occur due to hardware error or if the bus protection circuit has disabled the output driver in the LIN PHY. This bit is cleared when software reads this register.
7	INACTIVE	RC	R / W	o	This bit is set when the timeout counter for inactivity has expired. This bit is cleared when software reads this register.

© 2019 Renesas Electronics Corporation

Table 4.27 Register [0x4000_1808] Z1_LINDATA

Z1_LINDATA		ADD:0x4000_1808			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	LINDATA	R / W	R / W	0	When writing a byte to this register, the value is stored in the TX buffer. A WRITE access to this register also clears the TXEMPTY flag in the Z1_LINSTAT register (Table 4.26). When reading this register, the content of the RX buffer is returned. A READ access to this register also clears the RXFULL flag in the Z1_LINSTAT register. Note: When writing to this register if TX buffer is full, the TX buffer keeps its contents and the written byte is rejected. This is signaled by the WRCOLL flag in the Z1_LINSTAT register.
31 : 8	---	R	---	0	Unused; always read as 0.

Table 4.28 Register [0x4000_180C] Z1_LINIRQEN

Z1_LINIRQEN		ADD:0x4000_180C			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	SYNCDET	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
1	RXFULL	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
2	TXEMPTY	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
3	CONFLICT	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
4	RXOVERFLOW	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
5	WRCOLL	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
6	TXOFF	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
7	INACTIVE	R / W	R	0	When set to 1, the corresponding status bit is allowed to drive the IRQ output.
31 : 8	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

Table 4.29 Register [0x4000_1810] Z1_LINBAUDLOW

Z1_LINBAUDLOW		ADD:0x4000_1810			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
7 : 0	LINBAUDLOW	R / W	R / W	0xE7	Baud rate for LIN interface *baud rate = CLK/({LINBAUDHIGH, LINBAUDLOW} + 1)* Note: In NORMAL operating mode, this register must not be written. It is updated by the BREAK/SYNC detector. Note: For debugging purposes, the LIN controller can function as a TX UART. For this purpose, the baud rate can be selected by software but must be at least 0x3.
31 : 8	---	R	---	0	Unused; always read as 0.

Table 4.30 Register [0x4000_1814] Z1_LINBAUDHIGH

Z1_LINBAUDHIGH		ADD:0x4000_1814			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
6 : 0	LINBAUDHIGH	R / W	R / W	0x3	Baud rate for LIN interface *baud rate = clk / ({LINBAUDHIGH, LINBAUD LOW} + 1)* Note: In NORMAL operating mode, this register must not be written. It is updated by the BREAK/SYNC detector. Note: for debugging purposes, the LIN controller can function as a TX UART. For this purpose, the baud rate can be selected by software but must be at least 0x3.
31 : 7	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

4.10.6. FLASH Memory Registers

The registers of the FLASH controller are mapped into the system address space between 0x4000_0800 and 0x4000_0BFF. Unused addresses must not be accessed. Read accesses to registers are always performed even if a command is executed or a direct WRITE is active. Write accesses to the registers are postponed until the active command/direct-write access has finished.

Table 4.31 Register [0x4000_0800] FC_RAM_ADDR

FC_RAM_ADDR		ADD:0x4000_0800			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
10 : 0	ADDRRAM	R / W	R / W	0	RAM word address where data to be written into flash is located (first address); hardware increments this address when more than one word has to be written. Note: The last two address digits of the RAM are not included as they are always 0. For programming, the RAM address needs to be shifted right by two bits.
31 : 11	---	R	---	0	Unused; always read as 0.

Table 4.32 Register [0x4000_0804] FC_FLASH_ADDR

FC_FLASH_ADDR		ADD:0x4000_0804			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
14 : 0	FLASHADDR	R / W	R	0	FLASH word address (first address) where data is to be written; also used as the pointer to a page to be erased. Note: the last two address digits of the FLASH are not included as they are always 0. For programming, the FLASH address needs to be shifted right by nine bits.
31 : 15	---	R	---	0	Unused; always read as 0.

Table 4.33 Register [0x4000_080C] FC_EXE_CMD

FC_EXE_CMD		ADD:0x4000_080C			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	EXECMD	R / W	R	0	Writing 1 to this bit starts the execution of the configured command; always read as 0.
31 : 1	---	R	---	0	Unused; always read as 0.

© 2019 Renesas Electronics Corporation

Table 4.34 Register [0x4000_0808] FC_CMD_SIZE

FC_CMD_SIZE		ADD:0x4000_0808			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
3 : 0	COMMAND	R / W	R	0	Command to be executed by flash controller Valid commands: 0_{HEX}: ERASE_MAIN_CMD 2_{HEX}: ERASE_BOOT_PROG_CMD 3_{HEX}: ERASE_PROG_CMD 4_{HEX}: ERASE_MAINPAGE_CMD 6_{HEX}: ERASE_KEY_CMD 8_{HEX}: UNLOCK_CMD 9_{HEX}: GETENV_CMD A_{HEX}: WRITE_CMD C_{HEX}: SET_KEY_CMD D_{HEX}: SET_BOUNDARY_CMD E_{HEX}: LOCK_PERM_CMD F_{HEX}: LOCK_KEY_CMD
7:4	---	R	---	0	Unused; always read as 0.
12:8	WRSIZE	R / W	R	0	Number of words to be written to FLASH; 0 is interpreted as 32. Note: Writing is always performed within a row. 1 row contains 32 words. While the RAM address is always incremented, the FLASH address wraps at the row boundary to the beginning of the row. It is in the responsibility of the user to ensure this.
31 : 13	---	R	---	0	Unused; always read as 0.

