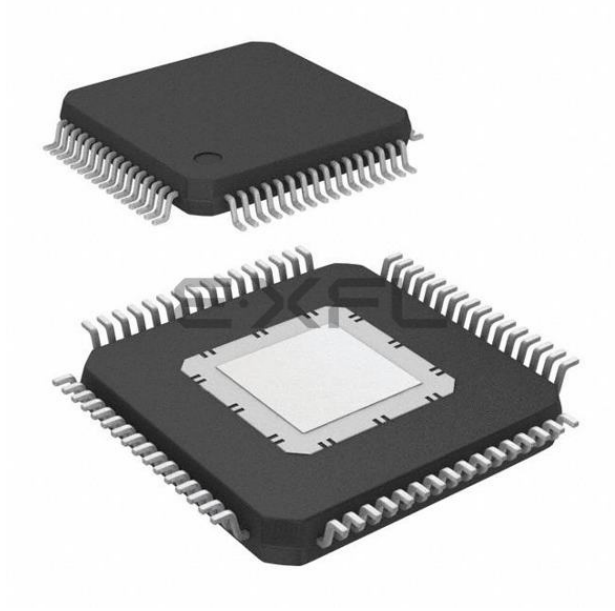


Welcome to E-XFL.COM

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	Obsolete
Core Processor	S12Z
Core Size	16-Bit
Speed	50MHz
Connectivity	LINbus, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 9x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP Exposed Pad
Supplier Device Package	64-HLQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvml12f1vkhr

Chapter 3
Memory Mapping Control (S12ZMMCV1)

Chapter 4
Interrupt (S12ZINTV0)

Chapter 11
Timer Module (TIM16B4CV3) Block Description

Chapter 12
Timer Module (TIM16B2CV3) Block Description

Chapter 13
Scalable Controller Area Network (S12MSCANV3)

Table 5-12. Consecutive READ_SAME Accesses With Variable Size

Row	Command	Base Address	00	01	10	11
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

5.4.6 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 5-6 and that of target-to-host in Figure 5-7 and Figure 5-8. 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 5-6. 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

- 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
- State sequencer control
  - State transitions forced by comparator matches
  - State transitions forced by software write to TRIG
  - State transitions forced by an external event
- The following types of breakpoints
  - CPU breakpoint entering active BDM on breakpoint (BDM)
  - CPU breakpoint executing SWI on breakpoint (SWI)
- Trace control
  - Tracing session triggered by state sequencer
  - Begin, End, and Mid alignment of tracing to trigger
- Four trace modes
  - Normal: change of flow (COF) PC information is stored (see Section 6.4.5.2.1) for change of flow definition.
  - Loop1: same as Normal but inhibits consecutive duplicate source address entries
  - Detail: address and data for all read/write access cycles are stored
  - Pure PC: All program counter addresses are stored.
- 2 Pin (data and clock) profiling interface
  - Output of code flow information

6.1.4 Modes of Operation

The DBG module can be used in all MCU functional modes.

The DBG module can issue breakpoint requests to force the device to enter active BDM or an SWI ISR. The BDC BACKGROUND command is also handled by the DBG to force the device to enter active BDM. When the device enters active BDM through a BACKGROUND command with the DBG module armed, the DBG remains armed.

- When a reset occurs the debugger pulls BKGD low until the reset ends, forcing SSC mode entry.
- Then the debugger reads the reset flags to determine the cause of reset.
- If required, the debugger can read the trace buffer to see what happened just before reset. Since the trace buffer and DBGCNT register are not affected by resets other than POR.
- The debugger configures and arms the DBG to start tracing on returning to application code.
- The debugger then sets the PC according to the reset flags.
- Then the debugger returns to user code with GO or STEP1.

6.5.3 Breakpoints from other S12Z sources

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

6.5.4 Code Profiling

The code profiling data output pin PDO is mapped to a device pin that can also be used as GPIO in an application. If profiling is required and all pins are required in the application, it is recommended to use the device pin for a simple output function in the application, without feedback to the chip. In this way the application can still be profiled, since the pin has no effect on code flow.

The PDO provides a simple bit stream that must be strobed at both edges of the profiling clock when profiling. The external development tool activates profiling by setting the DBG ARM bit, with PROFILE and PDOE already set. Thereafter the first bit of the profiling bit stream is valid at the first rising edge of the profiling clock. No start bit is provided. The external development tool must detect this first rising edge after arming the DBG. To detect the end of profiling, the DBG ARM bit can be monitored using the BDC.

