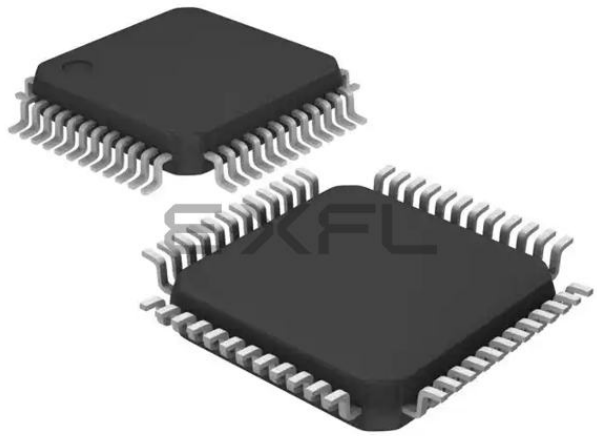


Welcome to E-XFL.COM

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, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	28
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 10x12b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvca64f0clf

1.4.1 S12Z Central Processor Unit (CPU)

The S12Z CPU is a revolutionary high-speed core, with code size and execution efficiencies over the S12X CPU. The S12Z CPU also provides a linear memory map eliminating the inconvenience and performance impact of page swapping.

- Harvard Architecture - parallel data and code access
- 3-stage pipeline
- 32-bit wide instruction and databus
- 32-bit ALU
- 24-bit addressing (16 MB linear address space)
- Instructions and Addressing modes optimized for C-Programming and Compiler
  - MAC unit 32bit += 32bit*32bit
  - Hardware divider
  - Single cycle multi-bit shifts (Barrel shifter)
  - Special instructions for fixed point math
- Unimplemented opcode traps
- Unprogrammed byte value (0xFF) defaults to SWI instruction

1.4.1.1 Background Debug Controller (BDC)

- 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

1.4.1.2 Debugger (DBG)

- Three comparators (A, B and D)
  - Comparator A compare the full address bus and full 32-bit data bus
  - Comparators B and D compare the full address bus onlyEach comparator can be configured to monitor PC addresses or addresses of data accesses
  - Each comparator can select either read or write access cycles
  - Comparator matches can force state sequencer state transitions
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin ≤ Address ≤ Addmax
  - Outside address range match mode, Address < Addmin or Address > Addmax

1.13.3 Flash IFR Mapping

Table 1-16. Flash IFR Mapping

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IFR Byte Address
ADC0 reference conversion using VDDA/VSSA																0x1F_C040 & 0x1F_C041
ADC0 reference conversion using PAD1/PAD0																0x1F_C042 & 0x1F_C043
Reserved																0x1F_C050 & 0x1F_C051
Reserved																0x1F_C052 & 0x1F_C053
Reserved																0x1F_C054 & 0x1F_C055
Reserved																0x1F_C056 & 0x1F_C057
Reserved																0x1F_C058 & 0x1F_C059
Reserved																0x1F_C05A & 0x1F_C05B
		ACLKTR[5:0][1] (CPMU)										HTTR[3:0][2] (CPMU)				0x1F_C0B8 & 0x1F_C0B9
	TCTRIM[4:0] (CPMU)[3]					IRCTRIM[9:0] (CPMU)[3]										0x1F_C0BA & 0x1F_C0BB

[1]see Section 8.3.2.17 Autonomous Clock Trimming Register (CPMUACLKTR)
[2]see Section 8.3.2.20 High Temperature Trimming Register (CPMUHTTR)
[3]see Section 8.3.2.21 S12CPMU_UHV_V7 IRC1M Trim Registers (CPMUIRCTRIMH / CPMUIRCTRIML)

which cannot be disabled, are always handled by the CPU and have a fixed priority levels. A priority level of 0 effectively disables the associated I-bit maskable interrupt request.

If more than one interrupt request is configured to the same interrupt priority level the interrupt request with the higher vector address wins the prioritization.

The following conditions must be met for an I-bit maskable interrupt request to be processed.

1. The local interrupt enabled bit in the peripheral module must be set.
2. The setup in the configuration register associated with the interrupt request channel must meet the following conditions:
   a) The priority level must be set to non zero.
   b) The priority level must be greater than the current interrupt processing level in the condition code register (CCW) of the CPU (PRIOLVL[2:0] > IPL[2:0]).
3. The I-bit in the condition code register (CCW) of the CPU must be cleared.
4. There is no access violation interrupt request pending.
5. There is no SYS, SWI, SPARE, TRAP, Machine Exception or $\overline{\text{XIRQ}}$ request pending.

NOTE

All non I-bit maskable interrupt requests always have higher priority than I-bit maskable interrupt requests. If an I-bit maskable interrupt request is interrupted by a non I-bit maskable interrupt request, the currently active interrupt processing level (IPL) remains unaffected. It is possible to nest non I-bit maskable interrupt requests, e.g., by nesting SWI, SYS or TRAP calls.