Table 4.35 Register [0x4000_0810] FC_IRQ_EN

FC_IRQ_EN		ADD:0x4000_0810			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
0	ENIRQ0	R / W	R	0	When set to 1, the status signal CMDRDY in the FC_STAT_CORE register is allowed to drive the interrupt line. See Table 4.36 for the related status signals for bits [3:0] this register.
1	ENIRQ1	R / W	R	0	When set to 1, the status signal INVALIDCMD is allowed to drive the interrupt line.
2	ENIRQ2	R / W	R	0	When set to 1, the status signal INVALIDAREA is allowed to drive the interrupt line.
3	ENIRQ3	R / W	R	0	When set to 1, the status signal UNLOCKFAIL is allowed to drive the interrupt line.
4	ENIRQ4	R / W	R	0	When set to 1, the status signal DATAALL1 in the FC_STAT_DATA register is allowed to drive the interrupt line. See Table 4.38 for the related status signals for bits [6:4] this register.
5	ENIRQ5	R / W	R	0	When set to 1, the status signal DATA1ERR is allowed to

© 2019 Renesas Electronics Corporation

FC_IRQ_EN		ADD:0x4000_0810			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
					drive the interrupt line.
6	ENIRQ6	R / W	R	0	When set to 1, the status signal DATA2ERR is allowed to drive the interrupt line.
7	ENIRQ7	R / W	R	0	When set to 1, the status signal PROG1ERR in the FC_STAT_PROG register (Table 4.37) is allowed to drive the interrupt line.
31 : 8	---	R	---	0	Unused; always read as 0.

Table 4.36 Register [0x4000_0814] FC_STAT_CORE

FC_STAT_CORE		ADD:0x4000_0814			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description** [1)]
0	CMDRDY	RC	R / W	0	This bit is set when a command execution has finished; it is cleared when this register is read.
1	INVALIDCMD	RC	R / W	0	This bit is set when an invalid command was executed; it is cleared when this register is read.
2	INVALIDAREA	RC	R / W	0	This bit is set when a command is executed targeting a protected area; e.g., performing ERASE_MAINPAGE_CMD to program space when flash is locked; it is cleared when this register is read.
3	UNLOCKFAIL	RC	R / W	0	This bit is set when the UNLOCK_CMD fails; it is cleared when this register is read.
4	COREACTIVE	R	R / W	0	This bit reflects the status of the core state machine; when set core is active.
5	ALLOWKEY	R	R / W	0	When this bit is set but FLASH is locked, the ERASE_KEY_CMD is allowed to be performed.
6	ALLOWBOOT	R	R / W	0	When this bit is set but FLASH is locked, the WRITE_CMD is allowed to be performed on the boot space.
7	ALLOWPROG	R	R / W	0	When set but FLASH is locked, the WRITE_CMD is allowed to be performed on the program space.
8	CLRALLOW	W1C	R	0	Writing 1 to this bit clears ALLOWKEY, ALLOWBOOT, and ALLOWPROG flags.
31 : 9	---	R	---	0	Unused; always read as 0.

1) For related FLASH controller command codes, see Table 4.34.

Table 4.37 Register [0x4000_0818] FC_STAT_PROG

FC_STAT_PROG		ADD:0x4000_0818			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
17 : 0	ADDRPROG	R	R / W	0	This register contains the address of the instruction fetch error that caused the first of the three flags below to be set; the highest bit is used to distinguish between MAIN (0) and INFO (1) area.
28 : 18	---	R	---	0	Unused; always read as 0.
29	PROGALL1	RC	R / W	0	This bit is set when an instruction fetch occurs to an erased memory address; it is cleared when this register is read.
30	PROG1ERR	RC	R / W	0	This bit is set when a correctable error occurs during an instruction fetch; it is cleared when this register is read.
31	PROG2ERR	RC	R / W	0	This bit is set when an un-correctable error occurs during an instruction fetch; it is cleared when this register is read.

Table 4.38 Register [0x4000_081C] FC_STAT_DATA

FC_STAT_DATA		ADD:0x4000_081C			
Bit	**Name**	**Ext. Access**	**Int. Access**	**Reset Value**	**Description**
17 : 0	ADDRDATA	R	R / W	0	This register contains the address of the read error which caused the first of the three flags below to be set; the highest bit is used to distinguish between MAIN (0) and INFO (1) area.
28 : 18	---	R	---	0	Unused; always read as 0.
29	DATAALL1	RC	R / W	0	This bit is set when a READ is performed to an erased memory address; it is cleared when this register is read.
30	DATA1ERR	RC	R / W	0	This bit is set when a correctable error occurs during a read; it is cleared when this register is read.
31	DATA2ERR	RC	R / W	0	This bit is set when an un-correctable error occurs during a READ; it is cleared when this register is read.

© 2019 Renesas Electronics Corporation

5 ZAMC4100 Protection and Diagnostic Features

This section is a summary description of all protection and diagnostic features available in the ZAMC4100. The description is structured similarly to the way the diagnostics are handled:

event → detection → capturing → reaction

5.1. Full-Time Diagnostic and Protection Features

It includes features that might always be present whenever the system is powered and do not interrupt or influence the normal operation. They are executed in the background automatically and interrupt the normal operation only if a specific event occurs. Some of these features might be masked by the software and MCU; others cannot be masked (mainly protection functions).

5.1.1. Over-Current (OC) Protection

Event The current through one of the output driver transistors is above the maximum specified value for the corresponding output.

Detection Continuous comparison with a fixed reference level with pure analog circuitry; possible measuring with the ADC in parallel; invoked by the software.