Table 8-7. CPMUCLKS Descriptions

Field	Description
7 PLLSEL	PLL Select Bit This bit selects the PLLCLK as source of the System Clocks (Core Clock and Bus Clock). PLLSEL can only be set to 0, if UPOSC=1. UPOSC= 0 sets the PLLSEL bit. Entering Full Stop Mode sets the PLLSEL bit. 0 System clocks are derived from OSCCLK if oscillator is up (UPOSC=1, $f_{bus} = f_{osc} / 2$). 1 System clocks are derived from PLLCLK, $f_{bus} = f_{PLL} / 2$.
6 PSTP	Pseudo Stop Bit This bit controls the functionality of the oscillator during Stop Mode. 0 Oscillator is disabled in Stop Mode (Full Stop Mode). 1 Oscillator continues to run in Stop Mode (Pseudo Stop Mode), option to run RTI and COP. **Note:** Pseudo Stop Mode allows for faster STOP recovery and reduces the mechanical stress and aging of the resonator in case of frequent STOP conditions at the expense of a slightly increased power consumption. **Note:** When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator t_{UPOSC} before entering Pseudo Stop Mode.
5 CSAD	COP in Stop Mode ACLK Disable — If this bit is set the ACLK for the COP in Stop Mode is disabled. Hence the COP is static while in Stop Mode and continues to operate after exit from Stop Mode. For CSAD = 1 and COP is running on ACLK (COPOSCSEL1 = 1) the following applies: Due to clock domain crossing synchronization there is a latency time of 2 ACLK cycles to enter Stop Mode. After exit from STOP mode (when interrupt service routine is entered) the software has to wait for 2 ACLK cycles before it is allowed to enter Stop mode again (STOP instruction). It is absolutely forbidden to enter Stop Mode before this time of 2 ACLK cycles has elapsed. 0 COP running in Stop Mode (ACLK for COP enabled in Stop Mode). 1 COP stopped in Stop Mode (ACLK for COP disabled in Stop Mode)
4 COP OSCSEL1	**COP Clock Select 1** — COPOSCSEL0 and COPOSCSEL1 combined determine the clock source to the COP (see also Table 8-8). If COPOSCSEL1 = 1, COPOSCSEL0 has no effect regarding clock select and changing the COPOSCSEL0 bit does not re-start the COP time-out period. COPOSCSEL1 selects the clock source to the COP to be either ACLK (derived from trimmable internal RC-Oscillator) or clock selected via COPOSCSEL0 (IRCCLK or OSCCLK). Changing the COPOSCSEL1 bit re-starts the COP time-out period. COPOSCSEL1 can be set independent from value of UPOSC. UPOSC= 0 does not clear the COPOSCSEL1 bit. 0 COP clock source defined by COPOSCSEL0 1 COP clock source is ACLK derived from a trimmable internal RC-Oscillator
3 PRE	**RTI Enable During Pseudo Stop Bit** — PRE enables the RTI during Pseudo Stop Mode. 0 RTI stops running during Pseudo Stop Mode. 1 RTI continues running during Pseudo Stop Mode if RTIOSCSEL=1. **Note:** If PRE=0 or RTIOSCSEL=0 then the RTI will go static while Stop Mode is active. The RTI counter will not be reset.
2 PCE	**COP Enable During Pseudo Stop Bit** — PCE enables the COP during Pseudo Stop Mode. 0 COP stops running during Pseudo Stop Mode 1 COP continues running during Pseudo Stop Mode if COPOSCSEL=1 **Note:** If PCE=0 or COPOSCSEL=0 then the COP will go static while Stop Mode is active. The COP counter will not be reset.

```
/* put your code to loop and wait for the LOCKIF or */
/* poll CPMUIFLG register until both LOCK status is “1” */
/* that is CPMUIFLG == 0x18 */

/*..............continue to your main code execution here..............*/
```

9.5.2.5 ADC Format Register (ADCFMT)

Module Base + 0x0004

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

= Unimplemented or Reserved

Figure 9-8. ADC Format Register (ADCFMT)

Read: Anytime

Write: Bits DJM and SRES[2:0] are writable if bit ADC_EN clear or bit SMOD_ACC set

Table 9-8. ADCFMT Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — Conversion result data format is always unsigned. This bit controls justification of conversion result data in the conversion result list. 0 Left justified data in the conversion result list. 1 Right justified data in the conversion result list.
2-0 SRES[2:0]	**ADC Resolution Select** — These bits select the resolution of conversion results. See Table 9-9 for coding.