5.4.2.1 Interrupt Priority Stack

The current interrupt processing level (IPL) is stored in the condition code register (CCW) of the CPU. This way the current IPL is automatically pushed to the stack by the standard interrupt stacking procedure. The new IPL is copied to the CCW from the priority level of the highest priority active interrupt request channel which is configured to be handled by the CPU. The copying takes place when the interrupt vector is fetched. The previous IPL is automatically restored from the stack by executing the RTI instruction.

5.4.3 Priority Decoder

The INT module contains a priority decoder to determine the relative priority for all interrupt requests pending for the CPU.

A CPU interrupt vector is not supplied until the CPU requests it. Therefore, it is possible that a higher priority interrupt request could override the original exception which caused the CPU to request the vector. In this case, the CPU will receive the highest priority vector and the system will process this exception first instead of the original request.

If the interrupt source is unknown (for example, in the case where an interrupt request becomes inactive after the interrupt has been recognized, but prior to the vector request), the vector address supplied to the CPU defaults to that of the spurious interrupt vector.

Table 6-22 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if INST is set, as matches based on instructions reaching the execution stage are data independent.

Table 6-22. Read or Write Comparison Logic Table

RWE Bit	RW Bit	RW Signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write match
1	0	1	No match
1	1	0	No match
1	1	1	Read match

6.3.2.13 Debug Comparator B Address Register (DBGBAH, DBGBAM, DBGBAL)

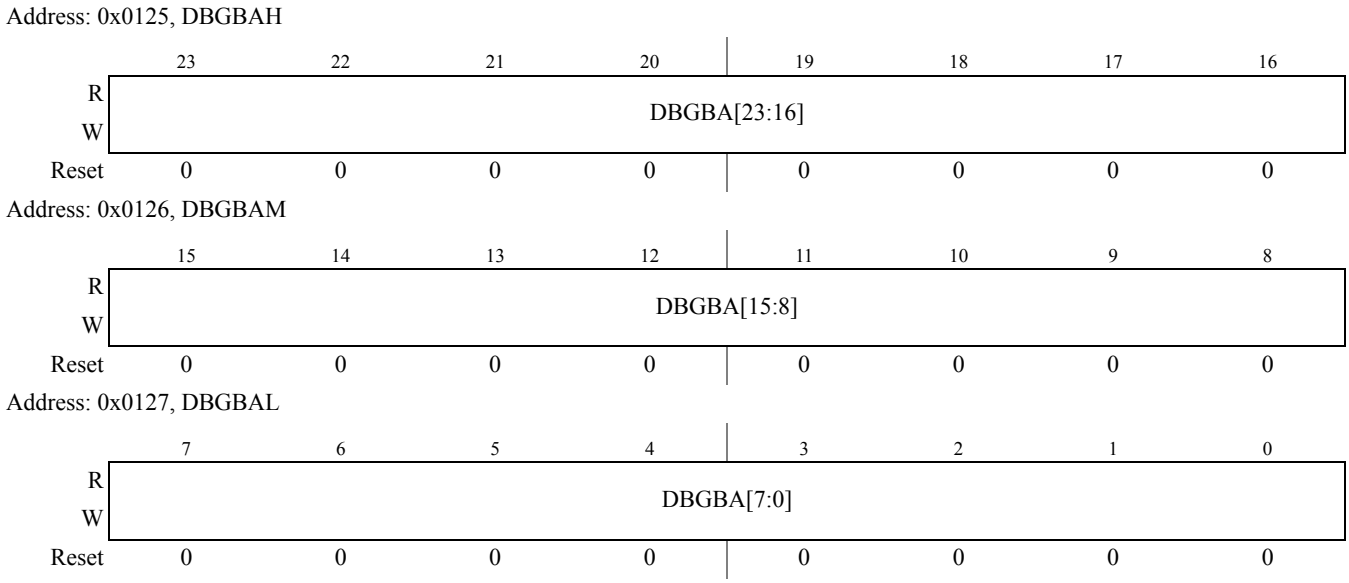


Figure 6-16. Debug Comparator B Address Register

Read: Anytime.

Write: If DBG not armed.

Table 6-23. DBGBAH, DBGBAM, DBGBAL Field Descriptions

Field	Description
23–16 DBGBA [23:16]	**Comparator Address Bits [23:16]**— These comparator address bits control whether the comparator compares the address bus bits [23:16] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one
15–0 DBGBA [15:0]	**Comparator Address Bits**[15:0]— These comparator address bits control whether the comparator compares the address bus bits [15:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

accesses to mask out individual data bus bits and to use R/W access qualification in the comparison. Comparators can be configured to monitor a range of addresses.

When configured for data access comparisons, the match is generated if the address (and optionally data) of a data access matches the comparator value.

Configured for monitoring opcode addresses, the match is generated when the associated opcode reaches the execution stage of the instruction queue, but before execution of that opcode.

When a match with a comparator register value occurs, the associated control logic can force the state sequencer to another state (see Figure 6-19).