Capturing Captured in the digital SBC.

Reaction Response in both digital and analog domains:

1) The driver of this transistor automatically limits the current in order to avoid silicon damage. **Important:** This reaction is implemented as analog circuitry, and it cannot be masked.
2) An over-current interrupt flag is set in the digital SBC, and the drivers are automatically disabled.

5.1.2. Over-Load (OVL) Detection using the ADC

Event The current through one of the output driver transistors is above a certain limit (OVL level) that is considered as a maximum current allowed for the driven load. The OVL level is always below the OC level.

Detection Initiated by the MCU firmware. The measured current by the ADC is compared with the OVL level in the MCU or with the ADC comparator.

Capturing The event is captured in the MCU.

Reaction A response should be implemented in the MCU software depending on the application requirements.

© 2019 Renesas Electronics Corporation

5.1.3. Over-Temperature (OVT) Protection

Event The internal chip temperature exceeds the maximum specified junction temperature for the system (see Table 1.2).

Detection Continuous comparison between the internal reference and the internal temperature sensor's voltage.

Capturing The event is captured in the SBC, and an interrupt is propagated to the MCU.

Reaction All drivers are automatically disabled.

5.1.4. Over-Voltage (OV) Detection

Event The voltage at the VDDE pin is higher than the upper supply range limit specified in Table 1.5.

Detection Continuous comparison of the internal reference and VDDE pin voltage.

Capturing The event is captured in the SBC, and an interrupt is propagated to the MCU.

Reaction All drivers are automatically disabled. This feature can be turned off if not needed.

5.1.5. Under-Voltage (UV) Detection

Event The voltage at the VDDE pin is higher than the lower supply range limit specified in Table 1.5.

Detection Continuous comparison of the internal reference and VDDE pin voltage.

Capturing The event is captured in the SBC, and an interrupt is propagated to the MCU.

Reaction All drivers are automatically disabled. This feature can be switched-off if not needed.

5.1.6. Short to GND or Vbat Protection of Sensor Supply Pin S_P and Analog Input Pins AIN1 and AIN2

Event The pin is shorted to the GND or Vbat rail (for the AIN1 and AIN2 pins through the external protection resistors; see the application diagram in Figure 6.1).

Detection A Zener-like structure limits the voltage propagated through these pins to internal blocks.

Capturing The event is not captured.

Reaction Limit the voltage in order to protect the internal circuitry, or limit the current through the S_P pin in the event of a short to ground.

5.1.7. OSCL Output Clock Failure Detection

Event There is a failure in the OSCL clock generation. The 125kHz clock is missing.

Detection Internal circuitry running at the OSCH clock continuously observes the output of OSCL. If OSCL stops, a failure event is detected.

Capturing The event is captured in the SBC, and an interrupt is propagated to the MCU.

Reaction A response should be implemented in the MCU software depending on application requirements.

© 2019 Renesas Electronics Corporation

5.2. MCU Activated Diagnostic Functions

All hardware resources needed for these diagnostics are available in the SBC, but the procedures must be executed by the MCU.

5.2.1. Open-Load Detection

Event There is an open in one or more drivers' outputs.

Detection A current source is connected to the output and the voltage drop over the load is observed. If the load is open, the voltage drop is zero. More information for this procedure is given in sections 3.8.3, 3.9.3, and 3.11.3. Also see Table 1.8 and Table 1.9 for electrical specifications.

Capturing The event is captured as diagnostic status flag in SBC chip and can be read via the SPI.

Reaction Should be implemented in MCU software depending on the application requirements.

5.2.2. Short to Vbat or GND Detection

Event The driver output is shorted to Vbat or GND.

Detection Two current sources (sink and source) are connected to the output and the voltage drop over them is observed. If an event occurs, the potential will not be changed (zero in the event of a short to GND or Vbat in the event of a short to Vbat). More information for this procedure is given in sections 3.8.3, 3.9.3, and 3.11.3. See Table 1.8 and Table 1.9 for electrical specifications.

Capturing The event is captured as a diagnostic status flag in the SBC and can be read via the SPI.

Reaction A response should be implemented in MCU software depending on the application requirements.

Note: The short to Vbat or GND diagnostic is possible only for the half-bridge drivers. For high-side drivers, only short to VBAT can be detected independently. Short to GND will cause an over-current event.

5.2.3. Floating Analog Input Detection

Event One of the analog inputs (AINx pins) is floating or shorted.

Detection Current sink and source are connected to the pin and the voltage drop over them is monitored. More information is given in section 3.12.10. See Table 1.13 for electrical specifications.

Capturing The event is captured as a diagnostic status flag in the SBC and can be read via the SPI.

Reaction A response should be implemented in the MCU software depending on application requirements.

© 2019 Renesas Electronics Corporation

5.2.4. Short or Open Sensor Supply Detection

Event The sensor supply pin S_P is shorted to one of the supply rails, or it is floating.

Detection Current sink and source are connected to the pin and the voltage drop over them is monitored. More information is given in section 3.12.10. See for Table 1.13 for electrical specifications.

Capturing The event is captured as a diagnostic status flag in the SBC and can be read via the SPI.

Reaction A response should be implemented in the MCU software depending on application requirements.

5.2.5. ADC Ratiometric Test

Event There is a failure in the ADC analog or digital control/data path.

Detection The input multiplexer connects the input of the ADC to fixed points from the resistive divider of the position potentiometers' supply voltage. The measurement is ratiometric and does not depend on internal reference sources. The conversion result is compared with the expected one.

