Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	73
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s912zvhl64f1vll

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Routing Register	Pin Function after Reset
U	PU[7]	M1SINP	I/O	SSD1 Sine+ Node		GPIO
		M1C1P	O	Motor control output for motor 1		
		PTU[7]	I/O	General purpose		
	PU[6]	M1SINM	I/O	SSD1 Sine- Node		
		M1C1M	O	Motor control output for motor 1		
		IOC0_3	I/O	TIM0 channel 3		
		PTU[6]	I/O	General purpose		
	PU[5]	M1COSP	I/O	SSD1 Cosine+ Node		
		M1C0P	O	Motor control output for motor 1		
		PTU[5]	I/O	General purpose		
	PU[4]	M1COSM	I/O	SSD1 Cosine- Node		
		M1C0M	O	Motor control output for motor 1		
		IOC0_2	I/O	TIM0 channel2		
		PTU[4]	I/O	General purpose		
	PU[3]	M0SINP	I/O	SSD0 Sine+ Node		
		M0C1P	O	Motor control output for motor 0		
		PTU[3]	I/O	General purpose		
	PU[2]	M0SINM	I/O	SSD0 Sine- Node		
		M0C1M	O	Motor control output for motor 0		
		IOC0_1	I/O	TIM0 channel 1		
		PTU[2]	I/O	General purpose		
	PU[1]	M0COSP	I/O	SSD0 Cosine+ Node		
		M0C0P	O	Motor control output for motor 0		
		PTU[1]	I/O	General purpose		
	PU[0]	M0COSM	I/O	SSD0 Cosine- Node		
		M0C0M	O	Motor control output for motor 0		
		IOC0_0	I/O	TIM0 channel 0		
		PTU[0]	I/O	General purpose		

1. Signals in parenthesis denote alternative module routing pins
2. Function active when $\overline{\text{RESET}}$ asserted.
3. The interrupt is enabled by clearing the X mask bit in the CPU CCR. The pin is forced to input upon first clearing of the X bit and is held in this state until reset. A stop or wait recovery with the X bit set (refer to S12ZCPU reference manual) is not available.

2.3 Memory Map and Register Definition

This section provides a detailed description of all port integration module registers.

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0280	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0281	PTADL	R	PTADL7	PTADL6	PTADL5	PTADL4	PTADL3	PTADL2	PTADL1	PTADL0
		W								
0x0282	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0283	PTIADL	R	PTIADL7	PTIADL6	PTIADL5	PTIADL4	PTIADL3	PTIADL2	PTIADL1	PTIADL0
		W								
0x0284	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0285	DDRADL	R	DDRADL7	DDRADL6	DDRADL5	DDRADL4	DDRADL3	DDRADL2	DDRADL1	DDRADL0
		W								
0x0286	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0287	PERADL	R	PERADL7	PERADL6	PERADL5	PERADL4	PERADL3	PERADL2	PERADL1	PERADL0
		W								
0x0288	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0289	PPSADL	R	PPSADL7	PPSADL6	PPSADL5	PPSADL4	PPSADL3	PPSADL2	PPSADL1	PPSADL0
		W								
0x028A–0x028B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x028C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x028D	PIEADL	R	PIEADL7	PIEADL6	PIEADL5	PIEADL4	PIEADL3	PIEADL2	PIEADL1	PIEADL0
		W								
0x028E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x028F	PIFADL	R	PIFADL7	PIFADL6	PIFADL5	PIFADL4	PIFADL3	PIFADL2	PIFADL1	PIFADL0
		W								
0x0290–0x0298	Reserved	R	0	0	0	0	0	0	0	0
		W								