The state sequencer can transition freely between the states 1, 2 and 3. On transition to Final State, a breakpoint can be generated and the state sequencer returns to state0, disarming the DBG.

Independent of the comparators, state sequencer transitions can be forced by the external event input or by writing to the TRIG bit in the DBGC1 control register.

6.4.2 Comparator Modes

The DBG contains three comparators, A, B, and D. Each comparator compares the address stored in DBGXAH, DBGXAM, and DBGXAL with the PC (opcode addresses) or selected address bus (data accesses). Furthermore, comparator A can compare the data buses to values stored in DBGXD3-0 and allow data bit masking.

The comparators can monitor the buses for an exact address or an address range. The comparator configuration is controlled by the control register contents and the range control by the DBGC2 contents.

The comparator control register also allows the type of data access to be included in the comparison through the use of the RWE and RW bits. The RWE bit controls whether the access type is compared for the associated comparator and the RW bit selects either a read or write access for a valid match.

The INST bit in each comparator control register is used to determine the matching condition. By setting INST, the comparator matches opcode addresses, whereby the databus, data mask, RW and RWE bits are ignored. The comparator register must be loaded with the exact opcode address.

The comparator can be configured to match memory access addresses by clearing the INST bit.

Each comparator match can force a transition to another state sequencer state (see Section 6.4.3, “Events”).

Once a successful comparator match has occurred, the condition that caused the original match is not verified again on subsequent matches. Thus if a particular data value is matched at a given address, this address may not contain that data value when a subsequent match occurs.

Match[0, 1, 3] map directly to Comparators [A, B, D] respectively, except in range modes (see Section 6.3.2.2, “Debug Control Register2 (DBGC2)”). Comparator priority rules are described in the event priority section (Section 6.4.3.4, “Event Priorities”).

6.5 Application Information

6.5.1 Avoiding Unintended Breakpoint Re-triggering

Returning from an instruction address breakpoint using an RTI or BDC GO command without PC modification, returns to the instruction that generated the breakpoint. If an active breakpoint or trigger still exists at that address, this can re-trigger, disarming the DBG. If configured for BDM breakpoints, the user must apply the BDC STEP1 command to increment the PC past the current instruction.

If configured for SWI breakpoints, the DBG can be re configured in the SWI routine. If a comparator match occurs at an SWI vector address then a code SWI and DBG breakpoint SWI could occur simultaneously. In this case the SWI routine is executed twice before returning.

6.5.2 Breakpoints from other S12Z sources

The DBG is neither affected by CPU BGND instructions, nor by BDC BACKGROUND commands.

9.2.1 Modes of Operation

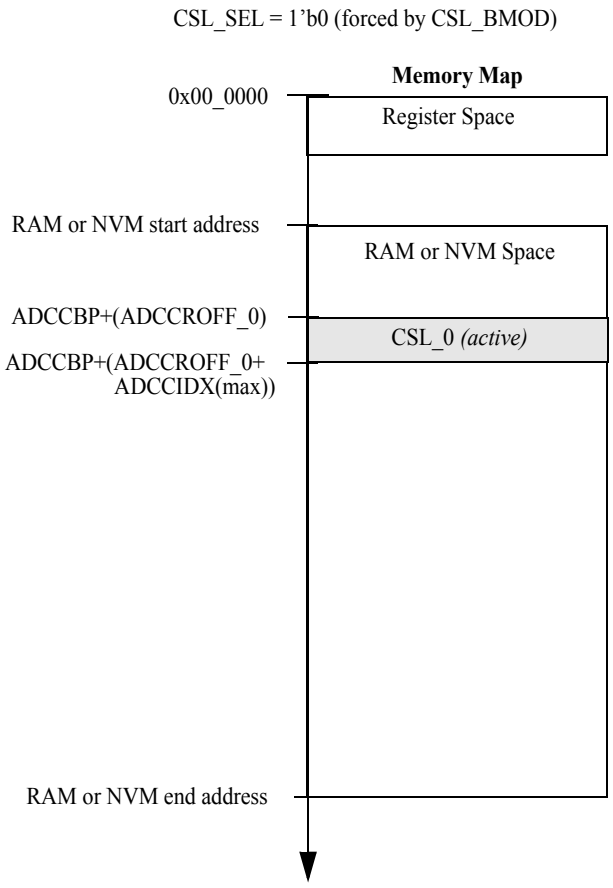
9.2.1.1 Conversion Modes

This architecture provides **single**, **multiple,** or **continuous conversion** on a **single channel** or on **multiple channels based on the Command Sequence List**.

