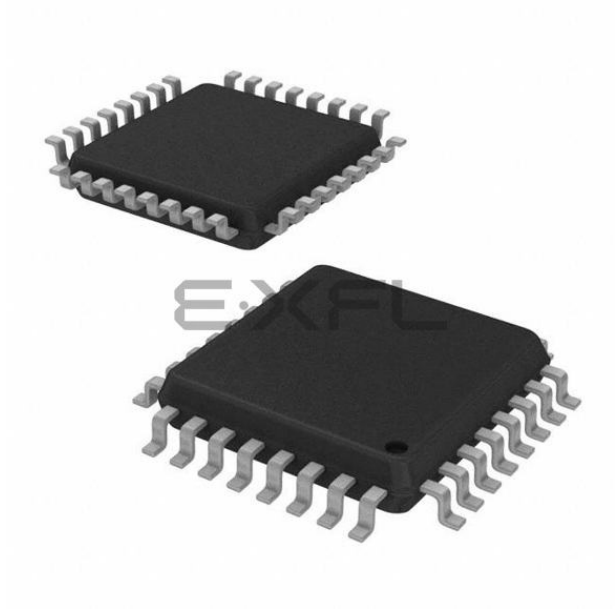


Welcome to E-XFL.COM

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, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvl12f0clc

Chapter 17
Pulse-Width Modulator (S12PWM8B8CV2)

Chapter 18
Serial Communication Interface (S12SCIV6)

1.6 Device Memory Map

Table 1-3 shows the device register memory map. All modules that can be instantiated more than once on S12 devices are listed with an index number, even if they are only instantiated once on this device family.

Table 1-3. Module Register Address Ranges

Address	Module	Size (Bytes)
0x0000–0x0003	Part ID Register Section 1.6.1, “Part ID Assignments	4
0x0004–0x000F	Reserved	12
0x0010–0x001F	INT	16
0x0020–0x006F	Reserved	80
0x0070–0x008F	MMC	32
0x0090–0x00FF	MMC Reserved	112
0x0100–0x017F	DBG	128
0x0180–0x01FF	Reserved	128
0x0200–0x037F	PIM	384
0x0380–0x039F	FTMRZ	32
0x03A0–0x03BF	Reserved	32
0x03C0–0x03CF	RAM ECC	16
0x03D0–0x03FF	Reserved	48
0x0400–0x042F	TIM1	48
0x0430–0x047F	Reserved	80
0x0480–0x04AF	PWM0	48
0x04B0–0x04FF	Reserved[1]	80
0x0500–0x052F	PWM1	48
0x0530–0x05BF	Reserved[1]	144
0x05C0–0x05EF	TIM0	48
0x05F0–0x05FF	Reserved	16
0x0600–0x063F	ADC	64
0x0640–0x067F	Reserved	64
0x0680–0x0687	DAC	8
0x0688–0x068F	Reserved	8
0x0690–0x0697	ACMP	8
0x0698–0x06BF	Reserved	40
0x06C0–0x06DF	CPMU	32
0x06E0–0x06EF	Reserved	16
0x06F0–0x06F7	BATS	8
0x06F8–0x06FF	Reserved	8

5.1.1 Glossary

Table 5-2. Glossary Of Terms

Term	Definition
DBG	On chip Debug Module
BDM	Active Background Debug Mode
CPU	S12Z CPU
SSC	Special Single Chip Mode (device operating mode
NSC	Normal Single Chip Mode (device operating mode)
BDCSI	Background Debug Controller Serial Interface. This refers to the single pin BKGD serial interface.
EWAIT	Optional S12 feature which allows external devices to delay external accesses until deassertion of EWAIT

5.1.2 Features

The BDC includes these distinctive features:

- Single-wire communication with host development system
- SYNC command to determine communication rate
- Genuine non-intrusive handshake protocol
- Enhanced handshake protocol for error detection and stop mode recognition
- Active out of reset in special single chip mode
- Most commands not requiring active BDM, for minimal CPU intervention
- Full global memory map access without paging
- Simple flash mass erase capability

5.1.3 Modes of Operation

S12 devices feature power modes (run, wait, and stop) and operating modes (normal single chip, special single chip). Furthermore, the operation of the BDC is dependent on the device security status.

5.1.3.1 BDC Modes

The BDC features module specific modes, namely disabled, enabled and active. These modes are dependent on the device security and operating mode. In active BDM the CPU ceases execution, to allow BDC system access to all internal resources including CPU internal registers.