Table 9-9. Selectable Conversion Resolution

SRES[2]	SRES[1]	SRES[0]	ADC Resolution
0	0	0	8-bit data
0	0	1	Reserved[1.]
0	1	0	10-bit data
0	1	1	Reserved[1.]
1	0	0	12-bit data
1	x	x	Reserved[(1)]

1. Reserved settings cause a severe error at ADC conversion start whereby the CMD_EIF flag is set and ADC ceases operation

Table 9-24. 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 (V1, V2, see Table 9-2) VRL_0 (V3, see Table 9-2)
0	0	0	0	0	1	VRH_0/1 (V1, V2, see Table 9-2) VRH_0/1/2 (V3, see Table 9-2)
0	0	0	0	1	0	(VRH_0/1 + VRL_0/1) / 2 (V1, V2, see Table 9-2) (VRH_0/1/2 + VRL_0) / 2 (V3, see Table 9-2)
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 (ADC temperature sense)
0	0	1	0	0	1	Internal_1
0	0	1	0	1	0	Internal_2
0	0	1	0	1	1	Internal_3
0	0	1	1	0	0	Internal_4
0	0	1	1	0	1	Internal_5
0	0	1	1	1	0	Internal_6
0	0	1	1	1	1	Internal_7
0	1	0	0	0	0	AN0
0	1	0	0	0	1	AN1
0	1	0	0	1	0	AN2
0	1	0	0	1	1	AN3
0	1	0	1	0	0	AN4
0	1	x	x	x	x	ANx
1	x	x	x	x	x	Reserved

NOTE

ANx in Table 9-24 is the maximum number of implemented analog input channels on the device. Please refer to the device overview of the reference manual for details regarding number of analog input channels.

9.6.3.2.1 Introduction of The Command Sequence List (CSL) Format

A Command Sequence List (CSL) contains up to 64 conversion commands. A user selectable number of successive conversion commands in the CSL can be grouped as a command sequence. This sequence of conversion commands is successively executed by the ADC at the occurrence of a Trigger Event. The commands of a sequence are successively executed until an “End Of Sequence” or “End Of List” command type identifier in a command is detected (command type is coded via bits CMD_SEL[1:0]). The number of successive conversion commands that belong to a command sequence and the number of command sequences inside the CSL can be freely defined by the user and is limited by the 64 conversion commands a CSL can contain. A CSL must contain at least one conversion command and one “end of list” command type identifier. The minimum number of command sequences inside a CSL is zero and the maximum number of command sequences is 63. A command sequence is defined with bits CMD_SEL[1:0] in the register ADCCMD_M by defining the end of a conversion sequence. The Figure 9-29 and Figure 9-30 provides examples of a CSL.

