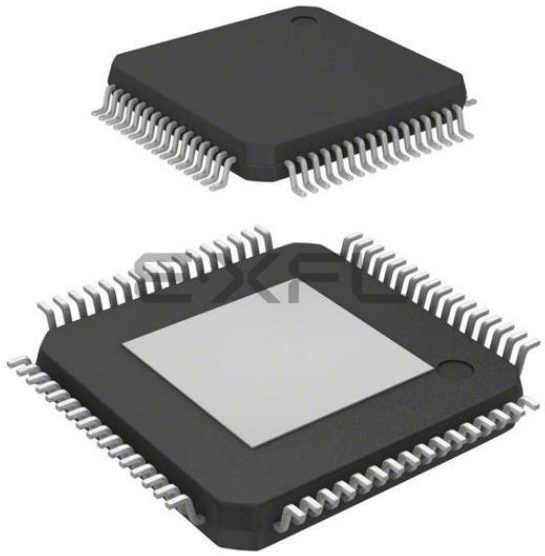


Welcome to E-XFL.COM

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	42
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 16x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP Exposed Pad
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvc64f0mkhr

Table 1-6. MC9S12ZVC-Family Pin Summary

LQFP			Function					Power Supply	Internal Pull Resistor	
64	48	Pin	1st Func.	2nd Func.	3rd Func.	4th Func.	5th Func.		CTRL	Reset State
1	1	VSUP	—	—	—	—	—	V_{SUP}	—	—
2	2	PL1	HVI1	KWL1	—	—	—	V_{DDX}	—	—
3	3	PL0	HVI0	KWL0	—	—	—	V_{DDX}	—	—
4	4	BCTL			—	—	—	V_{DDX}		
5	5	BCTLC			—	—	—	V_{DDX}		
6	6	PE0	EXTAL		—	—	—	V_{DDX}	PERE/ PPSE	Down
7	7	PE1	XTAL	—	—	—	—	V_{DDX}	PERE/ PPSE	Down
8		PJ1	SCL0	TXD0	—	—	—	V_{DDX}	PERJ/ PPSJ	Up
9		PJ0	SDA0	RXD0	—	—	—	V_{DDX}	PERJ/ PPSJ	Up
10	—	PAD11	KWAD11	AN11	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
11	—	PAD10	KWAD10	AN10	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
12	8	PAD9	KWAD9	AN9	AMP	—	—	V_{DDA}	PERADL/ PPSADL	Off
13	9	PAD8	KWAD8	AN8	AMPM	—	—	V_{DDA}	PERADL/ PPSADL	Off
14	10	PAD7	KWAD7	AN7	AMPP	—	—	V_{DDA}	PERADH/ PPSADH	Off
15	11	PAD6	KWAD6	AN6	DACU	—	—	V_{DDA}	PERADH/ PPSADH	Off
16	12	PAD5	KWAD5	AN5	ACMPO1	—	—	V_{DDA}	PERADL/ PPSADL	Off
17	13	PAD4	KWAD4	AN4	ACMP1_1	—	—	V_{DDA}	PERADL/ PPSADL	Off
18	14	VSSA					—	V_{DDA}		
19	15	VDDA					—	V_{DDA}		
20	16	PAD3	KWAD3	AN3	VRH_0	ACMP1_0	—	V_{DDA}	PERADL/ PPSADL	Off
21	17	PAD2	KWAD2	AN2	ACMPO0	—	—	V_{DDA}	PERADL/ PPSADL	Off

1.9.3.3 ADC Internal Channels

The ADC internal channel mapping is shown in Table 1-8.

Table 1-8. ADC Channel Assignment

ADCCMD_L[CH_SEL]							
[5]	[4]	[3]	[2]	[1]	[0]	**Analog Input Channel**	**Usage**
0	0	0	0	0	0	V_{RL}	
0	0	0	0	0	1	V_{RH}	
0	0	0	0	1	0	$(V_{RH}-V_{RL})/2$	
0	0	0	0	1	1	Reserved	
0	0	0	1	0	0	Reserved	
0	0	0	1	0	1	Reserved	
0	0	0	1	1	0	Reserved	
0	0	0	1	1	1	Reserved	
0	0	1	0	0	0	Internal_0	RESERVED
0	0	1	0	0	1	Internal_1	Bandgap Voltage V_{BG} or Chip temperature sensor V_{HT} see Section 8.3.2.14 High Temperature Control Register (CPMUHTCTL)
0	0	1	0	1	0	Internal_2	Flash Voltage V_{DDF}
0	0	1	0	1	1	Internal_3	RESERVED
0	0	1	1	0	0	Internal_4	V_{SUP} see Section 10.3.2.1 BATS Module Enable Register (BATE)
0	0	1	1	0	1	Internal_5	High voltage input port L0 see Section 2.3.4.10 Port L ADC Connection Enable Register (PTAENL)
0	0	1	1	1	0	Internal_6	High voltage input port L1 Section 2.3.4.10 Port L ADC Connection Enable Register (PTAENL)
0	0	1	1	1	1	Internal_7	RESERVED