5.1.3.2 Security and Operating mode Dependency

In device run mode the BDC dependency is as follows

- Normal modes, unsecure device

  General BDC operation available. The BDC is disabled out of reset.

elapse between the consecutive ERASE_FLASH commands then a timeout occurs, which forces a soft reset and initializes the sequence. The ERASE bit is cleared when the mass erase sequence has been completed. No ACK is driven.

During the mass erase operation, which takes many clock cycles, the command status is indicated by the ERASE bit in BDCCSR. Whilst a mass erase operation is ongoing, Always-available commands can be issued. This allows the status of the erase operation to be polled by reading BDCCSR to determine when the operation is finished.

The status of the flash array can be verified by subsequently reading the flash error flags to determine if the erase completed successfully.

ERASE_FLASH can be aborted by a SYNC pulse forcing a soft reset.

NOTE: Device Bus Frequency Considerations

The ERASE_FLASH command requires the default device bus clock frequency after reset. Thus the bus clock frequency must not be changed following reset before issuing an ERASE_FLASH command.

5.4.4.20 STEP1

This command is used to step through application code. In active BDM this command executes the next CPU instruction in application code. If enabled an ACK is driven.

If a STEP1 command is issued and the CPU is not halted, the command is ignored.

Using STEP1 to step through a CPU WAI instruction is explained in Section 5.1.3.3.2, “Wait Mode.

5.4.5 BDC Access Of Internal Resources

Unsuccessful read accesses of internal resources return a value of 0xEE for each data byte. This enables a debugger to recognize a potential error, even if neither the ACK handshaking protocol nor a status command is currently being executed. The value of 0xEE is returned in the following cases.

- Illegal address access, whereby ILLACC is set
- Invalid READ_SAME or DUMP_MEM sequence
- Invalid READ_Rn command (BDM inactive or CRN incorrect)
- Internal resource read with timeout, whereby NORESP is set

9.3.2.26 Reserved Register CPMUTEST2

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 S12CPMU_UHV's functionality.

Module Base + 0x001C

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

= Unimplemented or Reserved

Figure 9-35. Reserved Register CPMUTEST2

Read: Anytime

Write: Only in Special Mode

10.6.3.3 ADC List Usage and Conversion/Conversion Sequence Flow Description

It is the user's responsibility to make sure that the different lists do not overlap or exceed the system RAM area respectively the CSL does not exceed the NVM area if located in the NVM. The error flag IA_EIF will be set for accesses done outside the system RAM area and will cause an error interrupt if enabled for lists that are located in the system RAM.

Generic flow for ADC register load at conversion sequence start/restart:

- It is mandatory that the ADC is idle (no ongoing conversion or conversion sequence).
- It is mandatory to have at least one CSL with valid entries. See also Section 10.9.7.2, "Restart CSL execution with currently active CSL or Section 10.9.7.3, "Restart CSL execution with new/other CSL (alternative CSL becomes active CSL) — CSL swapping for more details on possible scenarios.
- A Restart Event occurs, which causes the index registers to be cleared (register ADCCIDX and ADCRIDX are cleared) and to point to the top of the corresponding lists (top of active RVL and CSL).
- Load conversion command to background conversion command register 1.
- The control bit(s) RSTA (and LDOK if set) are cleared.
- Wait for Trigger Event to start conversion.

Generic flow for ADC register load during conversion:

- The index registers ADCCIDX is incremented.
- The inactive background command register is loaded with a new conversion command.

Generic flow for ADC result storage at end of conversion:

- Index register ADCRIDX is incremented and the conversion result is stored in system RAM. As soon as the result is successfully stored, any conversion interrupt flags are set accordingly.
- At the conversion boundary the other background command register becomes active and visible in the ADC register map.
- If the last executed conversion command was of type "End Of Sequence", the ADC waits for the Trigger Event.
- If the last executed conversion command was of type "End Of List" and the ADC is configured in "Restart Mode", the ADC sets all related flags and stays idle awaiting a Restart Event to continue.
- If the last executed conversion command was of type "End Of List" and the ADC is configured in "Trigger Mode", the ADC sets all related flags and automatically returns to top of current CSL and is awaiting a Trigger Event to continue.
- If the last executed conversion command was of type "Normal Conversion" the ADC continues command execution in the order of the current CSL (continues conversion).