9.2.1.2 MCU Operating Modes

- **MCU Stop Mode**
  Before issuing an MCU Stop Mode request the ADC should be idle (no conversion or conversion sequence or Command Sequence List ongoing).
  If a conversion, conversion sequence, or CSL is in progress when an MCU Stop Mode request is issued, a Sequence Abort Event occurs automatically and any ongoing conversion finish. After the Sequence Abort Event finishes, if the STR_SEQA bit is set (STR_SEQA=1), then the conversion result is stored and the corresponding flags are set. If the STR_SEQA bit is cleared (STR_SEQA=0), then the conversion result is not stored and the corresponding flags are not set. The microcontroller then enters MCU Stop Mode without SEQAD_IF being set.
  Alternatively, the Sequence Abort Event can be issued by software before an MCU Stop Mode request. As soon as flag SEQAD_IF is set the MCU Stop Mode request can be is issued.
  With the occurrence of the MCU Stop Mode Request until exit from Stop Mode all flow control signals (RSTA, SEQA, LDOK, TRIG) are cleared.

  After exiting MCU Stop Mode, the following happens in the order given with expected event(s) depending on the conversion flow control mode:
  — In ADC conversion flow control mode “Trigger Mode” a Restart Event is expected to simultaneously set bits TRIG and RSTA, causing the ADC to execute the Restart Event (CMD_IDX and RVL_IDX cleared) followed by the Trigger Event. The Restart Event can be generated automatically after exit from MCU Stop Mode if bit AUT_RSTA is set.
  — In ADC conversion flow control mode “Restart Mode”, a Restart Event is expected to set bit RSTA only (ADC already aborted at MCU Stop Mode entry hence bit SEQA must not be set simultaneously) causing the ADC to execute the Restart Event (CDM_IDX and RVL_IDX cleared). The Restart Event can be generated automatically after exit from MCU Stop Mode if bit AUT_RSTA is set.
  — The RVL buffer select (RVL_SEL) is not changed if a CSL is in process at MCU Stop Mode request. Hence the same buffer will be used after exit from Stop Mode that was used when the Stop Mode request occurred.



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

Figure 9-32. Command Sequence List Schema in Single Buffer Mode

While the ADC is enabled, one CSL is active (indicated by bit CSL_SEL) and the corresponding list should not be modified anymore. At the same time the alternative CSL can be modified to prepare the ADC for new conversion sequences in CSL double buffered mode. When the ADC is enabled, the command address registers (ADCCBP, ADCCROFF_0/2, ADCCIDX) are read only and register ADCCIDX is under control of the ADC.

The comparator outputs BVLC and BVHC are forced to zero if the comparator is disabled (configuration bit BSUSE is cleared). If the software disables the comparator during a high or low Voltage condition (BVHC or BVLC active), then an additional interrupt is generated. To avoid this behavior the software must disable the interrupt generation before disabling the comparator.

The BATS interrupt vector is named in Table 10-6. Vector addresses and interrupt priorities are defined at MCU level.

The module internal interrupt sources are combined into one module interrupt signal.

Table 10-6. BATS Interrupt Sources

Module Interrupt Source	Module Internal Interrupt Source	Local Enable
BATS Interrupt (BATI)	BATS Voltage Low Condition Interrupt (BVLI)	BVLIE = 1
	BATS Voltage High Condition Interrupt (BVHI)	BVHIE = 1

10.4.2.1 BATS Voltage Low Condition Interrupt (BVLI)

To use the Voltage Low Interrupt the Level Sensing must be enabled (BSUSE =1).

If measured when

a) V_{LBI1} selected with BVLS[1:0] = 0x0
$V_{measure} < V_{LBI1\_A}$ (falling edge) or $V_{measure} < V_{LBI1\_D}$ (rising edge)

or when

b) V_{LBI2} selected with BVLS[1:0] = 0x1 at pin VSUP
$V_{measure} < V_{LBI2\_A}$ (falling edge) or $V_{measure} < V_{LBI2\_D}$ (rising edge)

or when

c) V_{LBI3} selected with BVLS[1:0] = 0x2
$V_{measure} < V_{LBI3\_A}$ (falling edge) or $V_{measure} < V_{LBI3\_D}$ (rising edge)

or when

d) V_{LBI4} selected with BVLS[1:0] = 0x3
$V_{measure} < V_{LBI4\_A}$ (falling edge) or $V_{measure} < V_{LBI4\_D}$ (rising edge)

then BVLC is set. BVLC status bit indicates that a low voltage at pin VSUP is present. The Low Voltage Interrupt flag (BVLIF) is set to 1 when the Voltage Low Condition (BVLC) changes state . The Interrupt flag BVLIF can only be cleared by writing a 1. If the interrupt is enabled by bit BVLIE the module requests an interrupt to MCU (BATI).

10.4.2.2 BATS Voltage High Condition Interrupt (BVHI)

To use the Voltage High Interrupt the Level Sensing must be enabled (BSUSE=1).