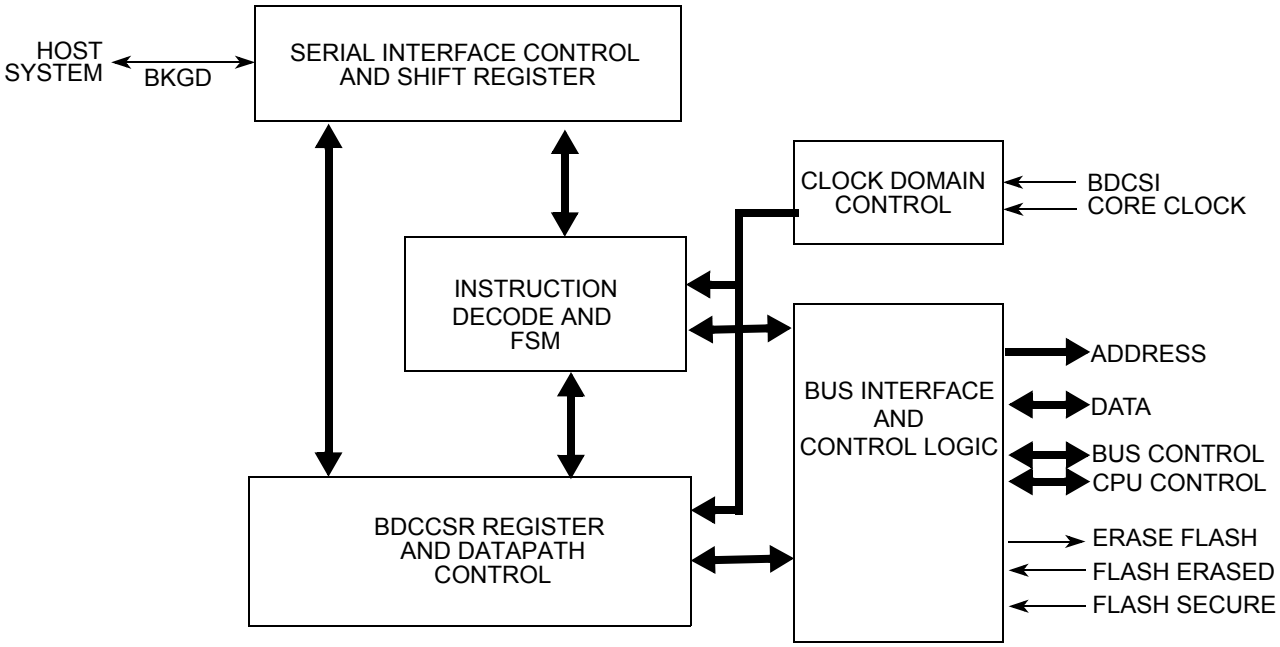


Figure 5-1. BDC Block Diagram

5.2 External Signal Description

A single-wire interface pin (BKGD) is used to communicate with the BDC system. During reset, this pin is a device mode select input. After reset, this pin becomes the dedicated serial interface pin for the BDC.

BKGD is a pseudo-open-drain pin with an on-chip pull-up. Unlike typical open-drain pins, the external RC time constant on this pin due to external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speed-up pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to Section 5.4.6" for more details.

5.3 Memory Map and Register Definition

5.3.1 Module Memory Map

Table 5-4 shows the BDC memory map.

Table 5-4. BDC Memory Map

Global Address	Module	Size (Bytes)
Not Applicable	BDC registers	2

Table 6-24. SSF[2:0] — State Sequence Flag Bit Encoding

SSF[2:0]	Current State
101,110,111	Reserved

6.3.2.12 Debug Comparator A Control Register (DBGACTL)

Address: 0x0110

	7	6	5	4	3	2	1	0
R	0	NDB	INST	0	RW	RWE	reserved	COMPE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 6-14. Debug Comparator A Control Register

Read: Anytime.

Write: If DBG not armed and PTACT is clear.

Table 6-25. DBGACTL Field Descriptions

Field	Description
6 NDB	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. This bit is ignored if the INST bit in the same register is set. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
5 INST	**Instruction Select** — This bit configures the comparator to compare PC or data access addresses. 0 Comparator compares addresses of data accesses 1 Comparator compares PC address
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is ignored if RWE is clear or INST is set. 0 Write cycle is matched 1 Read cycle is matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is ignored when INST is set. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
0 COMPE	**Enable Bit** — Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled

Table 6-26 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if INST is set, because matches based on opcodes reaching the execution stage are data independent.

Table 6-47. Normal and Loop1 Mode Trace Buffer Format without Timestamp

CPU	CINF1	CPCH1	CPCM1	CPCL1	CINF0	CPCH0	CPCM0	CPCL0
	CINF3	CPCH3	CPCM3	CPCL3	CINF2	CPCH2	CPCM2	CPCL2

Table 6-48. Normal and Loop1 Mode Trace Buffer Format with Timestamp

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
CPU	Timestamp	Timestamp	Reserved	Reserved	CINF0	CPCH0	CPCM0	CPCL0
	Timestamp	Timestamp	Reserved	Reserved	CINF1	CPCH1	CPCM1	CPCL1

CINF contains information relating to the CPU.

CPU Information Byte CINF For Normal And Loop1 Modes

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CET		0	0	CTI	EEVI	0	TOVF

Figure 6-27. CPU Information Byte CINF

Table 6-49. CINF Bit Descriptions

Field	Description
7–6 CET	**CPU Entry Type Field** — Indicates the type of stored address of the trace buffer entry as described in Table 6-50
3 CTI	**Comparator Timestamp Indicator** — This bit indicates if the trace buffer entry corresponds to a comparator timestamp. 0 Trace buffer entry initiated by trace mode specification conditions or timestamp counter overflow 1 Trace buffer entry initiated by comparator D match
2 EEVI	**External Event Indicator** — This bit indicates if the trace buffer entry corresponds to an external event. 0 Trace buffer entry not initiated by an external event 1 Trace buffer entry initiated by an external event
0 TOVF	**Timestamp Overflow Indicator** — Indicates if the trace buffer entry corresponds to a timestamp overflow 0 Trace buffer entry not initiated by a timestamp overflow 1 Trace buffer entry initiated by a timestamp overflow

Table 6-50. CET Encoding

CET	Entry Type Description
00	Non COF opcode address (entry forced by an external event)
01	Vector destination address
10	Source address of COF opcode
11	Destination address of COF opcode

7.3 Memory Map and Registers

This section provides a detailed description of all registers accessible in the S12CPMU_UHV_V5.

7.3.1 Module Memory Map

The S12CPMU_UHV_V5 registers are shown in Figure 7-3.