11.2.3 Block Diagram

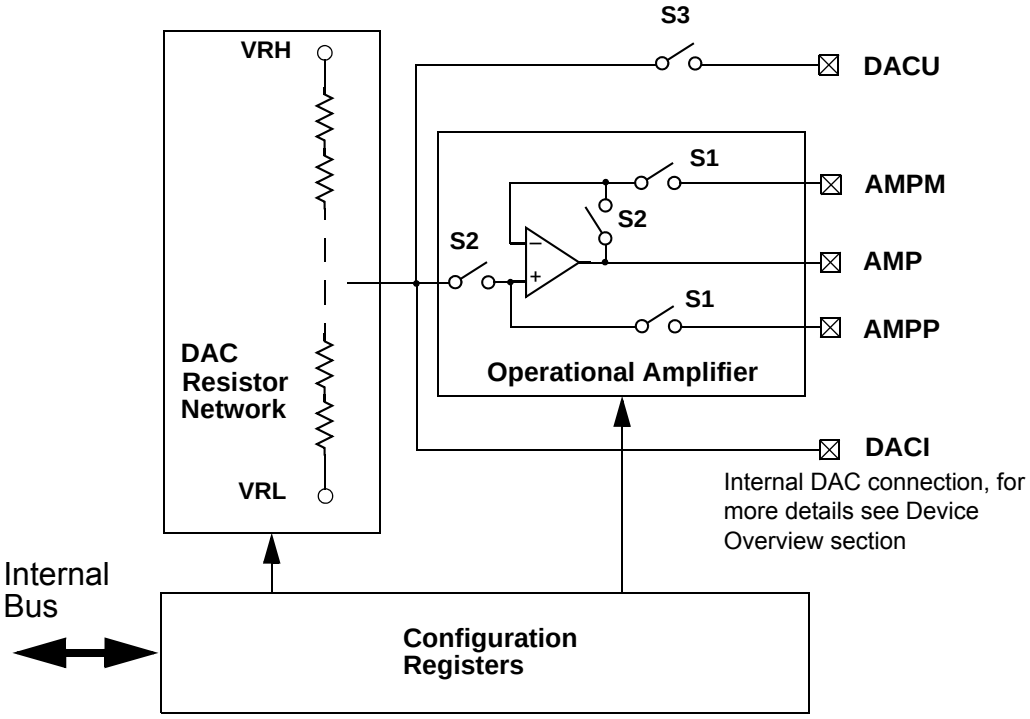


Figure 11-1. DAC_8B5V Block Diagram

11.3 External Signal Description

This section lists the name and description of all external ports.

11.3.1 DACU Output Pin

This analog pin drives the unbuffered analog output voltage from the DAC resistor network output, if the according mode is selected, see register bit DACM[2:0].

11.3.2 AMP Output Pin

This analog pin is used for the buffered analog output voltage from the operational amplifier output, if the according mode is selected, see register bit DACM[2:0].

11.3.3 AMPP Input Pin

This analog input pin is used as input signal for the operational amplifier positive input pin, if the according mode is selected, see register bit DACM[2:0].

11.4.2.2 Analog Output Voltage Level Register (DACVOL)

Module Base + 0x0002 Access: User read/write[1]

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

Figure 11-4. Analog Output Voltage Level Register (DACVOL)

1 Read: Anytime
Write: Anytime

Table 11-4. DACVOL Field Description

Field	Description
7:0 VOLTAGE[7:0]	**VOLTAGE** — This register defines (together with the FVR bit) the analog output voltage. For more detail see Equation 11-1 and Equation 11-2.

11.4.2.3 Reserved Register

Module Base + 0x0007 Access: User read/write[1]

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

Figure 11-5. Reserved Registerfv_dac_8b5v_RESERVED

1 Read: Anytime
Write: Only in special mode

NOTE

This reserved register bits are designed for factory test purposes only and are not intended for general user access. Writing to this register when in special modes can alter the modules functionality.

Table 13-25. Message Buffer Organization

Offset Address	Register	Access
0x00X0	IDR0 — Identifier Register 0	R/W
0x00X1	IDR1 — Identifier Register 1	R/W
0x00X2	IDR2 — Identifier Register 2	R/W
0x00X3	IDR3 — Identifier Register 3	R/W
0x00X4	DSR0 — Data Segment Register 0	R/W
0x00X5	DSR1 — Data Segment Register 1	R/W
0x00X6	DSR2 — Data Segment Register 2	R/W
0x00X7	DSR3 — Data Segment Register 3	R/W
0x00X8	DSR4 — Data Segment Register 4	R/W
0x00X9	DSR5 — Data Segment Register 5	R/W
0x00XA	DSR6 — Data Segment Register 6	R/W
0x00XB	DSR7 — Data Segment Register 7	R/W
0x00XC	DLR — Data Length Register	R/W
0x00XD	TBPR — Transmit Buffer Priority Register[1]	R/W
0x00XE	TSRH — Time Stamp Register (High Byte)	R
0x00XF	TSRL — Time Stamp Register (Low Byte)	R

[1] Not applicable for receive buffers

Figure 13-23 shows the common 13-byte data structure of receive and transmit buffers for extended identifiers. The mapping of standard identifiers into the IDR registers is shown in Figure 13-24.