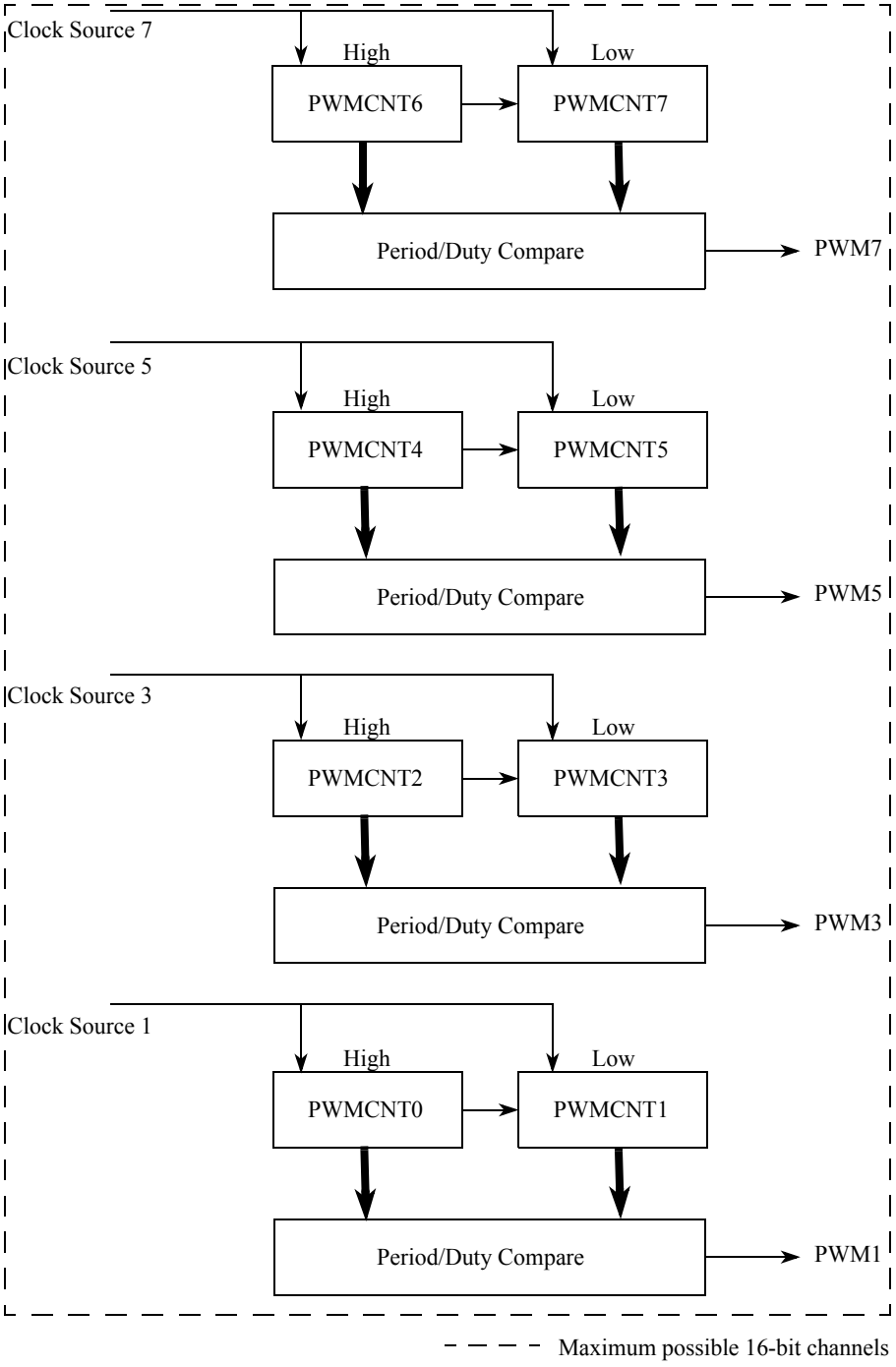


Figure 13-21. PWM 16-Bit Mode

Once concatenated mode is enabled (CONxx bits set in PWMCTL register), enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit. In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output is disabled.

17.5.4.1 Voltage Failure Interrupts

A voltage failure error is detected if voltage levels on the CAN bus lines exceed the specified limits.

The voltages on both lines CANH and CANL are monitored continuously for crossing the lower and higher thresholds, V_{H0}, V_{H5} and V_{L0}, V_{L5}, respectively.

A comparator output transition to error level results in setting the corresponding status bit in CAN Physical Layer Status Register (CPSR). A change of a status bit sets the related interrupt flag in CAN Physical Layer Interrupt Flag Register (CPIF).

The flags are used as interrupt sources of which either of the four can generate a CPI interrupt if the common enable bit CPVFIE in CAN Physical Layer Interrupt Enable Register (CPIE) is set.

17.5.4.2 CPTXD-Dominant Timeout Interrupt

For network lock-up protection of the CAN bus, the CAN physical layer features a permanent CPTXD-dominant timeout monitor. When the CPTXD signal has been dominant for more than $t_{CPTXDDT}$ the transmitter is disabled by entering listen-only mode and the bus is released to recessive state. The CPDT status and CPDTIF interrupt flags are both set.

To re-enable the transmitter, the CPDTIF flag must be cleared. If the CPTXD input is dominant or dominant timeout status is still active (CPDT=1), the CAN Physical Layer stays in listen-only mode and CPDTIF is set again after some microseconds to indicate that the attempt has failed. If CPTXD is recessive and CPDT=0 it takes 1 to 2 µs after clearing CPDTIF for returning to normal mode.