Details for software implementation:

ADC channel selected by the ADCCTRL register:	IMPSEL[4:0] = 19_{HEX} (i.e., channel 25_{DEC})
Expected ADC conversion result:	ADCRES[9:0] = 682 ± 15 (unsigned result)
ADC channel selected by ADCCTRL register:	IMPSEL[4:0] = $1A_{HEX}$
Expected ADC conversion result:	ADCRES[9:0] = 341 ± 15 (unsigned result)

Notes:

1. For proper execution of this test, the internal sensor supply should be enabled (set bits SPSWEN and SNSWEN in AINCONF register) or the external sensor supply should be applied between the S_P and S_N pins.
2. The expected conversion results are specified for maximum sampling rate: 125ksps. At higher speeds, the ADC conversion results obtained will be out of the specified range.

Capturing The event is captured in the MCU by reading the ADC conversion value.

Reaction A response should be implemented in the MCU software depending on application requirements.

5.2.6. ADC Test Absolute

Event There is a failure in the ADC analog or digital control/data path.

Detection The ADC input is connected to the internal reference voltage. The value of this voltage is Vref/2 and the ADC should return the midscale value.

Details for software implementation:

ADC channel selected by ADCCTRL register:	IMPSEL[4:0] = $1B_{HEX}$
Expected ADC conversion result:	ADCRES[9:0] = 512 ± 15 (unsigned result)

Capturing The event is captured in the MCU by reading the ADC conversion value.

Reaction A response should be implemented in the MCU software depending on application requirements.

© 2019 Renesas Electronics Corporation

5.2.7. Reference Voltage Test

Event There is failure in the ADC reference voltage.

Detection The ADC input is connected to the internal reference voltage (this is not the ADC reference). The value of this voltage is expected to be Vref/2 and the ADC should return the midscale value.

Details for software implementation:

ADC channel selected by the ADCCTRL register:	IMPSEL[4:0] = $1C_{HEX}$
Expected ADC conversion result:	ADCRES[9:0] = 512 ± 30 (unsigned result)

Capturing The event is captured in the MCU by reading the ADC conversion value.

Reaction A response should be implemented in the MCU software depending on application requirements.

5.2.8. 5.0V Supply Test

Event Regulator VDD5V0 voltage is out of range.

Detection The regulator's output is measured with the ADC. The expected result is the ADC midscale value.

Details for software implementation:

ADC channel selected by the ADCCTRL register:	IMPSEL[4:0] = $1D_{HEX}$
Expected ADC conversion result:	ADCRES[9:0] = 512 ± 50 (unsigned result)

Capturing The event is captured in the MCU by reading the ADC conversion value.

Reaction A response should be implemented in the MCU software depending on application requirements.

5.2.9. 2.5V Supply Test

Event Regulator VDD2V5 voltage is out of range.

Detection The regulator's output is measured with the ADC. The expected result is the ADC midscale value.

Details for software implementation:

ADC channel selected by the ADCCTRL register:	IMPSEL[4:0] = $1E_{HEX}$
Expected ADC conversion result:	ADCRES[9:0] = 512 ± 50 (unsigned result)

Capturing The event is captured in the MCU by reading the ADC conversion value.

Reaction A response should be implemented in the MCU software depending on application requirements.

© 2019 Renesas Electronics Corporation

5.2.10. Current Multiplexer Test

Event There is a failure in the current multiplexer or the current to voltage converter

Detection A dedicated diagnostic channel of the current multiplexer with fixed current level is connected to the ADC. The expected result is 10% of the ADC unsigned full-scale range.

Details for software implementation:

ADC channel selected by ADCCTRL register: IMPSEL[4:0] = 1Fh

Expected ADC conversion result: ADCRES[9:0] = 102 ± 10 (unsigned result)

Capturing The event is captured in MCU by reading the ADC conversion value

Reaction Should be implemented in MCU software depending on the application requirements.