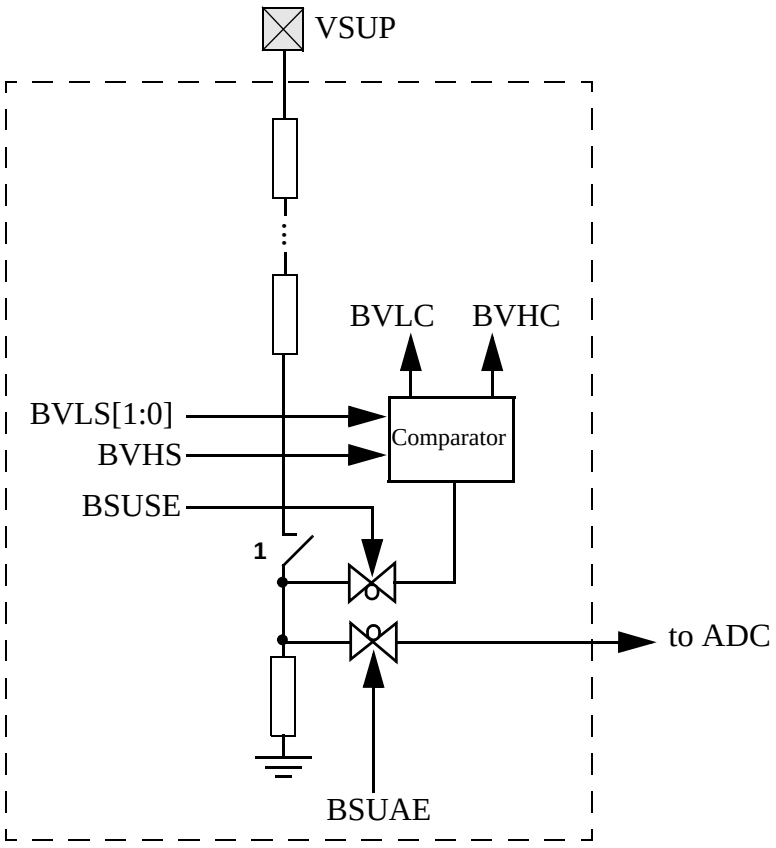
All bits of the receive and transmit buffers are ‘x’ out of reset because of RAM-based implementation[1]. All reserved or unused bits of the receive and transmit buffers always read ‘x’.

1. Exception: The transmit buffer priority registers are 0 out of reset.

14.1.3 Block Diagram

Figure 14-1 shows a block diagram of the BATS module. See device guide for connectivity to ADC channel.

Figure 14-1. BATSV3 Block Diagram



1automatically closed if BSUSE and/or BSUAE is active, open during Stop mode

14.2 External Signal Description

This section lists the name and description of all external ports.

14.2.1 VSUP — Voltage Supply Pin

This pin is the chip supply. It can be internally connected for voltage measurement. The voltage present at this input is scaled down by an internal voltage divider, and can be routed to the internal ADC or to a comparator.

In this way, the output of the PWM will always be either the old duty waveform or the new duty waveform, not some variation in between. If the channel is not enabled, then writes to the duty register will go directly to the latches as well as the buffer.

NOTE

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active duty due to the double buffering scheme.

See Section 17.4.2.3, “PWM Period and Duty” for more information.

NOTE

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time. If the polarity bit is one, the output starts high and then goes low when the duty count is reached, so the duty registers contain a count of the high time. If the polarity bit is zero, the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.

To calculate the output duty cycle (high time as a% of period) for a particular channel:

- Polarity = 0 (PPOL x =0)

  Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
- Polarity = 1 (PPOLx = 1)

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

For boundary case programming values, please refer to Section 17.4.2.8, “PWM Boundary Cases”.

Module Base + 0x001C = PWMDTY0, 0x001D = PWMDTY1, 0x001E = PWMDTY2, 0x001F = PWMDTY3
Module Base + 0x0020 = PWMDTY4, 0x0021 = PWMDTY5, 0x0022 = PWMDTY6, 0x0023 = PWMDTY7

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	1	1	1	1	1	1	1	1

Figure 17-14. PWM Channel Duty Registers (PWMDTYx)

[1] This register is available only when the corresponding channel exists and is reserved if that channel does not exist. Writes to a reserved register have no functional effect. Reads from a reserved register return zeroes.