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	CPMU RESERVED00	R	0	0	0	0	0	0	0	0
		W								
0x0001	CPMU RESERVED01	R	0	0	0	0	0	0	0	0
		W								
0x0002	CPMU RESERVED02	R	0	0	0	0	0	0	0	0
		W								
0x0003	CPMURFLG	R	0	PORF	LVRF	0	COPRF	0	OMRF	PMRF
		W								
0x0004	CPMU SYNR	R	VCOFRQ[1:0]		SYNDIV[5:0]					
		W								
0x0005	CPMU REFDIV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
		W								
0x0006	CPMU POSTDIV	R	0	0	0	POSTDIV[4:0]				
		W								
0x0007	CPMUIFLG	R	RTIF	0	0	LOCKIF	LOCK	0	OSCIF	UPOSC
		W								
0x0008	CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
		W								
0x0009	CPMUCLKS	R	PLLSEL	PSTP	CSAD	COP OSCSEL1	PRE	PCE	RTI OSCSEL	COP OSCSEL0
		W								
0x000A	CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
		W								
0x000B	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								
0x000C	CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		W			WRTMASK					
0x000D	RESERVED CPMUTEST0	R	0	0	0	0	0	0	0	0
		W								
0x000E	RESERVED CPMUTEST1	R	0	0	0	0	0	0	0	0
		W								
			= Unimplemented or Reserved							

Figure 7-3. CPMU Register Summary

10.4.2.24 ADC Command and Result Offset Register 1 (ADCCROFF1)

It is important to note that these bits do not represent absolute addresses instead it is an sample offset (object size 16bit for RVL, object size 32bit for CSL).

Module Base + 0x0025

	7	6	5	4	3	2	1	0
R	0	CMDRES_OFF1[6:0]						
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-27. ADC Command and Result Offset Register 1 (ADCCROFF1)

Read: Anytime

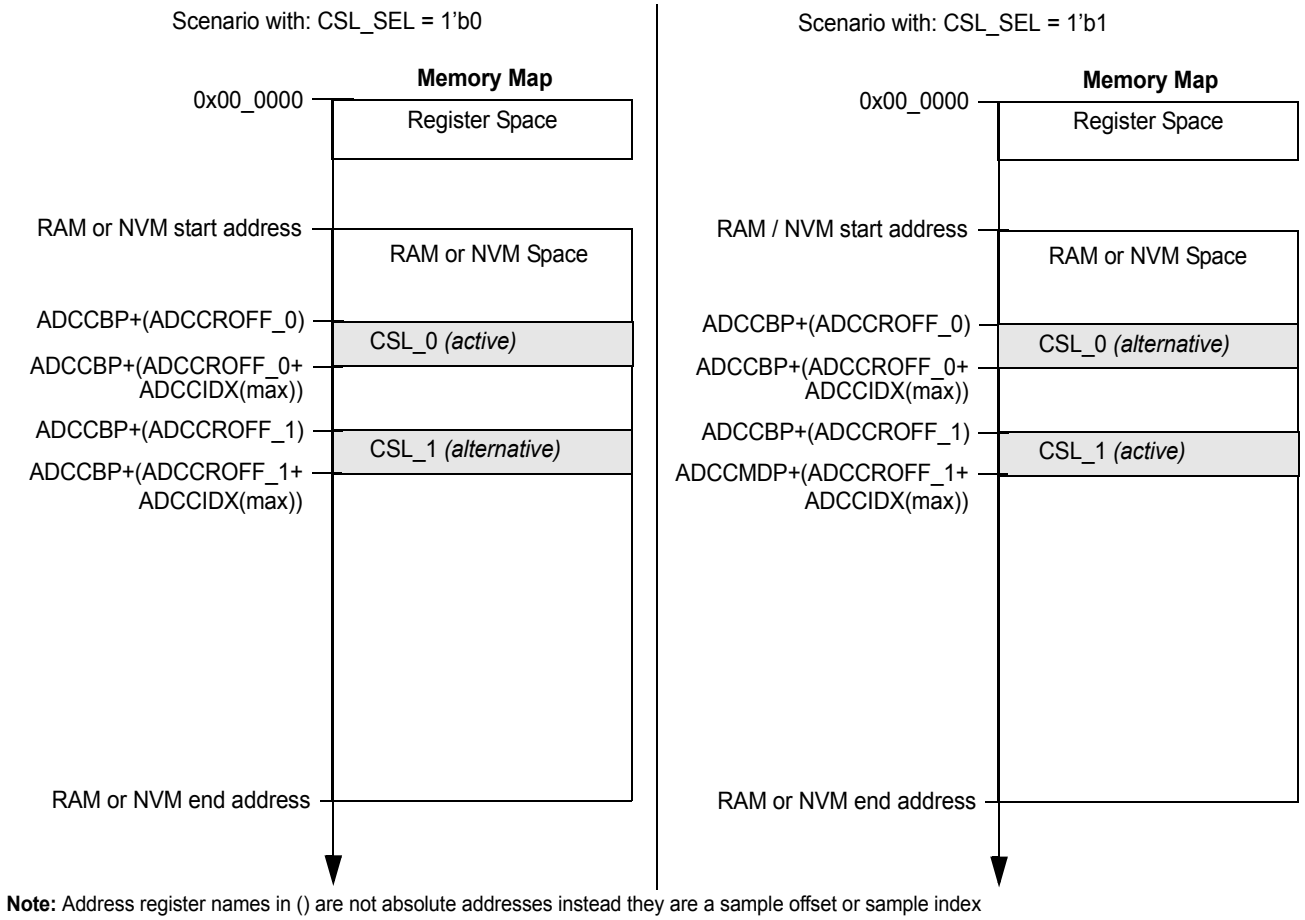
Write: These bits are writable if bit ADC_EN clear or bit SMOD_ACC set