© 2019 Renesas Electronics Corporation

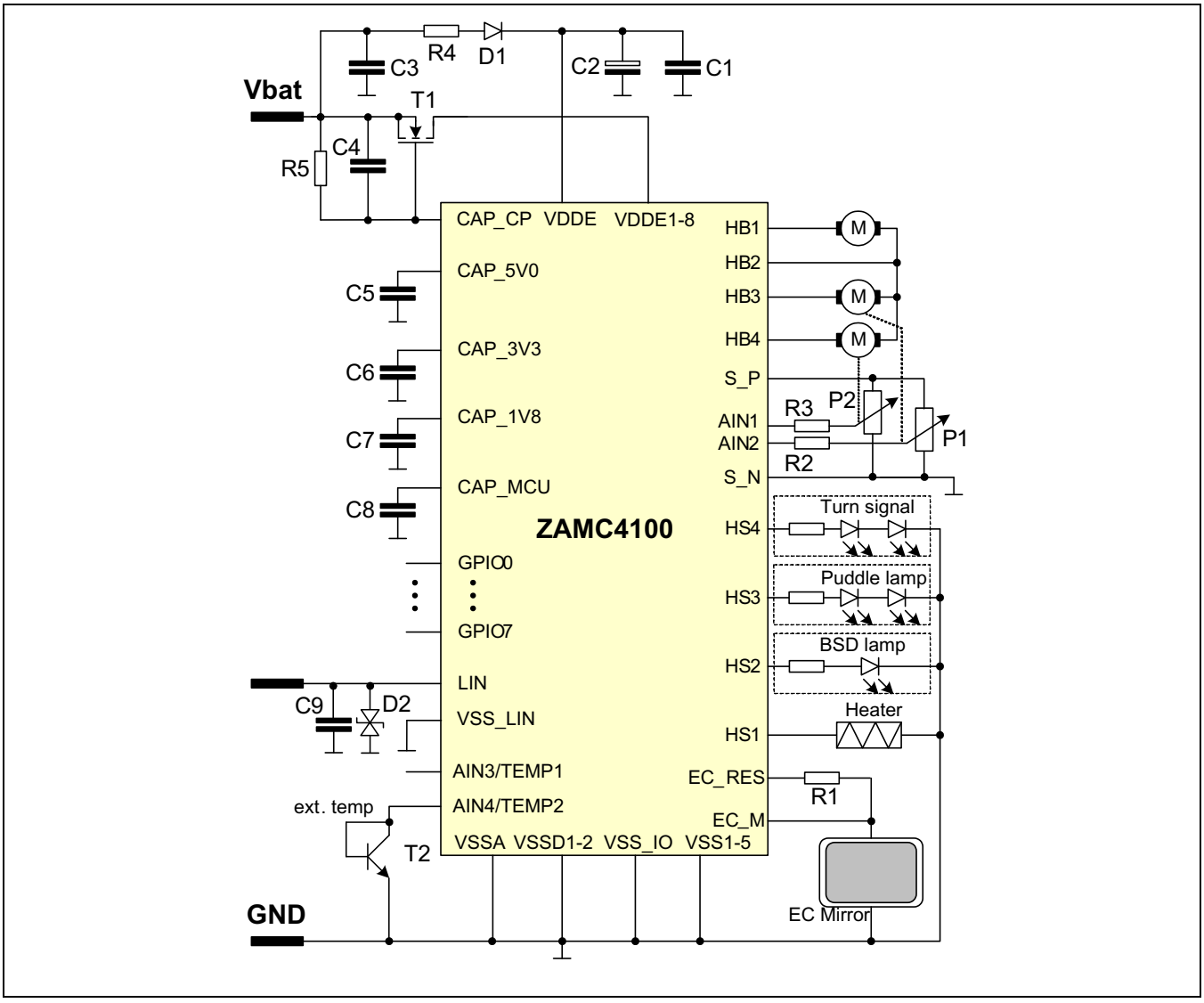
6 Application Circuit and External Components

6.1. Application Circuit Diagram for Mirror Controller

Figure 6.1 shows a typical application for ZAMC4100 as a controller for an automotive outside rear mirror.

See section 6.2 for the values and functions of the external components.

Figure 6.1 ZAMC4100 Application Circuit for Automotive Mirror Control



6.2. External Components for Automotive Mirror Control Application

Table 6.1 lists the values, purpose, and tolerances for the external components shown in Figure 6.1.

Table 6.1 Parameters for External Components in Typical Automotive Mirror Control Application

Component	Function	Min	Typ	Max	Tol	Units	Note
C1	Supply filtering	-	100	-	10%	nF	Capacitor (ceramic)
C2	Supply filtering	-	10	-	10%	µF	Capacitor (ceramic)
C3	Supply filtering	-	68	-	10%	nF	Capacitor (ceramic)
C4	Charge pump	-	68	-	10%	nF	Capacitor (ceramic)
C5	Regulator stability	-	220	-	10%	nF	Capacitor (ceramic)
C6	Regulator stability	-	47	-	10%	nF	Capacitor (ceramic)
C7	Regulator stability	-	47	-	10%	nF	Capacitor (ceramic)
C8	Regulator stability	-	47	-	10%	nF	Capacitor (ceramic)
C9 [1)]	LIN	-	100	-	1%	pF	Capacitor (ceramic)
T1	Reverse polarity	-	-	-	-	-	IRF7470
T2 [2)]	External temperature sensor	-	-	-	-	-	MMBT3904
R1	Load for EC mirror	-	47	-	5%	Ω	Resistor/~1W
R2/3	Short to Vbat prot.	-	1	-	10%	kΩ	Resistor
R4	Reverse polarity circuit	-	4.7	-	10%	Ω	Resistor
R5	Reverse polarity circuit	-	2.2	-	10%	MΩ	Resistor
P1/2	Position sensors	2		5	10%	kΩ	Potentiometer
D1	Reverse polarity	-	-	-	-	-	BAT54

1) According to the LIN 2.0 standard, the total capacitance of the SLAVE LIN pin must be less than 220pF.

2) Optional component for external temperature measurement.

© 2019 Renesas Electronics Corporation

7 Multi-chip Module Assembly, Pin Layout, and Pin Assignments

ZAMC4100 is a highly integrated solution consisting of Microcontroller Unit (MCU) and System Basis Chip (SBC) within a single plastic QFN package as shown below. Both chips are assembled in a multi-chip-package placed horizontally alongside one another. The exposed pad allows better thermal performance.

Figure 7.2 gives the pin layout and assignments. See Table 7.1 for the description of the pins.