Read: Anytime

Write: Anytime

17.4 Functional Description

17.4.1 PWM Clock Select

There are four available clocks: clock A, clock B, clock SA (scaled A), and clock SB (scaled B). These four clocks are based on the bus clock.

Clock A and B can be software selected to be 1, 1/2, 1/4, 1/8,..., 1/64, 1/128 times the bus clock. Clock SA uses clock A as an input and divides it further with a reloadable counter. Similarly, clock SB uses clock B as an input and divides it further with a reloadable counter. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB. Each PWM channel has the capability of selecting one of four clocks, clock A, Clock B, clock SA or clock SB.

The block diagram in Figure 17-15 shows the four different clocks and how the scaled clocks are created.

17.4.1.1 Prescale

The input clock to the PWM prescaler is the bus clock. It can be disabled whenever the part is in freeze mode by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode (freeze mode signal active) the input clock to the prescaler is disabled. This is useful for emulation in order to freeze the PWM. The input clock can also be disabled when all available PWM channels are disabled (PWMEx-0 = 0). This is useful for reducing power by disabling the prescale counter.

Clock A and clock B are scaled values of the input clock. The value is software selectable for both clock A and clock B and has options of 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 times the bus clock. The value selected for clock A is determined by the PCKA2, PCKA1, PCKA0 bits in the PWMPRCLK register. The value selected for clock B is determined by the PCKB2, PCKB1, PCKB0 bits also in the PWMPRCLK register.

17.4.1.2 Clock Scale

The scaled A clock uses clock A as an input and divides it further with a user programmable value and then divides this by 2. The scaled B clock uses clock B as an input and divides it further with a user programmable value and then divides this by 2. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

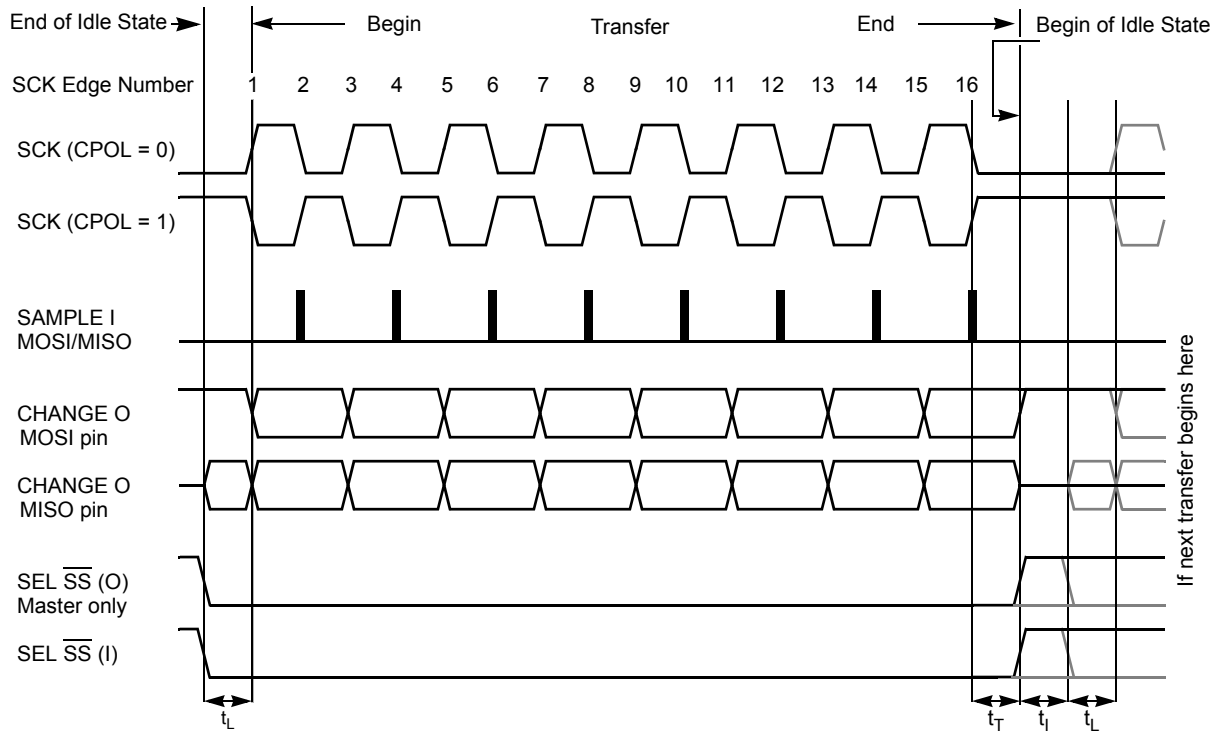
This process continues for a total of n^1 edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After $2n^1$ SCK edges:

- Data that was previously in the SPI data register of the master is now in the data register of the slave, and data that was in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating that the transfer is complete.