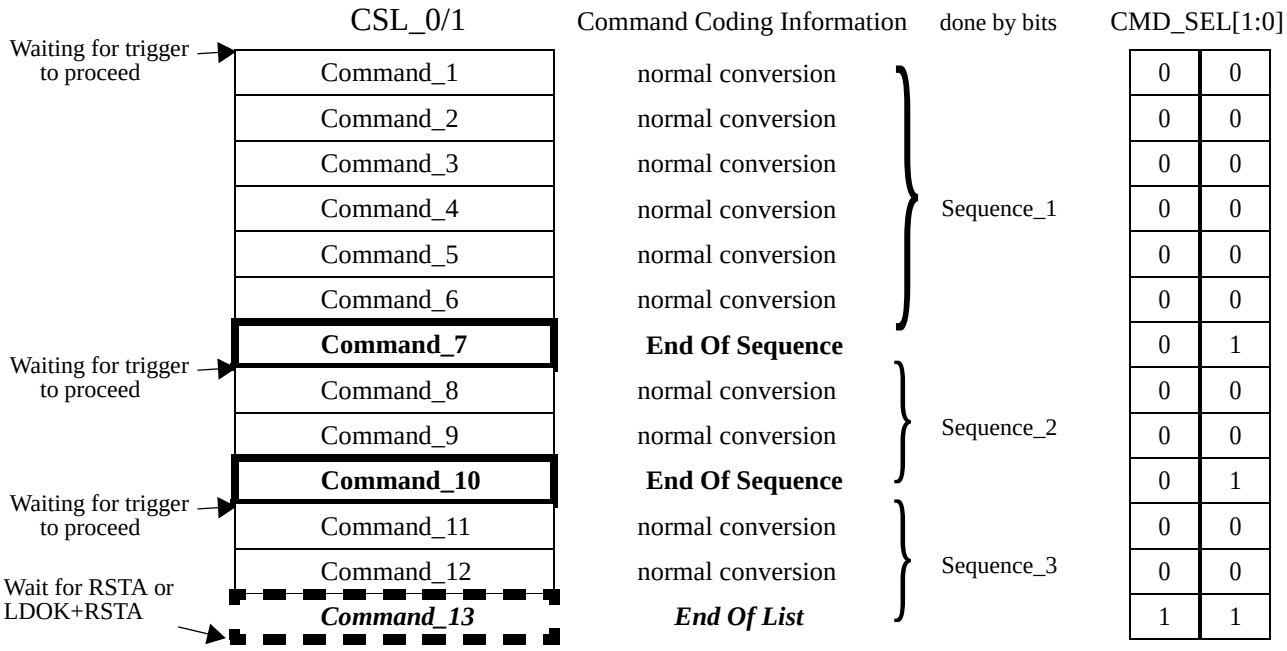


Figure 9-29. Example CSL with sequences and an “End Of List” command type identifier

Table 13-23. 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 13-23. MSCAN Identifier Mask Registers (Second Bank) — CANIDMR4–CANIDMR7

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

Table 13-24. CANIDMR4–CANIDMR7 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

13.3.3 Programmer’s Model of Message Storage

The following section details the organization of the receive and transmit message buffers and the associated control registers.

To simplify the programmer interface, the receive and transmit message buffers have the same outline. Each message buffer allocates 16 bytes in the memory map containing a 13 byte data structure.

An additional transmit buffer priority register (TBPR) is defined for the transmit buffers. Within the last two bytes of this memory map, the MSCAN stores a special 16-bit time stamp, which is sampled from an internal timer after successful transmission or reception of a message. This feature is only available for transmit and receiver buffers, if the TIME bit is set (see Section 13.3.2.1, “MSCAN Control Register 0 (CANCTL0)”).

The time stamp register is written by the MSCAN. The CPU can only read these registers.