Figure 7.1 ZAMC4100 Multi-chip-Module Assembly

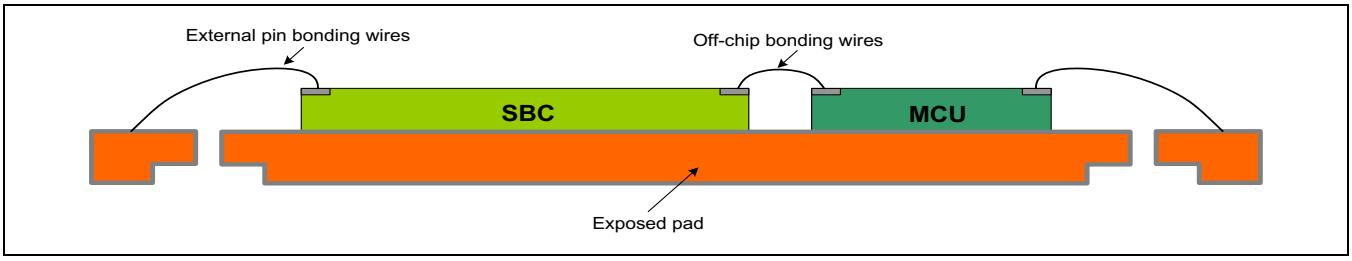


Figure 7.2 Pin Layout for ZAMC4100 PQFN65 – Top View

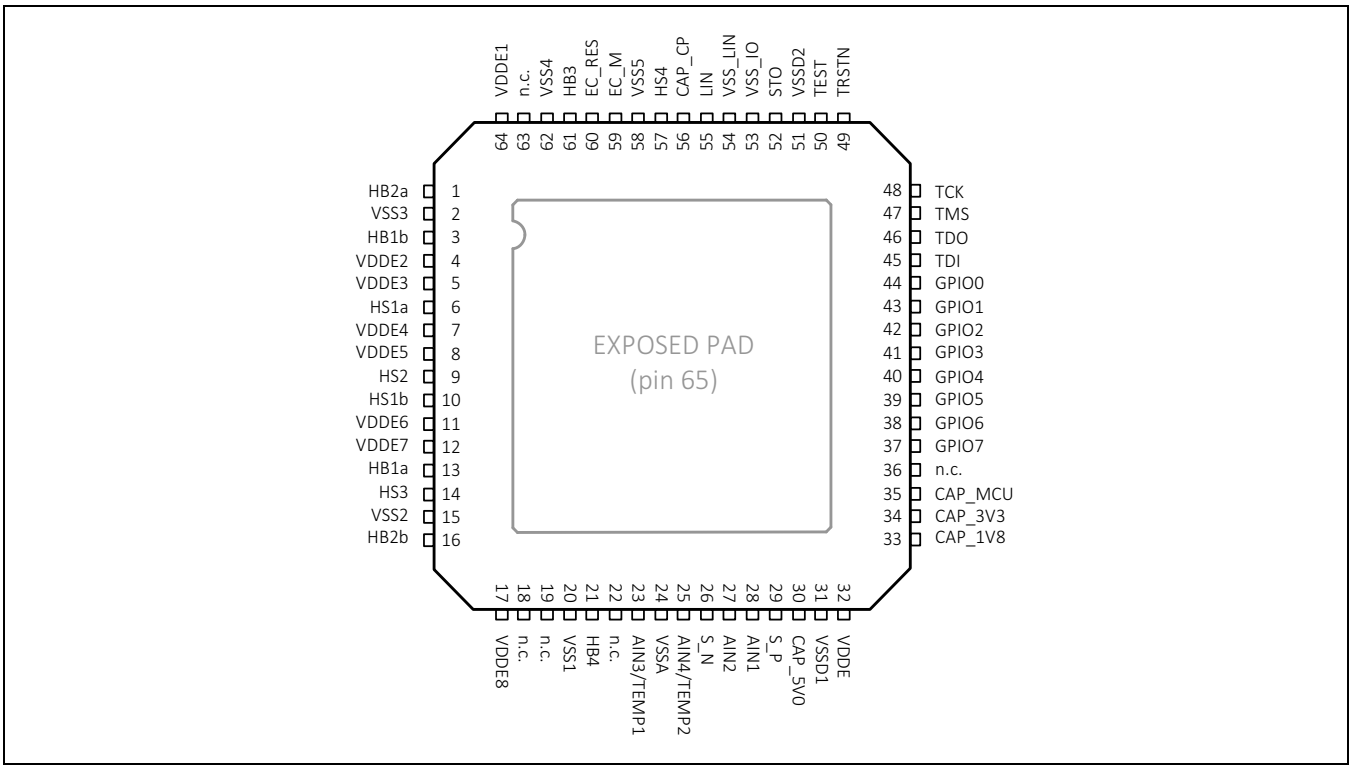


Table 7.1 Pin Description

Note: See important table notes at the end of the table.

Pin Name	Position	Purpose	Mode	Description
SUPPLY PINS				
VDDE	32	Supply	Input	Supply for the system
VDDE1	64	Supply	Input	High current supply for the drivers
VDDE2	4			
VDDE3	5			
VDDE4	7			
VDDE5	8			
VDDE6	11			
VDDE7	12			
VDDE8	17			
VSS1 (1)	20	Supply	Input	High current ground for drivers
VSS2 (1)	15			
VSS3 (1)	2			
VSS4 (1)	62			
VSS5 (1)	58			
VSS_LIN (1)	54	Supply	Input	Ground for LIN transceiver
VSSA (1)	24	Supply	Input	Ground for analog blocks (low noise)
VSSD1 (1)	31	Supply	Input	Ground for digital blocks of SBC
VSSD2 (1)	51	Supply	Input	Ground for MCU
VSS_IO (1)	53	Supply	Input	Ground for SBC digital IOs
EXPOSED PAD (2)	65	Heat sink	-	Connected to digital ground
CAP_5V0 (3)	30	Analog	In/Out	Connection to external capacitor for 5V (analog supply)
CAP_3V3 (3)	34	Analog	In/Out	Connection to external capacitor for 3.3V (MCU peripheral supply)
CAP_1V8 (3)	33	Analog	In/Out	Connection to external capacitor for 1.8V (SBC digital and MCU core supply)
CAP_MCU (3)	35	Analog	In/Out	Connection to external capacitor for MCU 1V8 core voltage supply
CAP_CP (3)	56	Analog	In/Out	Connection to external capacitor for the charge pump