Figure 19-14 shows two clocking variations for CPHA = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.



t_L = Minimum leading time before the first SCK edge, not required for back-to-back transfers
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

Figure 19-14. SPI Clock Format 1 (CPHA = 1), with 8-Bit Transfer Width selected (XFRW = 0)

20.2.2 IIC_SDA — Serial Data Line Pin

This is the bidirectional serial data line (SDA) of the module, compatible to the IIC bus specification.

20.3 Memory Map and Register Definition

This section provides a detailed description of all memory and registers for the IIC module.

20.3.1 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 IBAD	R	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	0
	W								
0x0001 IBFD	R	IBC7	IBC6	IBC5	IBC4	IBC3	IBC2	IBC1	IBC0
	W								
0x0002 IBCR	R	IBEN	IBIE	MS/$\overline{SL}$	Tx/$\overline{Rx}$	TXAK	0	0	IBSWAI
	W						RSTA		
0x0003 IBSR	R	TCF	IAAS	IBB	IBAL	0	SRW	IBIF	RXAK
	W								
0x0004 IBDR	R	D7	D6	D5	D4	D3	D2	D1	D0
	W								
0x0005 IBCR2	R	GCEN	ADTYPE	0	0	0	ADR10	ADR9	ADR8
	W								

= Unimplemented or Reserved

Figure 20-2. IIC Register Summary

21.4 Functional Description

21.4.1 General

The S12LINPHYV2 module implements the physical layer of the LIN interface. This physical layer can be driven by the SCI (Serial Communication Interface) module or directly through the LPDR register.

21.4.2 Slew Rate and LIN Mode Selection

The slew rate can be selected for Electromagnetic Compatibility (EMC) optimized operation at 10.4 kbit/s and 20 kbit/s as well as at fast baud rate (up to 250 kbit/s) for test and programming. The slew rate can be chosen with the bits LPSLR[1:0] in the LIN Slew Rate Mode Register (LPSLRM). The default slew rate corresponds to 20 kbit/s.

The LIN Physical Layer can also be configured to be used for non-LIN applications (for example, to transmit a PWM pulse) by disabling the TxD-dominant timeout (LPDTDIS=1).

Changing the slew rate (LPSLRM Register) during transmission is not allowed in order to avoid unwanted effects. To change the register, the LIN Physical Layer must first be disabled (LPE=0). Once it is updated the LIN Physical Layer can be enabled again.

NOTE

For 20 kbit/s and Fast Mode communication speeds, the corresponding slew rate ***MUST*** be set; otherwise, the communication is not guaranteed (violation of the specified LIN duty cycles). For 10.4 kbit/s, the 20 kbit/s slew rate ***can*** be set but the EMC performance is worse. The up to 250 kbit/s slew rate must be chosen ***ONLY*** for fast mode, not for any of the 10.4 kbit/s or 20 kbit/s LIN compliant communication speeds.

21.4.2.1 10.4 kbit/s and 20 kbit/s

When the slew rate is chosen for 10.4 kbit/s or 20 kbit/s communication, a control loop is activated within the module to make the rise and fall times of the LIN bus independent from VLINSUP and the load on the bus.

21.4.2.2 Fast Mode (not LIN compliant)

Choosing this slew rate allows baud rates up to 250 kbit/s by having much steeper edges (please refer to electricals). As for the 10.4 kbit/s and 20 kbit/s modes, the slope control loop is also engaged. This mode is used for fast communication only, and the LIN electricals are not supported (for example, the LIN duty cycles).

A stronger external pullup resistor might be necessary to sustain communication speeds up to 250 kbit/s. The LIN signal (and therefore the receive LPRxD signal) might not be symmetrical for high baud rates with high loads on the bus.

CAUTION

Writing to the Flash registers while a Flash command is executing (that is indicated when the value of flag CCIF reads as ’0’) is not allowed. If such action is attempted, the result of the write operation will be unpredictable.

Writing to the Flash registers is allowed when the Flash is not busy executing commands (CCIF = 1) and during initialization right after reset, despite the value of flag CCIF in that case (refer to Section 22.6 for a complete description of the reset sequence).

.