1.9.4 TIM0 and TIM1 Clock Source Connectivity

The clock for TIM1 is the device core clock generated in the CPMU module. (maximum core clock is 64MHz)

The clock for TIM0 is the device bus clock generated in the CPMU module. (maximum bus clock 32MHz)

Vector Address	Interrupt Source	CCR Mask	Local Enable	Wake up from STOP	Wake up from WAIT
Vector base + 0x0F0 to Vector base + 0x0C4	Reserved				
Vector base + 0x0C0	Port L interrupt (Key Wakeup)	I bit	PIEL(PIEL[1:0])	Yes	Yes
Vector base + 0x0BC to Vector base + 0x0B0	Reserved				
Vector base + 0x0AC	TIM1 timer channel 0	I bit	TIM1TIE (C0I)	No	Yes
Vector base + 0x0A8	TIM1 timer channel 1	I bit	TIM1TIE (C1I)	No	Yes
Vector base + 0x0A4	TIM1 timer channel 2	I bit	TIM1TIE (C2I)	No	Yes
Vector base + 0x0A0	TIM1 timer channel 3	I bit	TIM1TIE (C3I)	No	Yes
Vector base + 0x09C to Vector base + 0x090	Reserved				
Vector base + 0x08C	TIM1 timer overflow	I bit	TIM1TSCR2 (TOF)	No	Yes
Vector base + 0x088 to Vector base + 0x064	Reserved				
Vector base + 0x060	IIC0	I bit	IBCR(IBIE)	No	Yes
Vector base + 0x05C	SENTTX	I bit	INTEN(xxIE)	No	Yes
Vector base + 0x058 to Vector base + 0x000	Reserved				

1.12.2 Effects of Reset

When a reset occurs, MCU registers and control bits are initialized. Refer to the respective block sections for register reset states.

On each reset, the Flash module executes a reset sequence to load Flash configuration registers.

1.12.2.1 Flash Configuration Reset Sequence Phase

On each reset, the Flash module will hold CPU activity while loading Flash module registers from the Flash memory. If double faults are detected in the reset phase, Flash module protection and security may be active on leaving reset. This is explained in more detail in the Flash module description.

1.12.2.2 Reset While Flash Command Active

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.

3.3.2.2 BDC Control Status Register Low (BDCCSRL)

Register Address: This register is not in the device memory map. It is accessible using BDC inherent addressing commands

	7	6	5	4	3	2	1	0
R W	WAIT	STOP	RAMWF	OVRUN	NORESP	RDINV	ILLACC	ILLCMD
Reset	0	0	0	0	0	0	0	0

Figure 3-4. BDC Control Status Register Low (BDCCSRL)

Read: BDC access only.

Write: Bits [7:5], [3:0] BDC access only, restricted to flag clearing by writing a “1” to the bit position.
Write: Bit 4 never. It can only be cleared by a SYNC pulse.

If ACK handshaking is enabled then BDC commands with ACK causing a BDCCSRL[3:1] flag setting condition also generate a long ACK pulse. Subsequent commands that are executed correctly generate a normal ACK pulse. Subsequent commands that are not correctly executed generate a long ACK pulse. The first ACK pulse after WAIT or STOP have been set also generates a long ACK. Subsequent ACK pulses are normal, whilst STOP and WAIT remain set.

Long ACK pulses are not immediately generated if an overrun condition is caused by the host driving the BKGD pin low whilst a target ACK is pending, because this would conflict with an attempted host transmission following the BKGD edge. When a whole byte has been received following the offending BKGD edge, the OVRUN bit is still set, forcing subsequent ACK pulses to be long.

Unimplemented BDC opcodes causing the ILLCMD bit to be set do not generate a long ACK because this could conflict with further transmission from the host. If the ILLCMD is set for another reason, then a long ACK is generated for the current command if it is a BDC command with ACK.

Table 3-6. BDCCSRL Field Descriptions

Field	Description
7 WAIT	**WAIT Indicator Flag** — Indicates that the device entered wait mode. Writing a “1” to this bit whilst in wait mode has no effect. Writing a “1” after exiting wait mode, clears the bit. 0 Device did not enter wait mode 1 Device entered wait mode.
6 STOP	**STOP Indicator Flag** — Indicates that the CPU requested stop mode following a STOP instruction. Writing a “1” to this bit whilst not in stop mode clears the bit. Writing a “1” to this bit whilst in stop mode has no effect. This bit can only be set when the BDC is enabled. 0 Device did not enter stop mode 1 Device entered stop mode.
5 RAMWF	**RAM Write Fault** — Indicates an ECC double fault during a BDC write access to RAM. Writing a “1” to this bit, clears the bit. 0 No RAM write double fault detected. 1 RAM write double fault detected.