© 2019 Renesas Electronics Corporation

Pin Name	Position	Purpose	Mode	Description
SENSORS INTERFACE				
S_P	29	Analog	Output	Positive supply for sensors (5V)
S_N [4]	26	Analog	Output	Negative supply for sensors (0V)
AIN1 [5]	28	Analog	Input	General purpose ADC input
AIN2 [5]	27	Analog	Input	General purpose ADC input
AIN3/TEMP1 [5]	23	Analog	In/Out	General purpose ADC input/temperature sensor input
AIN4/TEMP2 [5]	25	Analog	In/Out	General purpose ADC input/temperature sensor input
ACTUATORS INTERFACE				
HB1a	13	Analog	Output	Output of half-bridge driver 1
HB1b	3			
HB2a	1	Analog	Output	Output of half-bridge driver 2
HB2b	16			
HB3	61	Analog	Output	Output of half-bridge driver 3
HB4	21	Analog	Output	Output of half-bridge driver 4
HS1a	6	Analog	Output	Output of high-side switch 1
HS1b	10			
HS2	9	Analog	Output	Output of high-side switch 2
HS3	14	Analog	Output	Output of high-side switch 3
HS4	57	Analog	Output	Output of high-side switch 4
EC_M	59	Analog	Output	Output for EC mirror control
EC_RES	60	Analog	Input	Connection to external resistor for EC
COMMUNICATION INTERFACE				
LIN	55	Analog	In/Out	LIN bus
GPIO0 [5]	44	Digital	In/Out	General purpose I/O pin
GPIO1 [5]	43	Digital	In/Out	General purpose I/O pin
GPIO2 [5]	42	Digital	In/Out	General purpose I/O pin
GPIO3 [5]	41	Digital	In/Out	General purpose I/O pin
GPIO4 [5]	40	Digital	In/Out	General purpose I/O pin
GPIO5 [5]	39	Digital	In/Out	General purpose I/O pin
GPIO6 [5]	38	Digital	In/Out	General purpose I/O pin
GPIO7 [5]	37	Digital	In/Out	General purpose I/O pin

© 2019 Renesas Electronics Corporation

Pin Name	Position	Purpose	Mode	Description
TEST INTERFACE				
TEST	50	Digital	In	Global test enable pin; connect this pin to GND in application
TRSTN	49	Digital	In	JTAG low-active reset
TCK	48	Digital	In	JTAG clock
TMS	47	Digital	In	JTAG mode select
TDO	46	Digital	Out	JTAG data out pin
TDI	45	Digital	In	JTAG data in pin
STO	52	Digital	Out	SBC test data out; leave this pin open in the application
n.c.	18,19, 22,36,63	-	-	Not bonded package pins; connect them to ground

(1) The ground connections of the ZAMC4100 on the PCB should be separated into three specific planes for ground decoupling and better noise immunity:

- Analog ground: VSSA, S_N, CAP_5V0 (ground plate)
- Digital ground: VSSD1, VSSD2, VSS_IO, VSS_LIN, CAP_3V3 (ground plate), CAP_1V8 (ground plate), CAP_MCU (ground plate)
- Power ground: VSS1 to 5

All three ground planes must be connected to pin 65 (ZAMC4100 exposed pad), which must be a single common GND point on the PCB.

(2) In the application, the exposed pad (pin 65) must be connected to ground.

(3) Do not supply any external circuitry via these pins.