Table 10-31. ADCCROFF1 Field Descriptions

Field	Description
6-0 CMDRES_OFF1 [6:0]	**ADC Result Address Offset Value** — These bits represent the conversion command and result offset value relative to the conversion command base pointer address and result base pointer address in the memory map to refer to CSL_1 and RVL_1. It is used to calculate the address inside the system RAM to which the result at the end of the current conversion is stored to and the area (RAM or NVM) from which the conversion commands are loaded from. These bits do not represent absolute addresses instead it is an sample offset (object size 16bit for RVL, object size 32bit for CSL).,These bits can only be modified if bit ADC_EN is clear. See also Section 10.5.3.2.2, “Introduction of the two Command Sequence Lists (CSLs) and Section 10.5.3.2.3, “Introduction of the two Result Value Lists (RVLs) for more details.

10.5.3.2.2 Introduction of the two Command Sequence Lists (CSLs)

The two Command Sequence Lists (CSLs) can be referred to via the Command Base Pointer Register plus the Command and Result Offset Registers plus the Command Index Register (ADCCBP, ADCCROFF_0/1, ADCCIDX).
The final address for conversion command loading is calculated by the sum of these registers (e.g.: ADCCBP+ADCCROFF_0+ADCCIDX or ADCCBP+ADCCROFF_1+ADCCIDX).
Bit CSL_BMOD selects if the CSL is used in double buffer or single buffer mode. In double buffer mode, the CSL can be swapped by flow control bits LDOK and RSTA. For detailed information about when and how the CSL is swapped, please refer to Section 10.5.3.2.5, “The four ADC conversion flow control bits - description of Restart Event + CSL Swap, Section 10.8.7.1, “Initial Start of a Command Sequence List and Section 10.8.7.3, “Restart CSL execution with new/other CSL (alternative CSL becomes active CSL) — CSL swapping
Which list is actively used for ADC command loading is indicated by bit CSL_SEL. The register to define the CSL start addresses (ADCCBP) can be set to any even location of the system RAM or NVM area. It is the user’s responsibility to make sure that the different ADC lists do not overlap or exceed the system RAM or the NVM area, respectively. The error flag IA_EIF will be set for accesses to ranges outside system RAM area and cause an error interrupt if enabled.



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 10-31. Command Sequence List Schema in Double Buffer Mode

12.3.2.5 SCI Alternative Control Register 2 (SCIACR2)

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	IREN	TNP1	TNP0	0	0	BERRM1	BERRM0	BKDFE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-8. SCI Alternative Control Register 2 (SCIACR2)

Read: Anytime, if AMAP = 1

Write: Anytime, if AMAP = 1

Table 12-7. SCIACR2 Field Descriptions

Field	Description
7 IREN	**Infrared Enable Bit** — This bit enables/disables the infrared modulation/demodulation submodule. 0 IR disabled 1 IR enabled
6:5 TNP[1:0]	**Transmitter Narrow Pulse Bits** — These bits enable whether the SCI transmits a 1/16, 3/16, 1/32 or 1/4 narrow pulse. See Table 12-8.
2:1 BERRM[1:0]	**Bit Error Mode** — Those two bits determines the functionality of the bit error detect feature. See Table 12-9.
0 BKDFE	**Break Detect Feature Enable** — BKDFE enables the break detect circuitry. 0 Break detect circuit disabled 1 Break detect circuit enabled

Table 12-8. IRSCI Transmit Pulse Width

TNP[1:0]	Narrow Pulse Width
11	1/4
10	1/32
01	1/16
00	3/16

Table 12-9. Bit Error Mode Coding

BERRM1	BERRM0	Function
0	0	Bit error detect circuit is disabled
0	1	Receive input sampling occurs during the 9th time tick of a transmitted bit (refer to Figure 12-19)
1	0	Receive input sampling occurs during the 13th time tick of a transmitted bit (refer to Figure 12-19)
1	1	Reserved

In Figure 12-22 the verification samples RT3 and RT5 determine that the first low detected was noise and not the beginning of a start bit. The RT clock is reset and the start bit search begins again. The noise flag is not set because the noise occurred before the start bit was found.

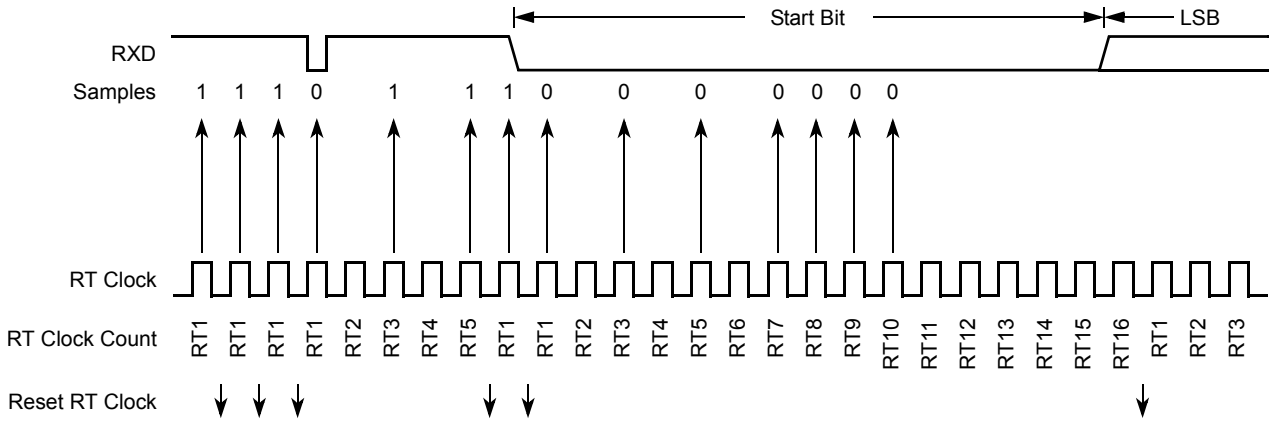


Figure 12-22. Start Bit Search Example 1

In Figure 12-23, verification sample at RT3 is high. The RT3 sample sets the noise flag. Although the perceived bit time is misaligned, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