Table 3-8. BDC Command Summary (continued)

Command Mnemonic	Command Classification	ACK	Command Structure	Description
DUMP_MEM.sz_WS	Non-Intrusive	No	(0x33+4 x sz)/d/ss/rd.sz	Dump (read) memory based on operand size (sz) and report status. Used with READ_MEM{_WS} to dump large blocks of memory. An initial READ_MEM{_WS} is executed to set up the starting address of the block and to retrieve the first result. Subsequent DUMP_MEM{_WS} commands retrieve sequential operands.
FILL_MEM.sz	Non-Intrusive	Yes	(0x12+4 x sz)/wd.sz/dack	Fill (write) memory based on operand size (sz). Used with WRITE_MEM to fill large blocks of memory. An initial WRITE_MEM is executed to set up the starting address of the block and to write the first operand. Subsequent FILL_MEM commands write sequential operands.
FILL_MEM.sz_WS	Non-Intrusive	No	(0x13+4 x sz)/wd.sz/d/ss	Fill (write) memory based on operand size (sz) and report status. Used with WRITE_MEM{_WS} to fill large blocks of memory. An initial WRITE_MEM{_WS} is executed to set up the starting address of the block and to write the first operand. Subsequent FILL_MEM{_WS} commands write sequential operands.
GO	Active Background	Yes	0x08/dack	Resume CPU user code execution
GO_UNTIL[2]	Active Background	Yes	0x0C/dack	Go to user program. ACK is driven upon returning to active background mode.
NOP	Non-Intrusive	Yes	0x00/dack	No operation
READ_Rn	Active Background	Yes	(0x60+CRN)/dack/rd32	Read the requested CPU register
READ_MEM.sz	Non-Intrusive	Yes	(0x30+4 x sz)/ad24/dack/rd.sz	Read the appropriately-sized (sz) memory value from the location specified by the 24-bit address
READ_MEM.sz_WS	Non-Intrusive	No	(0x31+4 x sz)/ad24/d/ss/rd.sz	Read the appropriately-sized (sz) memory value from the location specified by the 24-bit address and report status
READ_DBGTB	Non-Intrusive	Yes	(0x07)/dack/rd32/dack/rd32	Read 64-bits of DBG trace buffer
READ_SAME.sz	Non-Intrusive	Yes	(0x50+4 x sz)/dack/rd.sz	Read from location. An initial READ_MEM defines the address, subsequent READ_SAME reads return content of same address
READ_SAME.sz_WS	Non-Intrusive	No	(0x51+4 x sz)/d/ss/rd.sz	Read from location. An initial READ_MEM defines the address, subsequent READ_SAME reads return content of same address
READ_BDCCSR	Always Available	No	0x2D/rd16	Read the BDCCSR register

3.4.5.2.2 READ_SAME Effects Of Variable Access Size

READ_SAME uses the unadjusted address given in the previous READ_MEM command as a base address for subsequent READ_SAME commands. When the READ_MEM and READ_SAME size parameters differ then READ_SAME uses the original base address buts aligns 32-bit and 16-bit accesses, where those accesses would otherwise cross the aligned 4-byte boundary. Table 3-12 shows some examples of this.

Table 3-12. Consecutive READ_SAME Accesses With Variable Size

Row	Command	Base Address	00	01	10	11
1	READ_MEM.32	0x004003	Accessed	Accessed	Accessed	Accessed
2	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
3	READ_SAME.16	—			Accessed	Accessed
4	READ_SAME.08	—				Accessed
5	READ_MEM.08	0x004000	Accessed			
6	READ_SAME.08	—	Accessed			
7	READ_SAME.16	—	Accessed	Accessed		
8	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
9	READ_MEM.08	0x004002			Accessed	
10	READ_SAME.08	—			Accessed	
11	READ_SAME.16	—			Accessed	Accessed
12	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
13	READ_MEM.08	0x004003				Accessed
14	READ_SAME.08	—				Accessed
15	READ_SAME.16	—			Accessed	Accessed
16	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
17	READ_MEM.16	0x004001		Accessed	Accessed	
18	READ_SAME.08	—		Accessed		
19	READ_SAME.16	—		Accessed	Accessed	
20	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
21	READ_MEM.16	0x004003			Accessed	Accessed
22	READ_SAME.08	—				Accessed
23	READ_SAME.16	—			Accessed	Accessed
24	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed

3.4.6 BDC Serial Interface

The BDC communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDC.