The flag is used as an interrupt source to generate a CPI interrupt if the enable bit CPDTIE in CAN Physical Layer Interrupt Enable Register (CPIE) is set.

17.5.4.3 Over-Current Interrupt

An over-current error is detected if current levels on the CAN bus lines exceed the specified limits while driving a dominant bit.

The current levels on both lines CANH and CANL are monitored continuously for crossing the thresholds I_{CANHOC} and I_{CANLOC}, respectively.

A comparator output transition to error level results in setting the corresponding interrupt flag in CAN Physical Layer Interrupt Flag Register (CPIF).

The flags are the direct interrupt sources of which either of the two can generate a CPI interrupt if the common enable bit CPOCIE in CAN Physical Layer Interrupt Enable Register (CPIE) is set.

17.6 Initialization/Application Information

17.6.1 Initialization Sequence

Setup for immediate CAN communication:

1. Enable and configure MSCAN

Module Base + 0x0001 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
W								
Reset:	0	0	0	1	0	0	0	1

= Unimplemented

Figure 18-5. MSCAN Control Register 1 (CANCTL1)

[1] Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1), except CANE which is write once in normal and anytime in special system operation modes when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1)

Table 18-3. CANCTL1 Register Field Descriptions

Field	Description
7 CANE	**MSCAN Enable** 0 MSCAN module is disabled 1 MSCAN module is enabled
6 CLKSRC	**MSCAN Clock Source** — This bit defines the clock source for the MSCAN module (only for systems with a clock generation module; Section 18.4.3.2, “Clock System,” and Section Figure 18-43., “MSCAN Clocking Scheme,”). 0 MSCAN clock source is the oscillator clock 1 MSCAN clock source is the bus clock
5 LOOPB	**Loopback Self Test Mode** — When this bit is set, the MSCAN performs an internal loopback which can be used for self test operation. The bit stream output of the transmitter is fed back to the receiver internally. The RXCAN input is ignored and the TXCAN output goes to the recessive state (logic 1). The MSCAN behaves as it does normally when transmitting and treats its own transmitted message as a message received from a remote node. In this state, the MSCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated. 0 Loopback self test disabled 1 Loopback self test enabled
4 LISTEN	**Listen Only Mode** — This bit configures the MSCAN as a CAN bus monitor. When LISTEN is set, all valid CAN messages with matching ID are received, but no acknowledgement or error frames are sent out (see Section 18.4.4.4, “Listen-Only Mode”). In addition, the error counters are frozen. Listen only mode supports applications which require “hot plugging” or throughput analysis. The MSCAN is unable to transmit any messages when listen only mode is active. 0 Normal operation 1 Listen only mode activated
3 BORM	**Bus-Off Recovery Mode** — This bit configures the bus-off state recovery mode of the MSCAN. Refer to Section 18.5.2, “Bus-Off Recovery,” for details. 0 Automatic bus-off recovery (see Bosch CAN 2.0A/B protocol specification) 1 Bus-off recovery upon user request
2 WUPM	**Wake-Up Mode** — If WUPE in CANCTL0 is enabled, this bit defines whether the integrated low-pass filter is applied to protect the MSCAN from spurious wake-up (see Section 18.4.5.5, “MSCAN Sleep Mode”). 0 MSCAN wakes up on any dominant level on the CAN bus 1 MSCAN wakes up only in case of a dominant pulse on the CAN bus that has a length of T_{wup}

Table 18-22. CANIDAR4–CANIDAR7 Register Field Descriptions

Field	Description
7-0 AC[7:0]	**Acceptance Code Bits** — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

18.3.2.18 MSCAN Identifier Mask Registers (CANIDMR0–CANIDMR7)

The identifier mask register specifies which of the corresponding bits in the identifier acceptance register are relevant for acceptance filtering. To receive standard identifiers in 32 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1 and CANIDMR5 to “don’t care.” To receive standard identifiers in 16 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1, CANIDMR3, CANIDMR5, and CANIDMR7 to “don’t care.”

Module Base + 0x0014 to Module Base + 0x0017 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Figure 18-22. MSCAN Identifier Mask Registers (First Bank) — CANIDMR0–CANIDMR3

1 Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 18-23. CANIDMR0–CANIDMR3 Register Field Descriptions

Field	Description
7-0 AM[7:0]	**Acceptance Mask Bits** — If a particular bit in this register is cleared, this indicates that the corresponding bit in the identifier acceptance register must be the same as its identifier bit before a match is detected. The message is accepted if all such bits match. If a bit is set, it indicates that the state of the corresponding bit in the identifier acceptance register does not affect whether or not the message is accepted. 0 Match corresponding acceptance code register and identifier bits 1 Ignore corresponding acceptance code register bit

Module Base + 0x001C to Module Base + 0x001F Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Figure 18-23. MSCAN Identifier Mask Registers (Second Bank) — CANIDMR4–CANIDMR7

18.4.5.5 MSCAN Sleep Mode

The CPU can request the MSCAN to enter this low power mode by asserting the SLPRQ bit in the CANCTL0 register. The time when the MSCAN enters sleep mode depends on a fixed synchronization delay and its current activity:

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into sleep mode.