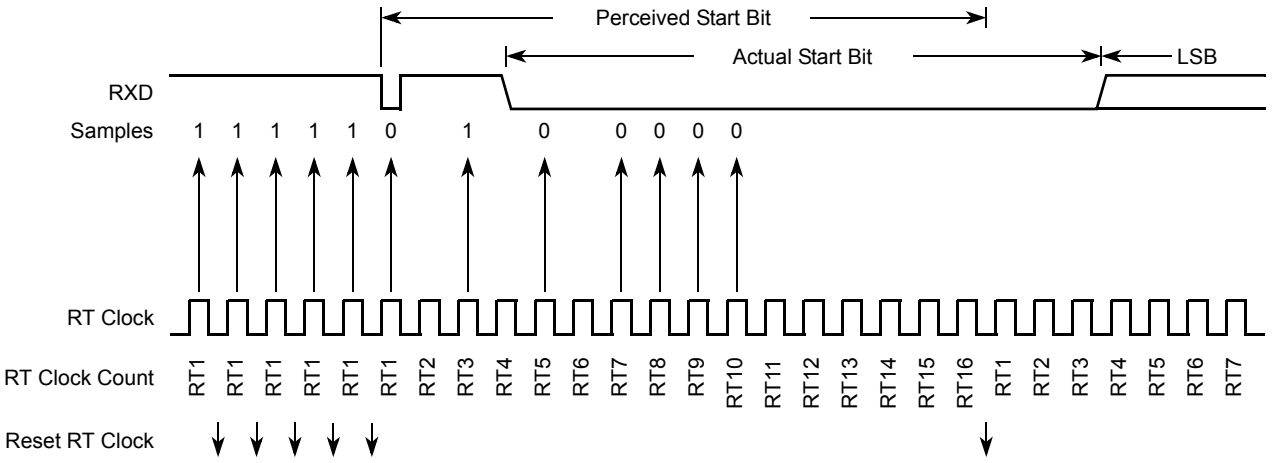


Figure 12-23. Start Bit Search Example 2

12.4.6.5 Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit. A noise error will occur if the RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic zero.

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Re synchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

12.4.6.5.1 Slow Data Tolerance

Figure 12-28 shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

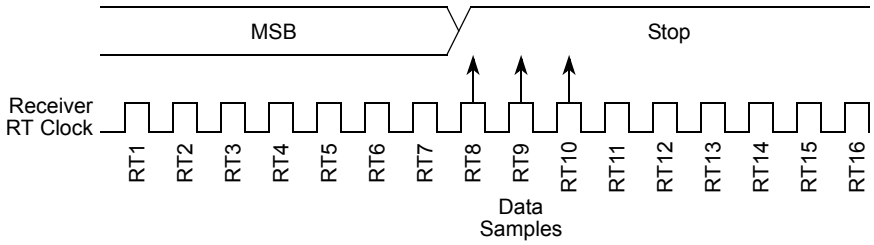


Figure 12-28. Slow Data

Let’s take RTr as receiver RT clock and RTt as transmitter RT clock.

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles +7 RTr cycles = 151 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 12-28, the receiver counts 151 RTr cycles at the point when the count of the transmitting device is 9 bit times x 16 RTt cycles = 144 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

((151 – 144) / 151) x 100 = 4.63%

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 7 RTr cycles = 167 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 12-28, the receiver counts 167 RTr cycles at the point when the count of the transmitting device is 10 bit times x 16 RTt cycles = 160 RTt cycles.

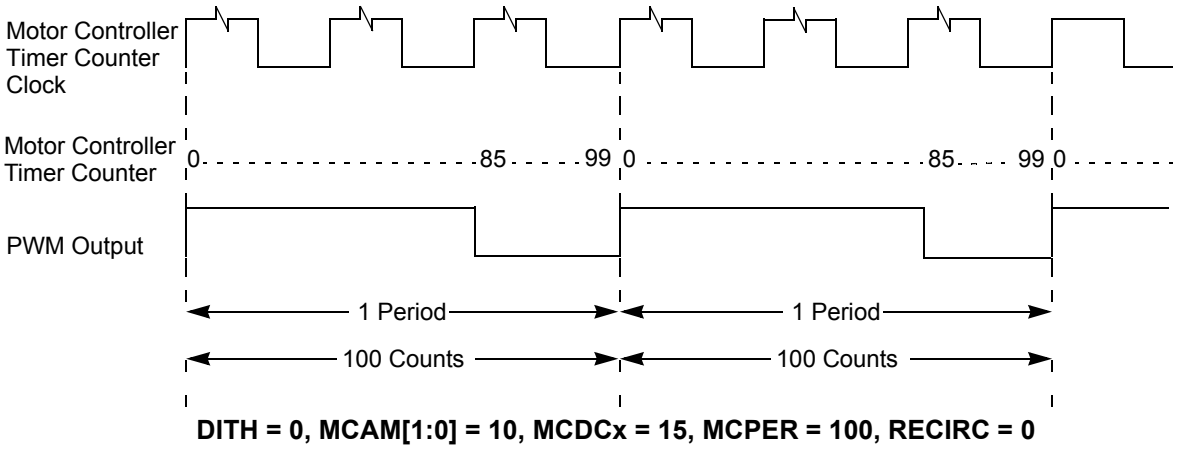
The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