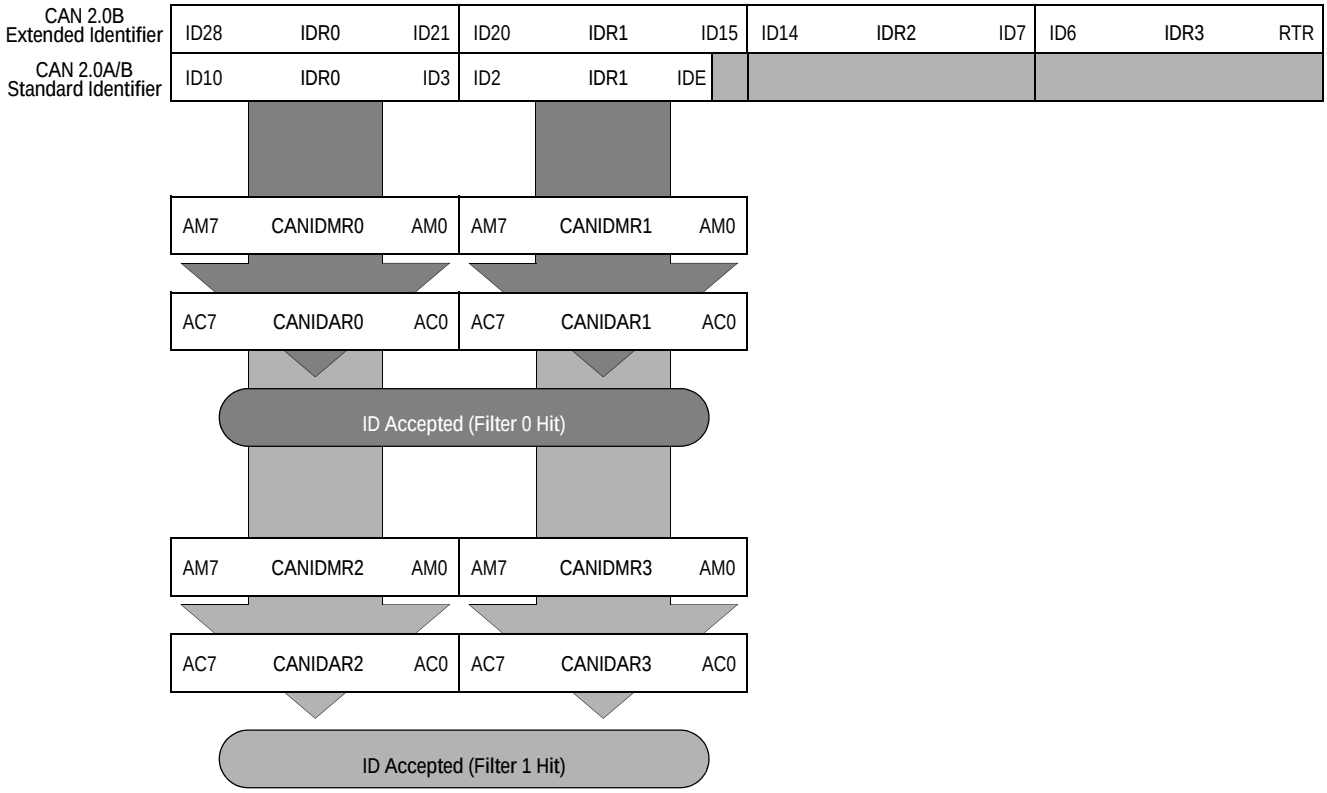


Figure 13-41. 16-bit Maskable Identifier Acceptance Filters

13.4.5.7 Disabled Mode

The MSCAN is in disabled mode out of reset (CANE=0). All module clocks are stopped for power saving, however the register map can still be accessed as specified.

13.4.5.8 Programmable Wake-Up Function

The MSCAN can be programmed to wake up from sleep or power down mode as soon as CAN bus activity is detected (see control bit WUPE in MSCAN Control Register 0 (CANCTL0). The sensitivity to existing CAN bus action can be modified by applying a low-pass filter function to the RXCAN input line (see control bit WUPM in Section 13.3.2.2, “MSCAN Control Register 1 (CANCTL1)”).

This feature can be used to protect the MSCAN from wake-up due to short glitches on the CAN bus lines. Such glitches can result from—for example—electromagnetic interference within noisy environments.

13.4.6 Reset Initialization

The reset state of each individual bit is listed in Section 13.3.2, “Register Descriptions,” which details all the registers and their bit-fields.

13.4.7 Interrupts

This section describes all interrupts originated by the MSCAN. It documents the enable bits and generated flags. Each interrupt is listed and described separately.

13.4.7.1 Description of Interrupt Operation

The MSCAN supports four interrupt vectors (see Table 13-38), any of which can be individually masked (for details see Section 13.3.2.6, “MSCAN Receiver Interrupt Enable Register (CANRIER)” to Section 13.3.2.8, “MSCAN Transmitter Interrupt Enable Register (CANTIER)”).

Refer to the device overview section to determine the dedicated interrupt vector addresses.

Table 13-38. Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Wake-Up Interrupt (WUPIF)	I bit	CANRIER (WUPIE)
Error Interrupts Interrupt (CSCIF, OVRIF)	I bit	CANRIER (CSCIE, OVRIE)
Receive Interrupt (RXF)	I bit	CANRIER (RXFIE)
Transmit Interrupts (TXE[2:0])	I bit	CANTIER (TXEIE[2:0])

13.4.7.2 Transmit Interrupt

At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The TXEx flag of the empty message buffer is set.

The complementary channel operation is for driving top and bottom transistors in a motor drive circuit, such as the one in Figure 15-49.

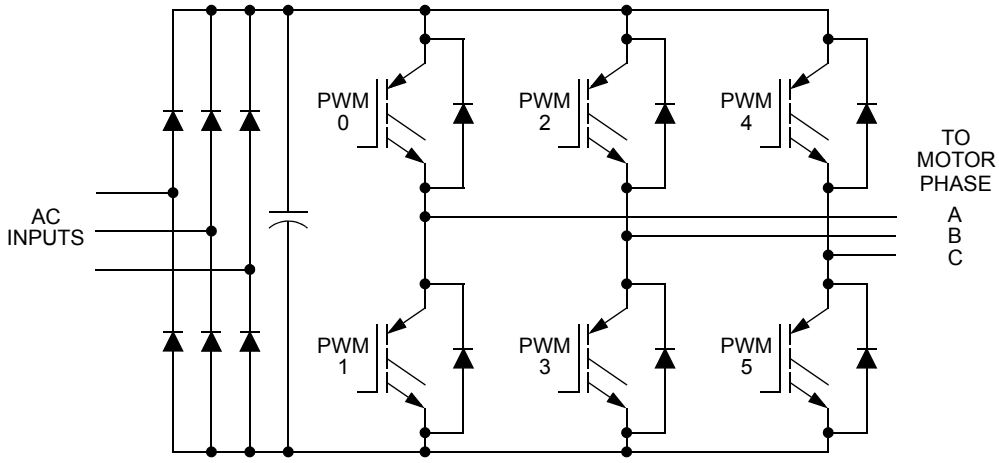


Figure 15-49. Typical 3-Phase AC Motor Drive

In complementary channel operation following additional features exist:

- Deadtime insertion
- Separate top and bottom pulse width correction via current status inputs or software
- Three variants of PWM output:
  - — Asymmetric in center-aligned mode
  - — Variable edge placement in edge-aligned mode
  - — Double switching in center-aligned mode

15.4.5 Deadtime Generators

While in complementary operation, each PWM pair can be used to drive top/bottom transistors, as shown in Figure 15-50. Ideally, the PWM pairs are an inversion of each other. When the top PWM channel is active, the bottom PWM channel is inactive, and vice versa.