The BDC serial interface uses an internal clock source, selected by the CLKSW bit in the BDCCSR register. This clock is referred to as the target clock in the following explanation.

The BDC serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if during a command 512 clock cycles occur between falling edges from the host. The timeout forces the current command to be discarded.

The BKGD pin is a pseudo open-drain pin and has a weak on-chip active pull-up that is enabled at all times. It is assumed that there is an external pull-up and that drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief drive-high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for host-to-target is shown in Figure 3-6 and that of target-to-host in Figure 3-7 and Figure 3-8. All cases begin when the host drives the BKGD pin low to generate a falling edge. Since the host and target operate from separate clocks, it can take the target up to one full clock cycle to recognize this edge; this synchronization uncertainty is illustrated in Figure 3-6. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 3-6 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires the pin be driven high no later than eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

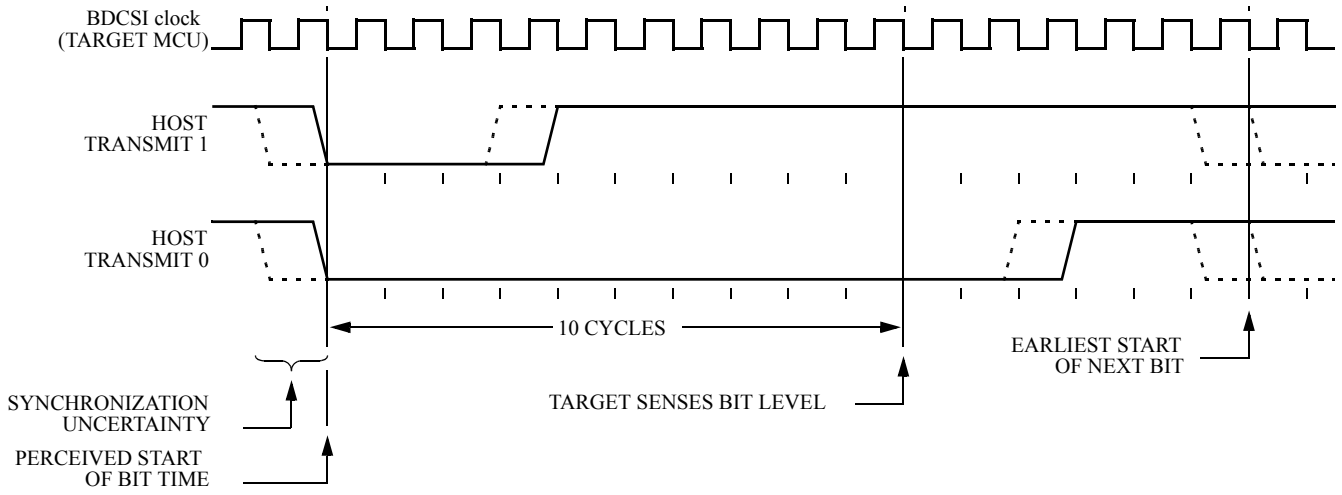


Figure 3-6. BDC Host-to-Target Serial Bit Timing

Figure 3-7 shows the host receiving a logic 1 from the target system. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low

6.3.2.15 Debug Comparator D Address Register (DBGDAH, DBGDAM, DBGDAL)

Address: 0x0145, DBGDAH

	23	22	21	20	19	18	17	16
R W	DBGDA[23:16]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0146, DBGDAM

	15	14	13	12	11	10	9	8
R W	DBGDA[15:8]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0147, DBGDAL

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

Figure 6-18. Debug Comparator D Address Register

Read: Anytime.

Write: If DBG not armed.

Table 6-26. DBGDAH, DBGDAM, DBGDAL Field Descriptions

Field	Description
23–16 DBGDA [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 DBGDA [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

6.4 Functional Description

This section provides a complete functional description of the DBG module.

6.4.1 DBG Operation

The DBG module operation is enabled by setting ARM in DBGC1. When armed it can be used to generate breakpoints to the CPU. The DBG module is made up of comparators, control logic, and the state sequencer, Figure 6-1.

The comparators monitor the bus activity of the CPU. Comparators can be configured to monitor opcode addresses (effectively the PC address) or data accesses. Comparators can be configured during data

8.3.2.23 S12CPMU_UHV_V7 Protection Register (CPMUPROT)

This register protects the clock configuration registers from accidental overwrite:

CPMUSYNR, CPMUREFDIV, CPMUCLKS, CPMUPLL, CPMUIRCTRIMH/L, CPMUOSC and CPMUOSC2

Module Base + 0x001B

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

Figure 8-32. S12CPMU_UHV_V7 Protection Register (CPMUPROT)

Read: Anytime

Write: Anytime

Field	Description
PROT	**Clock Configuration Registers Protection Bit** — This bit protects the clock configuration registers from accidental overwrite (see list of protected registers above): Writing 0x26 to the CPMUPROT register clears the PROT bit, other write accesses set the PROT bit. 0 Protection of clock configuration registers is disabled. 1 Protection of clock configuration registers is enabled. (see list of protected registers above).