Table 22-2. FTMRZ Memory Map

Global Address (in Bytes)	Description
0x0_0000 – 0x0_0FFF	Register Space
0x10_0000 – 0x1F_4000	EEPROM memory range. Allocation is device dependent.
0x1F_4000 – 0x1F_FFFF	NVM Resource Area[1] (see Figure 22-3)
0x80_0000 – 0xFD_FFFF	P-Flash memory range (Hardblock 0S). Allocation is device dependent.
0xFE_0000 – 0xFF_FFFF	P-Flash memory range (Hardblock 0N). Allocation is device dependent.

[1] See NVM Resource area description in Section 22.4.4

22.3.1 Module Memory Map

The P-Flash memory is located between global addresses 0x80_0000 and 0xFF_FFFF. The P-Flash is high aligned from 0xFF_FFFF. Thus, for example, a 128 KB P-Flash extends from 0xFF_FFFF to 0xFE_0000.

The flash configuration field is mapped to the same addresses independent of the P-Flash memory size, as shown in Figure 22-2.

The FPROT register, described in Section 22.3.2.9, can be set to protect regions in the Flash memory from accidental program or erase. Three separate memory regions, one growing upward from global address 0xFF_8000 in the Flash memory (called the lower region), one growing downward from global address 0xFF_FFFF in the Flash memory (called the higher region), and the remaining addresses in the Flash memory, can be activated for protection. The Flash memory addresses covered by these protectable regions are shown in the P-Flash memory map. The higher address region is mainly targeted to hold the boot loader code since it covers the vector space. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 22-3.

Table A-1. Revision History Table

Revision Number	Revision Date	Description Of Changes
0.50	14 March 2016	• added the latest characterization data, updated: Table A-11, Table A-19 • change voltage specification for MC9S12ZVL128/96/64 analog modules to VDDX ±3%:
0.60	31 March 2016	• update V_{BG} output voltage and V_{BG} voltage distribution specification: Table B-1
0.70	18 April 2016	• correct min V_{DDX} specification for MC9S12ZVL128/96/64 device: Table B-1
0.80	20 June 2016	• update Table A-19, add missing stop current for 85°C and 105°C, correct stop value for 125°C • update Table I-2, set ACMP input offset to 25mV
0.90	08 August 2017	• added 175°C parameters • update current injection consideration, section **Section C.1.1.4 Current Injection**
1.0	12 September 2017	• added 175°C Run and Wait current parameters
1.1	10 October 2017	• added Pin input leakage values for Pins PAD0 and PAD1 at 150°C < T_J < 175°C, Table A-10 • added Pin input leakage values for Pins PP1,PP3,PP5 and PP7 at 150°C < T_J < 175°C, Table A-10 • changed typical Reduced Performance Mode V_{DDX} Voltage to 5.0V, Table B-1
1.2	19 October 2017	• correct max value for Input leakage current on PP1, PP3, PP5 and PP7 for 150°C < T_J < 175°C on Table A-10
1.21	24 October 2017	• fixed minor bug in this revision history to make sure all updates are correct documented

A.1 General

This supplement contains the most accurate electrical information for the MC9S12ZVL-Family available at the time of publication.

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

Table A-2. Power Supplies

Mnemonic	Nominal Voltage	Description
VSS	0V	Ground pin for 1.8V core supply voltage generated by on chip voltage regulator
VDDX	5.0 V	5V power supply output for I/O drivers generated by on chip voltage regulator if VREG5VEN is set
VDDX	3.3 V	3.3V power supply output for I/O drivers generated by on chip voltage regulator if VREG5VEN is cleared
VSSX1	0V	Ground pin for I/O drivers

NOTE

Not every combination is offered. Table 1-2 lists available derivatives. The mask identifier suffix and the Tape & Reel suffix are always both omitted from the partnumber which is actually marked on the device.