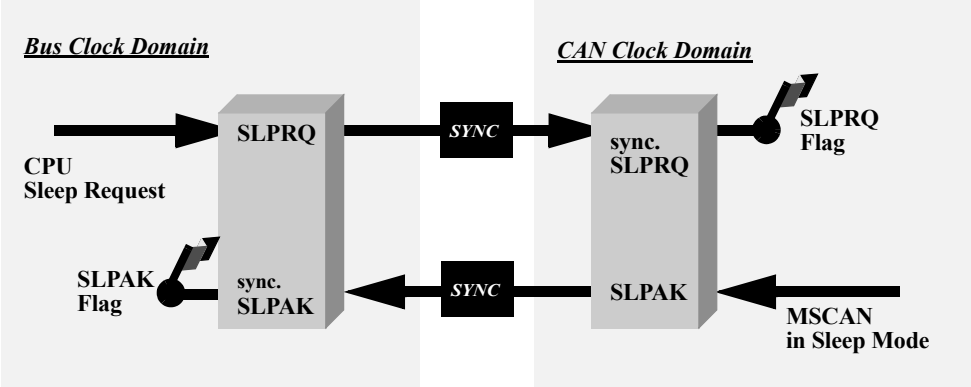


Figure 18-46. Sleep Request / Acknowledge Cycle

NOTE

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into sleep mode directly depends on the exact sequence of operations.

If sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 18-46). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode.

When in sleep mode (SLPRQ = 1 and SLPAK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. TXCAN remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

19.3.4 AMPM Input Pin

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

19.4 Memory Map and Register Definition

This sections provides the detailed information of all registers for the DAC_8B5V module.

19.4.1 Register Summary

Figure 19-2 shows the summary of all implemented registers inside the DAC_8B5V module.

NOTE

Register Address = Module Base Address + Address Offset, where the Module Base Address is defined at the MCU level and the Address Offset is defined at the module level.

Address Offset / Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 DACCTL	R	FVR	DRIVE	0	0	0	DACM[2:0]		
	W								
0x0001 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0002 DACVOL	R	VOLTAGE[7:0]							
	W								
0x0003 - 0x0006 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0007 Reserved	R	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	W								

= Unimplemented

Figure 19-2. DAC_8B5V Register Summary

19.4.2 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.

Chapter 20 5V Analog Comparator (ACMPV2)

Table 20-1. Revision History

Rev. No. (Item No.)	Date (Submitted By)	Sections Affected	Substantial Change(s)
V02.01	22 Mar 2013		• Various changes based on shared review feedback • Separated ACIE and ACIF in two registers • Made ACPSEL and ACNSEL write only while module disabled • Changed initialization delay to 127 bus clock cycles •
V02.02	29 Feb 2013		• Minor corrections based on feedback from shared review
V02.03	25 Jun 2013		• Corrected ACMPO behavior in shutdown mode • Corrected ACMPC1 write restriction • Made input change during initialization delay description non-ambiguous

20.1 Introduction

The analog comparator (ACMP) provides a circuit for comparing two analog voltages. The comparator circuit is designed to operate across the full range between 0V and VDDA supply voltage (rail-to-rail operation).

20.2 Features

The ACMP has the following features:

- 0V to VDDA supply rail-to-rail inputs
- Low offset
- Up to 4 inputs selectable as inverting and non-inverting comparator inputs:
  - 2 low-impedance inputs with selectable low pass filter for external pins
  - 2 high-impedance inputs with fixed filter for SoC-internal signals
- Selectable hysteresis
- Selectable interrupt on rising edge, falling edge, or rising and falling edges of comparator output
- Option to output comparator signal on an external pin with selectable polarity
- Support for triggering timer input capture events
- Operational over supply range from 3V-5% to 5V+10%
- Temperature range (T_J): -40°C to 175°C

22.3.2.9.1 P-Flash Protection Restrictions

In Normal Single Chip Mode the general guideline is that P-Flash protection can only be added and not removed. Table 22-22 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

Table 22-22. P-Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

1 Allowed transitions marked with X, see Figure 22-14 for a definition of the scenarios.

22.3.2.10 EEPROM Protection Register (DFPROT)

The DFPROT register defines which EEPROM sectors are protected against program and erase operations.

Offset Module Base + 0x0009

	7	6	5	4	3	2	1	0
R	DPOPEN	0	DPS[5:0]					
W								
Reset	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]

Figure 22-15. EEPROM Protection Register (DFPROT)

1 Loaded from Flash configuration field, during reset sequence.

The (unreserved) bits of the DFPROT register are writable in Normal Single Chip Mode with the restriction that protection can be added but not removed. Writes in Normal Single Chip Mode must increase the DPS value and the DPOPEN bit can only be written from 1 (protection disabled) to 0 (protection enabled). If the DPOPEN bit is set, the state of the DPS bits is irrelevant.All DPOPEN/DPS bit registers are writable without restriction in Special Single Chip Mode.