When COP is enabled, the program must write $55 and $AA (in this order) to the CPMUARMCOP register during the selected time-out period. Once this is done, the COP time-out period is restarted. If the program fails to do this and the COP times out, a COP reset is generated. Also, if any value other than $55 or $AA is written, a COP reset is generated.

Windowed COP operation is enabled by setting WCOP in the CPMUCOP register. In this mode, writes to the CPMUARMCOP register to clear the COP timer must occur in the last 25% of the selected time-out period. A premature write will immediately reset the part.

In MCU Normal Mode the COP time-out period (CR[2:0]) and COP window (WCOP) setting can be automatically pre-loaded at reset release from NVM memory (if values are defined in the NVM by the application). By default the COP is off and no window COP feature is enabled after reset release via NVM memory. The COP control register CPMUCOP can be written once in an application in MCU Normal Mode to update the COP time-out period (CR[2:0]) and COP window (WCOP) setting loaded from NVM memory at reset release. Any value for the new COP time-out period and COP window setting is allowed except COP off value if the COP was enabled during pre-load via NVM memory.
The COP clock source select bits can not be pre-loaded via NVM memory at reset release. The IRC clock is the default COP clock source out of reset.
The COP clock source select bits (COPOSCSEL0/1) and ACLK clock control bit in Stop Mode (CSAD) can be modified until the CPMUCOP register write once has taken place. Therefore these control bits should be modified before the final COP time-out period and window COP setting is written.
The CPMUCOP register access to modify the COP time-out period and window COP setting in MCU Normal Mode after reset release must be done with the WRTMASK bit cleared otherwise the update is ignored and this access does not count as the write once.

8.5.6 Power-On Reset (POR)

The on-chip POR circuitry detects when the internal supply VDD drops below an appropriate voltage level. The POR is deasserted, if the internal supply VDD exceeds an appropriate voltage level (voltage levels not specified, because the internal supply can not be monitored externally). The POR circuitry is always active. It acts as LVR in Stop Mode.

8.5.7 Low-Voltage Reset (LVR)

The on-chip LVR circuitry detects when one of the supply voltages VDD, VDDX and VDDF drops below an appropriate voltage level. If LVR is deasserted the MCU is fully operational at the specified maximum speed. The LVR assert and deassert levels for the supply voltage VDDX are V_{LVRXA} and V_{LVRXD} and are specified in the device Reference Manual.The LVR circuitry is active in Run- and Wait Mode.

9.4.2.16 ADC Command Register 1 (ADCCMD_1)

A command which contains reserved bit settings causes the error flag CMD_EIF being set and ADC cease operation.

Module Base + 0x0015

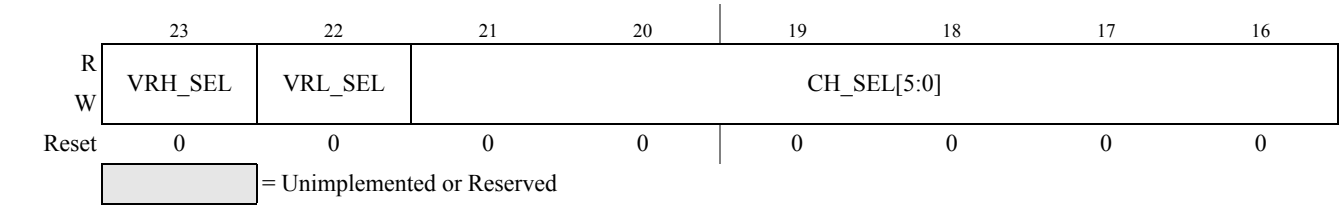


Figure 9-19. ADC Command Register 1 (ADCCMD_1)

Read: Anytime

Write: Only writable if bit SMOD_ACC is set
(see also Section 9.4.2.2, “ADC Control Register 1 (ADCCTL_1) bit SMOD_ACC description for more details)

Table 9-22. ADCCMD_1 Field Descriptions

Field	Description
23 VRH_SEL	**Reference High Voltage Select Bit** — This bit selects the high voltage reference for current conversion. 0 VRH_0 input selected as high voltage reference. 1 VRH_1 input selected as high voltage reference.
22 VRL_SEL	**Reference Low Voltage Select Bit** — This bit selects the voltage reference for current conversion. 0 VRL_0 input selected as low voltage reference. 1 VRL_1 input selected as low voltage reference.
21-16 CH_SEL[5:0]	**ADC Input Channel Select Bits** — These bits select the input channel for the current conversion. See Table 9-23 for channel coding information.