NOTE

To avoid a short-circuit on the DC bus and endangering the transistor, there must be no overlap of conducting intervals between the top and bottom transistor. But the transistor's characteristics make its switching-off time longer than switching-on time. To avoid the conducting overlap of the top and bottom transistors, deadtime needs to be inserted in the switching period.

Deadtime generators automatically insert software-selectable activation delays into each pair of PWM outputs. The deadtime register (PMFDTMx) specifies the number of PWM clock cycles to use for deadtime delay. Every time the deadtime generator inputs changes state, deadtime is inserted. Deadtime forces both PWM outputs in the pair to the inactive state.

A method of correcting this, adding to or subtracting from the PWM value used, is discussed next.

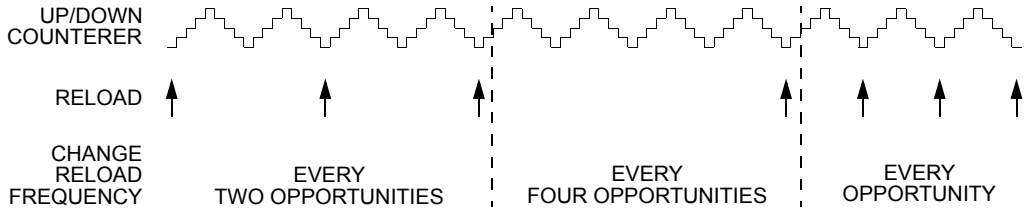


Figure 15-73. Full Cycle Reload Frequency Change

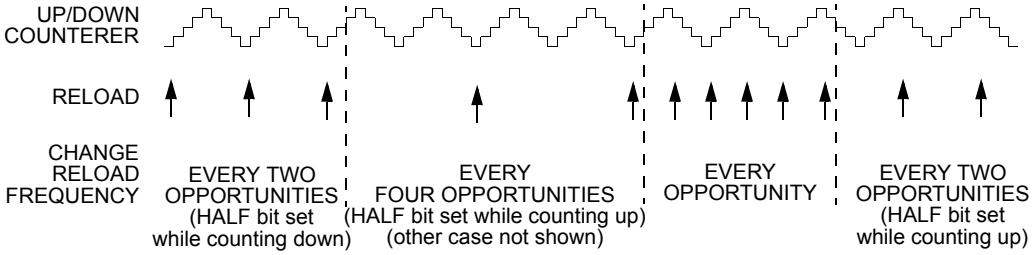


Figure 15-74. Half Cycle Reload Frequency Change

15.4.12.4 Reload Flag

The PWMRF reload flag is set at every reload opportunity, regardless of whether an actual reload occurs (as determined by the related LDOK bit or global load OK). If the PWM reload interrupt enable bit PWMRIE is set, the PWMRF flag generates CPU interrupt requests allowing software to calculate new PWM parameters in real time. When PWMRIE is not set, reloads still occur at the selected reload rate without generating CPU interrupt requests.