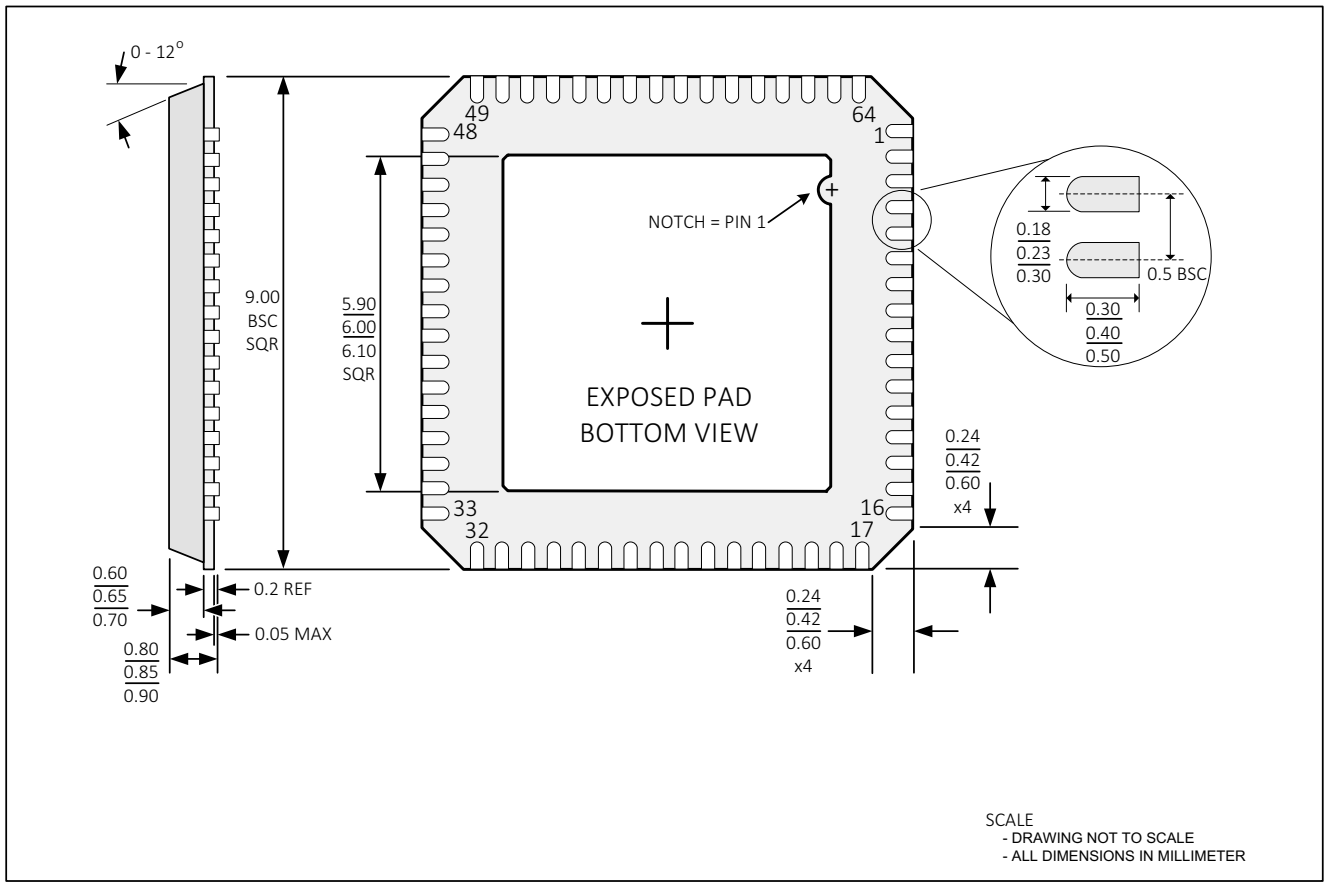
(4) If the S_N pin is not used for supplying sensors, it must be connected to the GND rail on the PCB.

(5) All unused ADC inputs (pins AIN1 to 4) and GPIO ports (pins GPIO0 to 7) can be left open or grounded for better EMI.

© 2019 Renesas Electronics Corporation

8 ZAMC4100 Outline Dimensions: Plastic QFN64 9x9mm (Package Type 1)

Figure 8.1 ZAMC4100 Package Dimensions – Plastic QFN64



9 Ordering Information

Product Sales Code	Description	Package
ZAMC4100GA2R	ZAMC4100GA PQFN64 – Temperature range: -40°C to +85°C	13” Tape and Reel
ZAMC4100GA2V	ZAMC4100GA PQFN64 – Temperature range: -40°C to +85°C	Tray
ZAMC4100 Evaluation Kit V2.0 including the ZAMC4100 Evaluation Board; software available at *www.IDT.com/ZAMC4100*		
ZAMC4100 Application Kit V1.0 including the ZAMC4100 Application Board		

10 Related Documents

Document
ZAMC4100 Feature Sheet
ZAMC4100 Evaluation Kit Description
ZAMC4100 Application Kit Description

Visit the ZAMC4100 product page at www.IDT.com/zamc4100 or contact your nearest sales office for the latest version of these documents.

11 Glossary

Abbreviation	Description
ADC	Analog-to-Digital Converter
ARM®	Advanced RISC Machine (RISC = Reduced Instruction Set Computing)
Cortex™-M0	32-bit RISC processor core licensed by ARM
CRC	Cyclic Redundancy Check
CS	Current Source
CSH	Current Source High-Side
CSL	Current Source Low-Side
CSN	Chip Select (Negative)
DAC	Digital to Analog Converter
DDM	Driver Door Module
ECC	Error Correction Controller
ECM	Electrochromatic Mirror
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
FLASH	Non-volatile memory that can be electrically erased and programmed. Usually where the program is stored.

© 2019 Renesas Electronics Corporation

Abbreviation	Description
FSR	Full-Scale Range
GND	Ground potential of the application
GPIO	General Purpose Input Output interface
IC	Integrated Circuit
IRQN	Interrupt Request (Negative)
JTAG	Based on IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture
LIN	Local Interconnect Network
LIN PHY	LIN Physical layer
MCU	Microcontroller Unit
MOSI	Master Out Slave In
MISO	Master In Slave Out
MUX	Multiplexer
NMI	Non-maskable Interrupt
NMOS	N-type MOSFET
NVIC	Nested Vector Interrupt Controller
OC	Over-Current
OCH	Over-Current for High-Side Driver
OCL	Over-Current for Low-Side Driver
OSCH	High frequency oscillator
OSCL	Low frequency oscillator
OSI	Open Systems Interconnection
OV	Over-Voltage
OVL	Over-Load
OVT	Over-Temperature
PCB	Printed Circuit Board
POR	Power-On-Reset
PWM	Pulse-Width Modulation
RESCODE	Result Code from ADC
SAR	Successive Approximation
SBC	System Basis Chip
SPI	Serial Peripheral Interface
SIP	System-in-Package
SRAM	Static RAM (RAM = Random Access Memory)

© 2019 Renesas Electronics Corporation

Abbreviation	Description
SW-LIN	Software layer of the LIN protocol
STBY	Standby
UART	Universal Asynchronous Receiver Transmitter
UV	Under Voltage
VBAT	Battery supply
WDT	Watchdog Timer

12 Document Revision History

Revision	Date	Description
1.00	May 11, 2015	First release
	January 25, 2016	Changed to IDT branding.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2026 Renesas Electronics Corporation. All rights reserved.