During the reset sequence, fields DPOPEN and DPS of the DFPROT register are loaded with the contents of the EEPROM protection byte in the Flash configuration field at global address 0xFF_FE0D located in

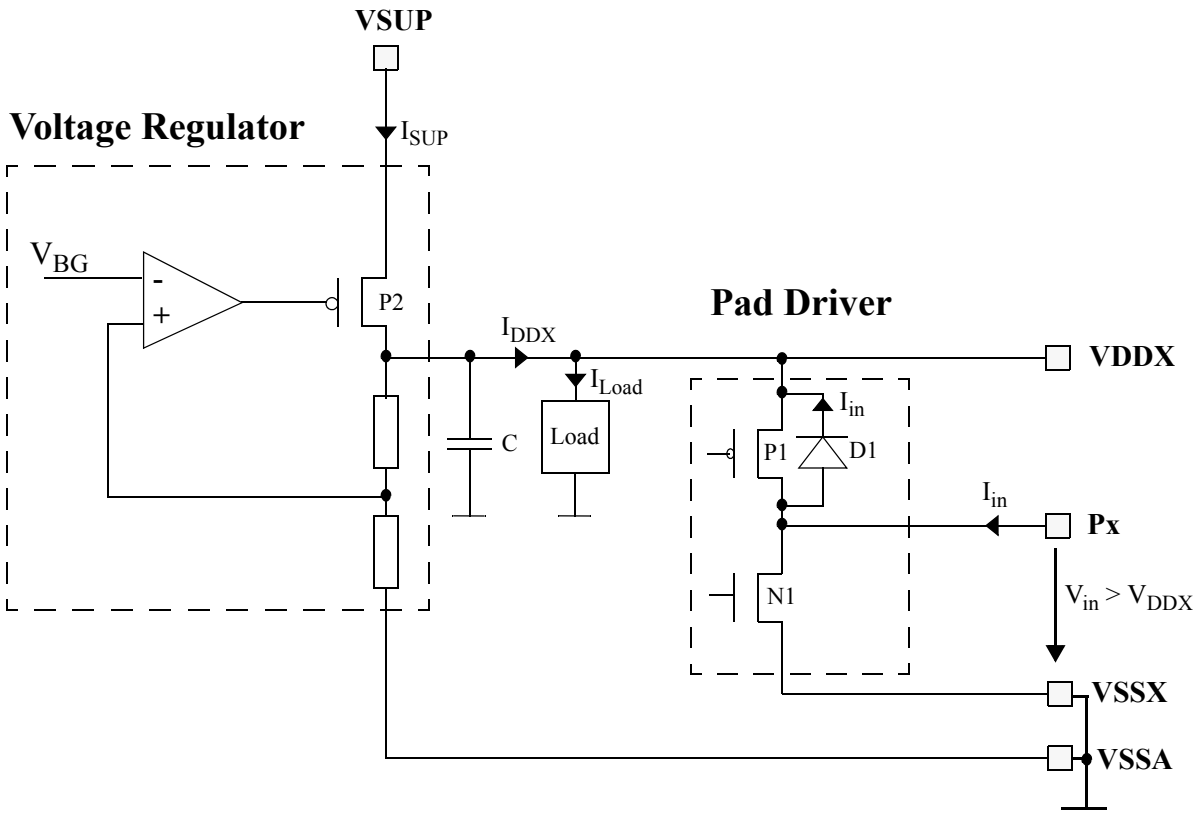
A.1.2.4 TEST

This pin is used for production testing only. The TEST pin must be tied to ground in all applications.

A.1.3 Current Injection

Power supply must maintain regulation within operating V_{DDX} or V_{DD} range during instantaneous and operating maximum current conditions. **Figure A-1.** shows a 5 V GPIO pad driver and the on chip voltage regulator with VDDX output. It shows also the power and ground pins VSUP, VDDX, VSSX and VSSA. Px represents any 5 V GPIO pin. Assume Px is configured as an input. The pad driver transistors P1 and N1 are switched off (high impedance). If the voltage V_{in} on Px is greater than V_{DDX} a positive injection current I_{in} will flow through diode D1 into VDDX node. If this injection current I_{in} is greater than I_{Load}, the internal power supply VDDX may go out of regulation. Ensure the external V_{DDX} load will shunt current greater than maximum injection current. This is the greatest risk when the MCU is not consuming power; e.g., if no system clock is present, or if the clock rate is very low which would reduce overall power consumption.

Figure A-1. Current Injection on GPIO Port if $V_{in} > V_{DDX}$



A.1.4 Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only. A functional operation outside these ranges is not guaranteed. Stress beyond these limits may affect the reliability or cause permanent damage of the device.

Figure L-2. 64 LQFP Exposed Pad Package