NOTE

If bit SMOD_ACC is set modifying this register must be done carefully - only when no conversion and conversion sequence is ongoing.

Table 9-23. Analog Input Channel Select

CH_SEL[5]	CH_SEL[4]	CH_SEL[3]	CH_SEL[2]	CH_SEL[1]	CH_SEL[0]	Analog Input Channel
0	0	0	0	0	0	VRL_0/1
0	0	0	0	0	1	VRH_0/1
0	0	0	0	1	0	(VRH_0/1 + VRL_0/1) / 2
0	0	0	0	1	1	Reserved
0	0	0	1	0	0	Reserved
0	0	0	1	0	1	Reserved
0	0	0	1	1	0	Reserved

Table 11-4. OC7M Field Descriptions

Field	Description
7:0 OC7M[7:0]	**Output Compare 7 Mask** — A channel 7 event, which can be a counter overflow when TTOV[7] is set or a successful output compare on channel 7, overrides any channel 6:0 compares. For each OC7M bit that is set, the output compare action reflects the corresponding OC7D bit. 0 The corresponding OC7Dx bit in the output compare 7 data register will not be transferred to the timer port on a channel 7 event, even if the corresponding pin is setup for output compare. 1 The corresponding OC7Dx bit in the output compare 7 data register will be transferred to the timer port on a channel 7 event. **Note:** The corresponding channel must also be setup for output compare (IOSx = 1 and OCPDx = 0) for data to be transferred from the output compare 7 data register to the timer port.

11.3.2.4 Output Compare 7 Data Register (OC7D)

[1] .

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R W	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
Reset	0	0	0	0	0	0	0	0

Figure 11-9. Output Compare 7 Data Register (OC7D)

Read: Anytime

Write: Anytime

Table 11-5. OC7D Field Descriptions

Field	Description
7:0 OC7D[7:0]	**Output Compare 7 Data** — A channel 7 event, which can be a counter overflow when TTOV[7] is set or a successful output compare on channel 7, can cause bits in the output compare 7 data register to transfer to the timer port data register depending on the output compare 7 mask register.

11.3.2.5 Timer Count Register (TCNT)

Module Base + 0x0004

	15	14	13	12	11	10	9	9
R W	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
Reset	0	0	0	0	0	0	0	0

Figure 11-10. Timer Count Register High (TCNTH)

15.3.2 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.

15.3.2.1 SPI Control Register 1 (SPICR1)

Module Base +0x0000

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Figure 15-3. SPI Control Register 1 (SPICR1)

Read: Anytime

Write: Anytime

Table 15-1. SPICR1 Field Descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set. 0 SPI interrupts disabled. 1 SPI interrupts enabled.
6 SPE	**SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset. 0 SPI disabled (lower power consumption). 1 SPI enabled, port pins are dedicated to SPI functions.
5 SPTIE	**SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set. 0 SPTEF interrupt disabled. 1 SPTEF interrupt enabled.
4 MSTR	**SPI Master/Slave Mode Select Bit** — This bit selects whether the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state. 0 SPI is in slave mode. 1 SPI is in master mode.
3 CPOL	**SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Active-high clocks selected. In idle state SCK is low. 1 Active-low clocks selected. In idle state SCK is high.
2 CPHA	**SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Sampling of data occurs at odd edges (1,3,5,...) of the SCK clock. 1 Sampling of data occurs at even edges (2,4,6,...) of the SCK clock.

Module Base + 0x000E — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 18-18. MSCAN Receive Error Counter (CANRXERR)

[1] Read: Only when in sleep mode (SLPRQ = 1 and SLPAK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
Write: Unimplemented

NOTE

Reading this register when in any other mode other than sleep or initialization mode may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

18.3.2.16 MSCAN Transmit Error Counter (CANTXERR)

This register reflects the status of the MSCAN transmit error counter.

Module Base + 0x000F — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 18-19. MSCAN Transmit Error Counter (CANTXERR)

[1] Read: Only when in sleep mode (SLPRQ = 1 and SLPAK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
Write: Unimplemented

NOTE

Reading this register when in any other mode other than sleep or initialization mode, may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

18.4.4.2 Special System Operating Modes

The MSCAN module behaves as described within this specification in all special system operating modes. Write restrictions which exist on specific registers in normal modes are lifted for test purposes in special modes.

18.4.4.3 Emulation Modes

In all emulation modes, the MSCAN module behaves just like in normal system operating modes as described within this specification.