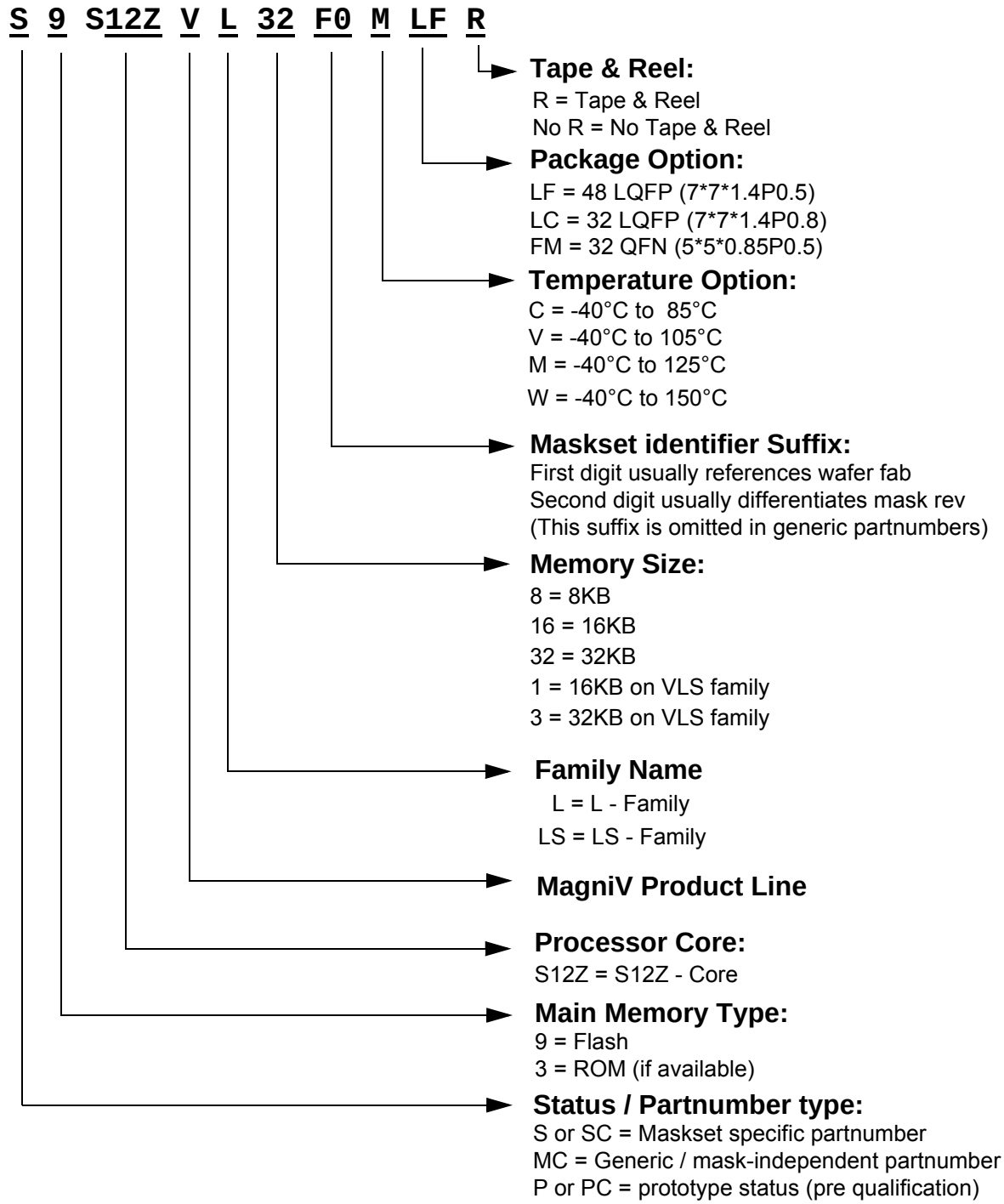


Figure N-1. Part Numbers for MC9S12ZVL(S)32/16/8

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0830-0x083F	CANTXFG	R W	See Section 13.3.3, “Programmer’s Model of Message Storage”							

O.22 0x0980-0x0987 LINPHY0

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0980	LP0DR	R	0	0	0	0	0	0	LPDR1	LPDR0
		W								
0x0981	LP0CR	R	0	0	0	0	LPE	RXONLY	LPWUE	LPPUE
		W								
0x0982	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0983	LPSLRM	R	LPDTDIS	0	0	0	0	0	LPSLR1	LPSLR0
		W								
0x0984	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0985	LP0SR	R	LPDT	0	0	0	0	0	0	0
		W								
0x0986	LP0IE	R	LPDTIE	LPOCIE	0	0	0	0	0	0
		W								
0x0987	LP0IF	R	LPDTIF	LPOCIF	0	0	0	0	0	0
		W								

O.23 0x0B40-0x0B47 PGA

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0B40	PGAEN	R	0	0	0	0	0		PGAOFFSCEN	PGAEN
		W								
0x0B41	PGACNTL	R	0	0	PGAREFSEL[1:0]		0	0	PGAINSEL[1:0]	
		W								
0x0B42	PGAGAIN	R	0	0	0	0	PGAGAIN[3:0]			
		W								
0x0B43	PGAOFFSET	R	0		PGAOFFSET[5:0]					
		W								
0x0B44-0xB46	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0B47	Reserved	R	0	0	0	0	0	Reserved	Reserved	Reserved
		W								