((167 – 160) / 167) X 100 = 4.19%

Table 14-8. IBCR Field Descriptions

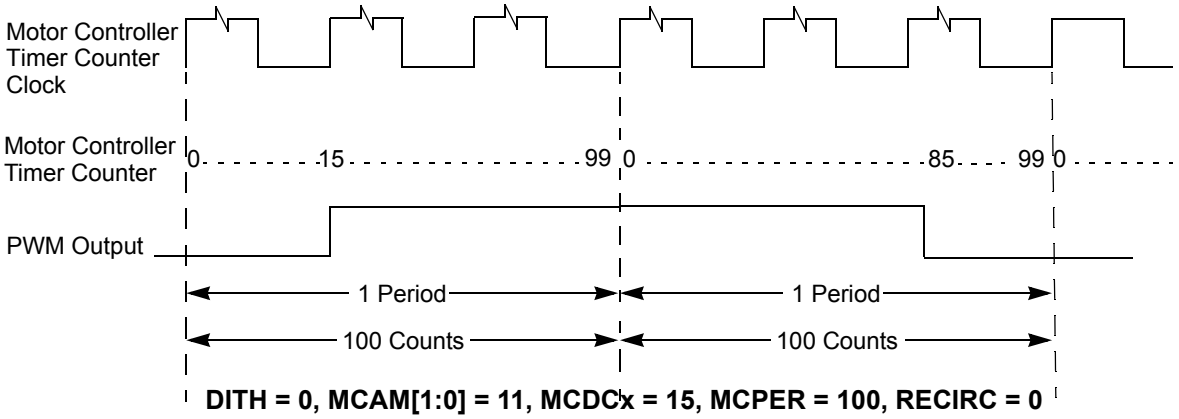
Field	Description
7 IBEN	**I-Bus Enable** — This bit controls the software reset of the entire IIC bus module. 0 The module is reset and disabled. This is the power-on reset situation. When low the interface is held in reset but registers can be accessed 1 The IIC bus module is enabled.This bit must be set before any other IBCR bits have any effect If the IIC bus module is enabled in the middle of a byte transfer the interface behaves as follows: slave mode ignores the current transfer on the bus and starts operating whenever a subsequent start condition is detected. Master mode will not be aware that the bus is busy, hence if a start cycle is initiated then the current bus cycle may become corrupt. This would ultimately result in either the current bus master or the IIC bus module losing arbitration, after which bus operation would return to normal.
6 IBIE	**I-Bus Interrupt Enable** 0 Interrupts from the IIC bus module are disabled. Note that this does not clear any currently pending interrupt condition 1 Interrupts from the IIC bus module are enabled. An IIC bus interrupt occurs provided the IBIF bit in the status register is also set.
5 MS/SL	**Master/Slave Mode Select Bit** — Upon reset, this bit is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus, and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave.A STOP signal should only be generated if the IBIF flag is set. MS/SL is cleared without generating a STOP signal when the master loses arbitration. 0 Slave Mode 1 Master Mode
4 Tx/Rx	**Transmit/Receive Mode Select Bit** — This bit selects the direction of master and slave transfers. When addressed as a slave this bit should be set by software according to the SRW bit in the status register. In master mode this bit should be set according to the type of transfer required. Therefore, for address cycles, this bit will always be high. 0 Receive 1 Transmit
3 TXAK	**Transmit Acknowledge Enable** — This bit specifies the value driven onto SDA during data acknowledge cycles for both master and slave receivers. The IIC module will always acknowledge address matches, provided it is enabled, regardless of the value of TXAK. Note that values written to this bit are only used when the IIC bus is a receiver, not a transmitter. 0 An acknowledge signal will be sent out to the bus at the 9th clock bit after receiving one byte data 1 No acknowledge signal response is sent (i.e., acknowledge bit = 1)
2 RSTA	**Repeat Start** — Writing a 1 to this bit will generate a repeated START condition on the bus, provided it is the current bus master. This bit will always be read as a low. Attempting a repeated start at the wrong time, if the bus is owned by another master, will result in loss of arbitration. 1 Generate repeat start cycle
1 RESERVED	**Reserved** — Bit 1 of the IBCR is reserved for future compatibility. This bit will always read 0.
0 IBSWAI	**I Bus Interface Stop in Wait Mode** 0 IIC bus module clock operates normally 1 Halt IIC bus module clock generation in wait mode

Wait mode is entered via execution of a CPU WAI instruction. In the event that the IBSWAI bit is set, all clocks internal to the IIC will be stopped and any transmission currently in progress will halt.If the CPU were woken up by a source other than the IIC module, then clocks would restart and the IIC would resume



DITH = 0, MCAM[1:0] = 10, MCDCx = 15, MCPER = 100, RECIRC = 0

Center aligned (MCAM[1:0] = 11): Even periods will be output left aligned, odd periods will be output right aligned. PWM operation starts with the even period after the channel has been enabled. PWM operation in center aligned mode might start with the odd period if the channel has not been disabled before changing the alignment mode to center aligned.