18.4.4.4 Listen-Only Mode

In an optional CAN bus monitoring mode (listen-only), the CAN node is able to receive valid data frames and valid remote frames, but it sends only “recessive” bits on the CAN bus. In addition, it cannot start a transmission.

If the MAC sub-layer is required to send a “dominant” bit (ACK bit, overload flag, or active error flag), the bit is rerouted internally so that the MAC sub-layer monitors this “dominant” bit, although the CAN bus may remain in recessive state externally.

18.4.4.5 MSCAN Initialization Mode

The MSCAN enters initialization mode when it is enabled (CANE=1).

When entering initialization mode during operation, any on-going transmission or reception is immediately aborted and synchronization to the CAN bus is lost, potentially causing CAN protocol violations. To protect the CAN bus system from fatal consequences of violations, the MSCAN immediately drives TXCAN into a recessive state.

NOTE

The user is responsible for ensuring that the MSCAN is not active when initialization mode is entered. The recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before setting the INITRQ bit in the CANCTL0 register. Otherwise, the abort of an on-going message can cause an error condition and can impact other CAN bus devices.

In initialization mode, the MSCAN is stopped. However, interface registers remain accessible. This mode is used to reset the CANCTL0, CANRFLG, CANRIER, CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL registers to their default values. In addition, the MSCAN enables the configuration of the CANBTR0, CANBTR1 bit timing registers; CANIDAC; and the CANIDAR, CANIDMR message filters. See Section 18.3.2.1, “MSCAN Control Register 0 (CANCTL0),” for a detailed description of the initialization mode.

20.6.2.2 ACMP Control Register 1 (ACMPC1)

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

	7	6	5	4	3	2	1	0
R	0	0	ACPSEL1-0		0	0	ACNSEL1-0	
W								
Reset	0	0	0	0	0	0	0	0

Figure 20-3. ACMP Control Register (ACMPC1)

[1] Read: Anytime
Write: Only if ACE=0

Table 20-4. ACMPC1 Routing Register Field Descriptions

Field	Description
5-4 ACPSEL1-0	**ACMP Positive Input Select —** These bits select the ACMP non-inverting input connected to ACMPP. 11 acmpi_1 10 acmpi_0 01 ACMP_1 00 ACMP_0
1-0 ACNSEL1-0	**ACMP Negative Input Select —** These bits select the ACMP inverting input connected to ACMPN. 11 acmpi_1 10 acmpi_0 01 ACMP_1 00 ACMP_0

20.6.2.3 ACMP Control Register 2 (ACMPC2)

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

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

Figure 20-4. ACMP Control Register (ACMPC2)

[1] Read: Anytime
Write: Anytime

Table 22-53. Verify Backdoor Access Key Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x0C	Not required
FCCOB1	Key 0	
FCCOB2	Key 1	
FCCOB3	Key 2	
FCCOB4	Key 3	

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0xFF_FE00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 22-54. Verify Backdoor Access Key Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see <st-blue>Section 22.3.2.2 Flash Security Register (FSEC))
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

22.4.7.12 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or EEPROM block.

Table 22-55. Set User Margin Level Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x0D	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] to identify Flash block	
FCCOB2	Margin level setting.	

Table 22-65. Erase EEPROM Sector Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x12	Global address [23:16] to identify EEPROM block
FCCOB1	Global address [15:0] anywhere within the sector to be erased. See <st-blue>Section 22.1.2.2 EEPROM Features for EEPROM sector size.	

Upon clearing CCIF to launch the Erase EEPROM Sector command, the Memory Controller will erase the selected Flash sector and verify that it is erased. The CCIF flag will set after the Erase EEPROM Sector operation has completed.

Table 22-66. Erase EEPROM Sector Command Error Handling

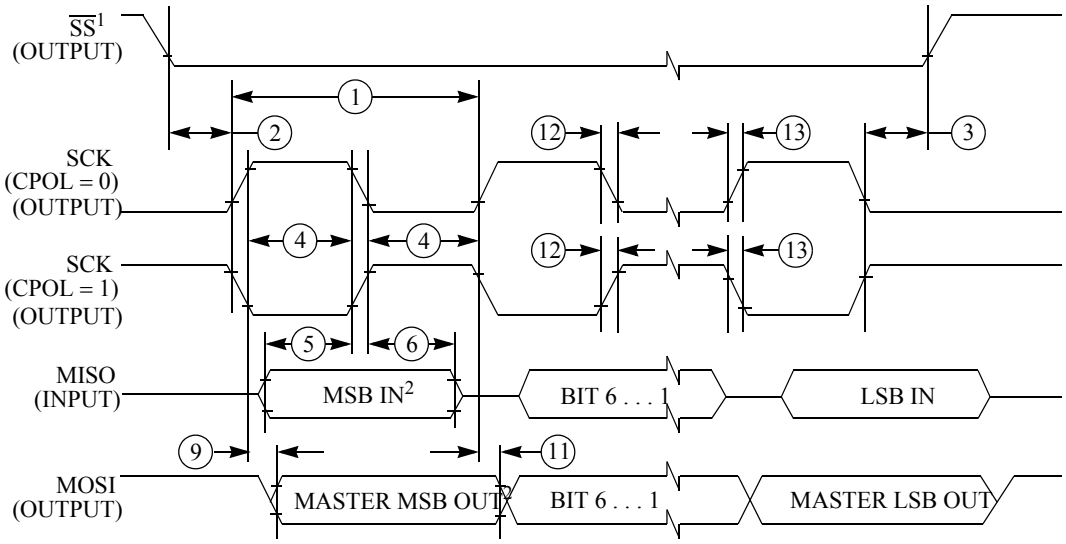
Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 22-28)
		Set if an invalid global address [23:0] is suppliedsee Table 22-2
		Set if a misaligned word address is supplied (global address [0] != 0)
	FPVIOL	Set if the selected area of the EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