DITH = 0, MCAM[1:0] = 11, MCDCx = 15, MCPER = 100, RECIRC = 0

16.4.1.3.2 Sign Bit (S)

Assuming RECIRC = 0 (the active state of the PWM signal is low), when the S bit for the corresponding channel is cleared, MnC0P (if the PWM channel number is even, n = 0, 1, 2, 3, see Table 16-11) or MnC1P (if the PWM channel number is odd, n = 0, 1, 2, 3, see Table 16-11), outputs a logic high while in (dual) full H-bridge mode. In half H-bridge mode the state of the S bit has no effect. The PWM output signal is generated on MnC0M (if the PWM channel number is even, n = 0, 1, 2, 3, see Table 16-11) or MnC1M (if the PWM channel number is odd, n = 0, 1, 2, 3).

Assuming RECIRC = 0 (the active state of the PWM signal is low), when the S bit for the corresponding channel is set, MnC0M (if the PWM channel number is even, n = 0, 1, 2, 3, see Table 16-11) or MnC1M (if the PWM channel number is odd, n = 0, 1, 2, 3, see Table 16-11), outputs a logic high while in (dual) full H-bridge mode. In half H-bridge mode the state of the S bit has no effect. The PWM output signal is generated on MnC0P (if the PWM channel number is even, n = 0, 1, 2, 3, see Table 16-11) or MnC1P (if the PWM channel number is odd, n = 0, 1, 2, 3).

16.7 Initialization/Application Information

This section provides an example of how the PWM motor controller can be initialized and used by application software. The configuration parameters (e.g., timer settings, duty cycle values, etc.) are not guaranteed to be adequate for any real application.

The example software is implemented in assembly language.

16.7.1 Code Example

One way to use the motor controller is:

1. Perform global initialization
   a) Set the motor controller control registers MCCTL0 and MCCTL1 to appropriate values.
      i) Prescaler disabled (MCPRE1 = 0, MCPRE0 = 0).
      ii) Fast mode and dither disabled (FAST = 0, DITH = 0).
      iii) Recirculation feature in dual full H-bridge mode disabled (RECIRC = 0).
      All other bits in MCCTL0 and MCCTL1 are set to 0.
   b) Configure the channel control registers for the desired mode.
      i) Dual full H-bridge mode (MCOM[1:0] = 11).
      ii) Left aligned PWM (MCAM[1:0] = 01).
      iii) No channel delay (MCCD[1:0] = 00).
2. Perform the startup phase
   a) Clear the duty cycle registers MCDC0 and MCDC1
   b) Initialize the period register MCPER, which is equivalent to enabling the motor controller.
   c) Enable the timer which generates the timebase for the updates of the duty cycle registers.
3. Main program
   a) Check if pin PB0 is set to “1” and execute the sub program if a timer interrupt is pending.
   b) Initiate the shutdown procedure if pin PB0 is set to “0”.
4. Sub program
   a) Update the duty cycle registers
      Load the duty cycle registers MCDC0 and MCDC1 with new values from the table and clear the timer interrupt flag.
      The sub program will initiate the shutdown procedure if pin PB0 is set to “0”.
   b) Shutdown procedure

The timer is disabled and the duty cycle registers are cleared to drive an inactive value on the PWM output as long as the motor controller is enabled. The period register is cleared after a certain time, which disables the motor controller. The table address is restored and the timer interrupt flag is cleared.

Chapter 19
Simple Sound Generator (SSGV1)

Table 19-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V1.0	Apr. 20, 2011	All	Initial revision of Sound Generator Module.

19.1 Introduction

This document describes the Simple Sound Generator module.

The SSG module generates audio frequency tone with autonomous amplitude control.

Refer to Figure 19-1 for the detailed block diagram of the module.

19.1.1 Features

The SSG block includes these distinctive features:

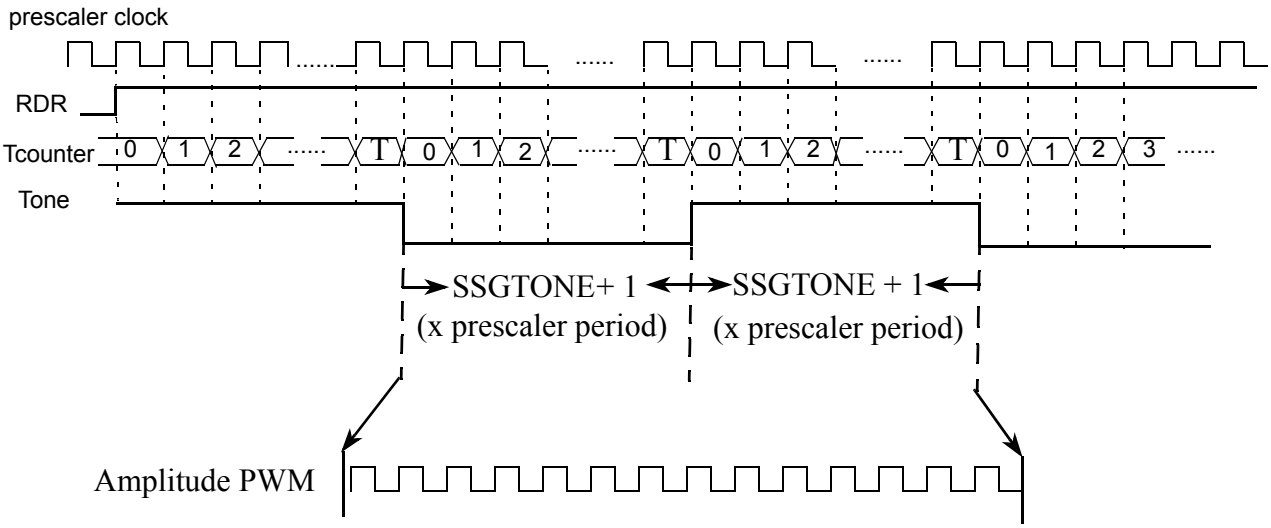
- Programmable amplitude level encoded with 11 bit resolution from zero amplitude to max amplitude
- Sound STOP function to stop sound generation immediately
- Registers double-buffered synchronously reload at edge of tone to avoid distortion of output tone
- Interrupt generates when SSG is ready to configure new sound data
- Input clock prescaler with 11 bit resolution
- Module disable for power saving when not in use
- Separate or mixed frequency and amplitude outputs for flexibility in external hardware variations
- Attack/decay function which can increase/decrease sound amplitude automatically without CPU interaction, including linear, gong and exponential attack/decay operation

19.1.2 Block Diagram

Figure 19-1 shows the block diagram for SSG block.

19.4.2 SSG Tone Generation

The tone is a low frequency square waveform. Tone generation logic consists of a tone counter triggered by prescaler signal and the tone period register SSGTONE with its buffer register. The register SSGTONE contains the coefficient value of half tone period, one tone cycle period = 2 x (SSGTONE + 1) x prescaler cycle. The tone signal is also used as one of the synchronous reload signal.



where : RDR is the RDR bit of SSGCR

Tcounter is the tone counter

T is value of SSGTONE buffer

Figure 19-22. SSG Tone Generation

19.4.3 SSG Attack and Decay function

In SSG the value hold in the amplitude buffer register can be automatically increased or decreased. The amplitude increase implements sound volume attack and the amplitude decrease implements sound volume decay. If ADS is cleared while the ADE is set, attack function will be enabled, if the ADS is set while ADE is set, decay function will be enabled. If ADE is cleared, attack/decay function will be disabled. The attack/decay function implements sound volume automatic increase/decrease with a small number of CPU interventions.

Attack/decay function includes linear operation, gong operation and exponential operation.

to re-enable the Verify Backdoor Access Key command. The security as defined in the Flash security byte (0xFF_FE0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0xFF_FE00-0xFF_FE07 are unaffected by the Verify Backdoor Access Key command sequence. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0xFF_FE00-0xFF_FE07 in the Flash configuration field.

21.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

A secured MCU can be unsecured in special single chip mode using an automated procedure described in Section 21.4.7.7.1, “Erase All Pin”, For a complete description about how to activate that procedure please look into the Reference Manual.

21.5.3 .Mode and Security Effects on Flash Command Availability

The availability of Flash module commands depends on the MCU operating mode and security state as shown in Table 21-29.

21.6 Initialization

On each system reset the flash module executes an initialization sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and DFPROT protection registers, and the FOPT and FSEC registers. The initialization routine reverts to built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register will be set.

CCIF is cleared throughout the initialization sequence. The Flash module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed once the hold is removed. Completion of the initialization sequence is marked by setting CCIF high which enables user commands.

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

23.3.2.3 Reserved Register

Module Base + Address 0x0002 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	x	x	x	x	x	x	x	x

= Unimplemented

Figure 23-5. LIN Test register

1. Read: Anytime
   Write: Only in special mode

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in special mode can alter the module's functionality.

Table 23-4. Reserved Register Field Description

Field	Description
7-0 Reserved	These reserved bits are used for test purposes. Writing to these bits can alter the module functionality.

23.3.2.4 LIN Slew Rate Mode Register (LPSLRM)

Module Base + Address 0x0003 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	LPDTDIS	0	0	0	0	0	LPSLR1	LPSLR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 23-6. LIN Slew Rate Mode Register (LPSLRM)

1. Read: Anytime
   Write: Only in shutdown mode (LPE=0)

Appendix A
MCU Electrical Specifications

A.1 General

This supplement contains the most accurate electrical information for the MC9S12ZVHY/MC9S12ZVHL Families available at the time of publication.

This introduction is intended to give an overview on several common topics like power supply, current injection etc.

A.1.1 Parameter Classification

The electrical parameters shown in this supplement are guaranteed by various methods. To give the customer a better understanding the following classification is used and the parameters are tagged accordingly in the tables where appropriate.

NOTE

This classification is shown in the column labeled “C” in the parameter tables where appropriate.

- P:Those parameters are guaranteed during production testing on each individual device.
- C: Those parameters are achieved in design characterization by measuring a statistically relevant sample size across process variations.
- T:Those parameters are achieved by design characterization on a small sample size from typical devices under typical conditions unless otherwise noted. All values shown in the typical column are within this category.
- D:Those parameters are derived mainly from simulations.