22.4.7.17 Protection Override Command

The Protection Override command allows the user to temporarily override the protection limits, either decreasing, increasing or disabling protection limits, on P-Flash and/or EEPROM, if the comparison key provided as a parameter loaded on FCCOB matches the value of the key previously programmed on the Flash Configuration Field (see Table 22-3.). The value of the Protection Override Comparison Key must not be 16'hFFFF, that is considered invalid and if used as argument will cause the Protection Override feature to be disabled. Any valid key value that does not match the value programmed in the Flash Configuration Field will cause the Protection Override feature to be disabled. Current status of the Protection Override feature can be observed on FPSTAT FPOVRD bit (see Section 22.3.2.4, "Flash Protection Status Register (FPSTAT)).

Table 22-67. Protection Override Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x13	Protection Update Selection [1:0] See Table 22-68.
FCCOB1	Comparison Key	
FCCOB2	reserved	New FPROT value
FCCOB3	reserved	New DFPROT value



1. If enabled.
2. LSBFE = 0. For LSBFE = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure D-3. SPI Master Timing (CPHA=1)

In **Table D-1.** the timing characteristics for master mode are listed.

Table D-1. SPI Master Mode Timing Characteristics (Junction Temperature From –40°C To +175°C)

Num	Characteristic	Symbol				Unit
			Min	Typ	Max	
1	SCK Frequency	f_{sck}	1/2048	—	1/2	f_{bus}
1	SCK Period	t_{sck}	2	—	2048	t_{bus}
2	Enable Lead Time	t_{lead}	—	1/2	—	t_{sck}
3	Enable Lag Time	t_{lag}	—	1/2	—	t_{sck}
4	Clock (SCK) High or Low Time	t_{wsck}	—	1/2	—	t_{sck}
5	Data Setup Time (Inputs)	t_{su}	8	—	—	ns
6	Data Hold Time (Inputs)	t_{hi}	8	—	—	ns
9	Data Valid after SCK Edge	t_{vsck}	—	—	15	ns
10	Data Valid after $\overline{SS}$ fall (CPHA=0)	t_{vss}	—	—	15	ns
11	Data Hold Time (Outputs)	t_{ho}	0	—	—	ns
12	Rise and Fall Time Inputs	t_{rfi}	—	—	8	ns
13	Rise and Fall Time Outputs	t_{rfo}	—	—	8	ns

D.0.2 Slave Mode

In **Figure D-4.** the timing diagram for slave mode with transmission format CPHA=0 is depicted.

N.14 0x0690-0x0697 ACMP0

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0690	ACMPC0	R	ACE	ACOPE	ACOPS	ACDLY	ACHYS1-0		ACMOD1-0	
		W								
0x0691	ACMPC1	R	0	0	ACPSEL1-0		0	0	ACNSEL1-0	
		W								
0x0692	ACMPC2	R	0	0	0	0	0	0	0	ACIE
		W								
0x0693	ACMPS	R	ACO	0	0	0	0	0	0	ACIF
		W								
0x0694–0x0697	Reserved	R	0	0	0	0	0	0	0	0
		W								

N.15 0x0698-0x069F ACMP1

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0698	ACMPC0	R	ACE	ACOPE	ACOPS	ACDLY	ACHYS1-0		ACMOD1-0	
		W								
0x0699	ACMPC1	R	0	0	ACPSEL1-0		0	0	ACNSEL1-0	
		W								
0x069A	ACMPC2	R	0	0	0	0	0	0	0	ACIE
		W								
0x069B	ACMPS	R	ACO	0	0	0	0	0	0	ACIF
		W								
0x069C–0x069F	Reserved	R	0	0	0	0	0	0	0	0
		W								