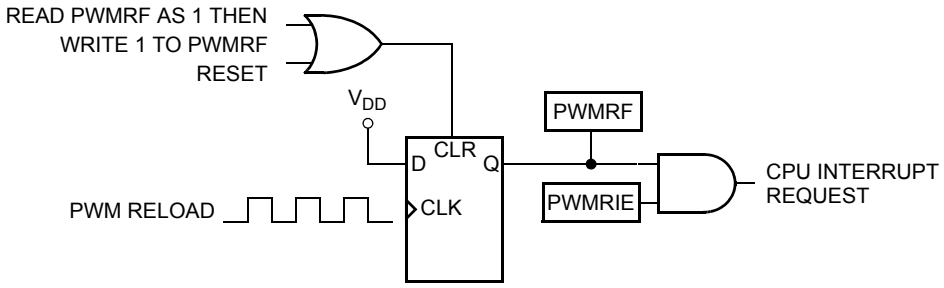


Figure 15-75. PWMRF Reload Interrupt Request

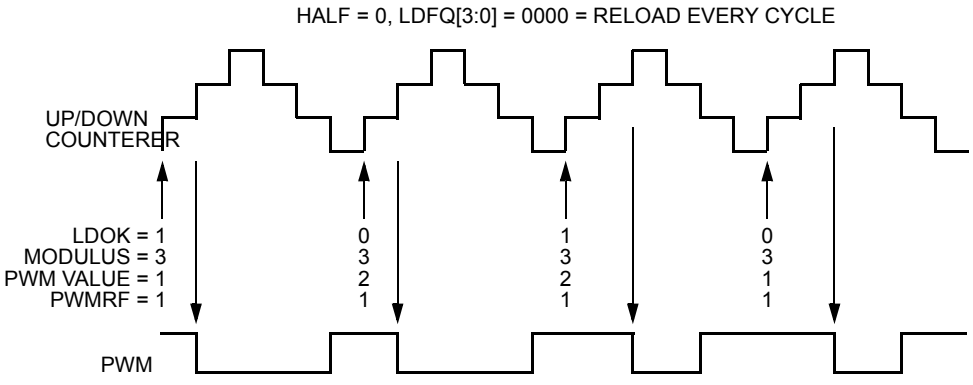


Figure 15-76. Full-Cycle Center-Aligned PWM Value Loading

Figure 18-25. Short to Supply Detection

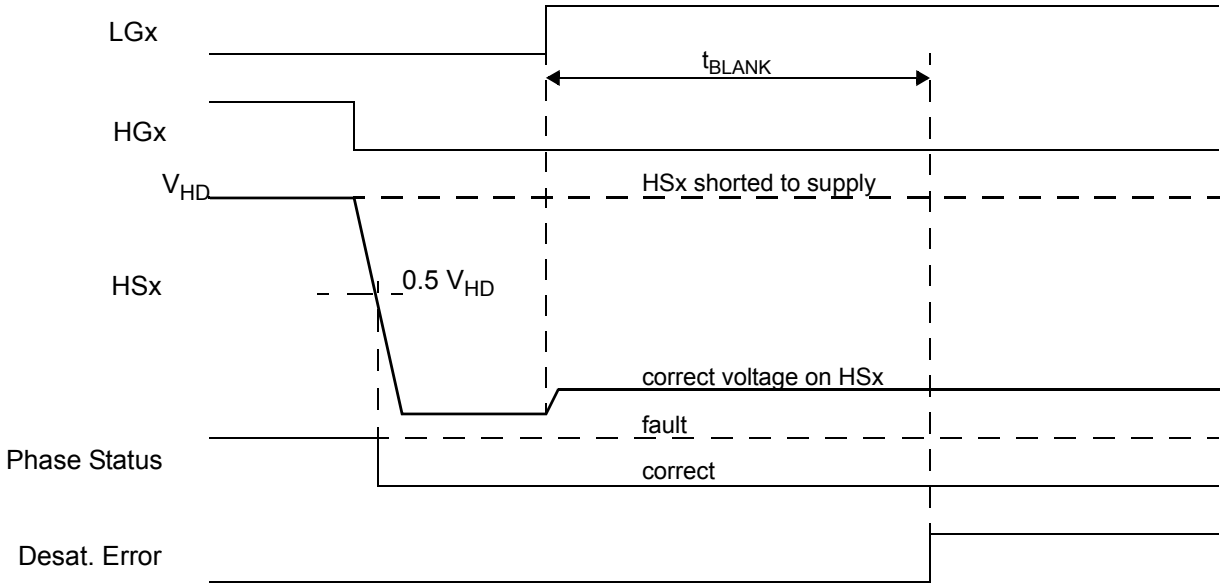
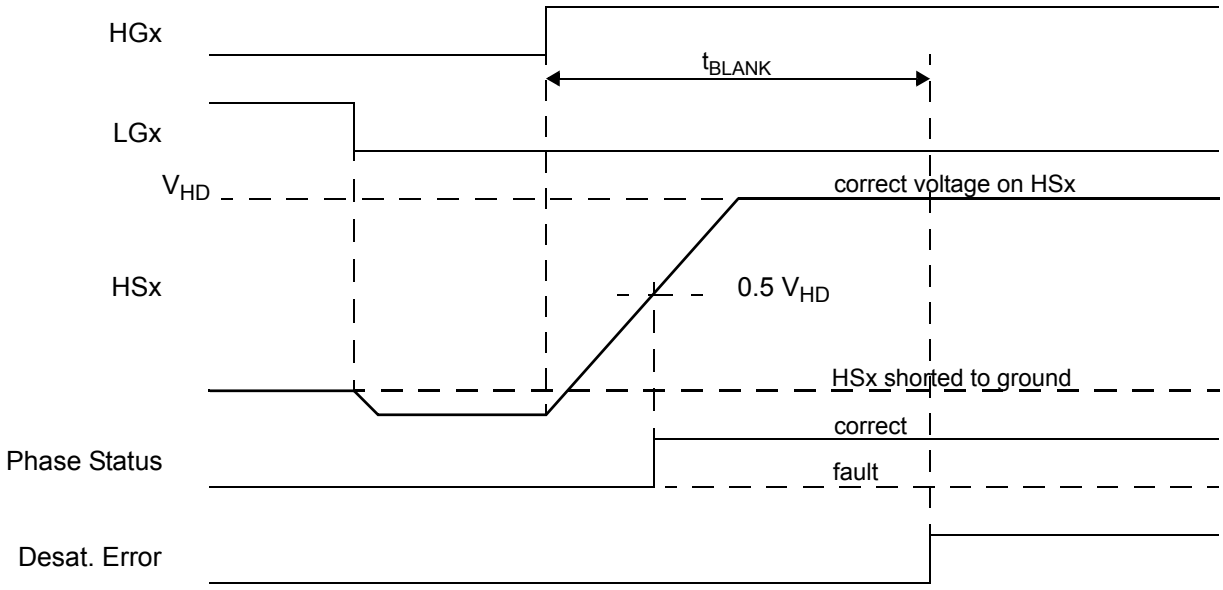
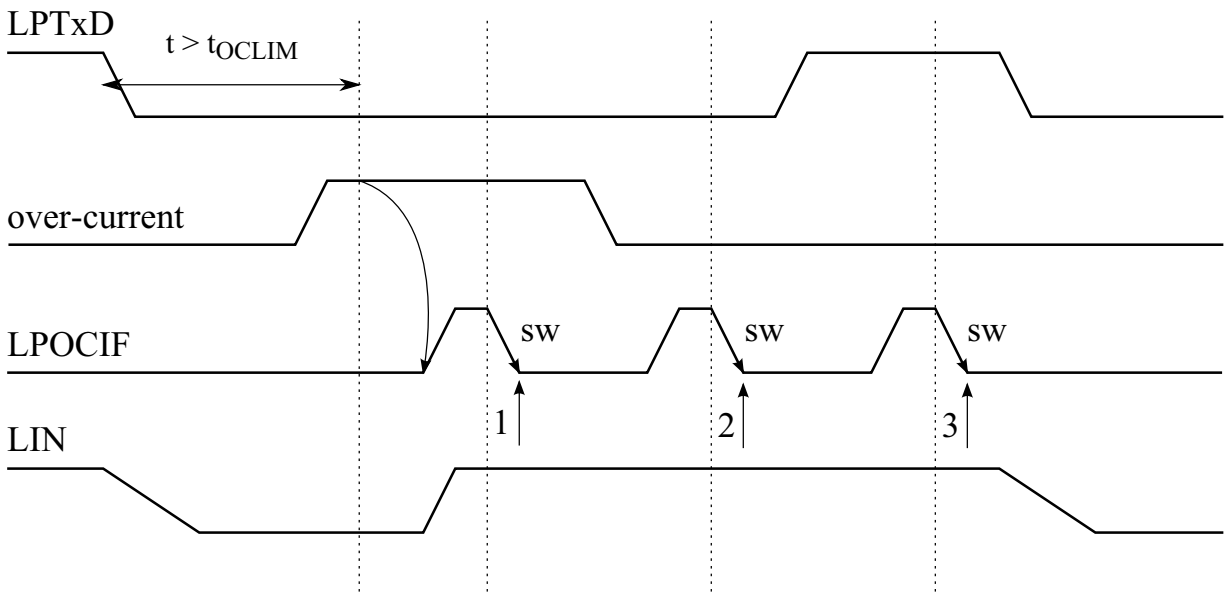


Figure 18-26. Short to Ground Detection





1: Flag cleared, transmitter re-enable not successful because over-current is still present
2: Flag cleared, transmitter re-enable not successful because LPTxD is dominant
3: Flag cleared, transmitter re-enable successful

Figure 19-12. Overcurrent interrupt handling

19.4.4.2 TxD-dominant timeout Interrupt

NOTE

In order to perform PWM communication, the TxD-dominant timeout feature must be disabled.

To protect the LIN bus from a network lock-up, the LIN Physical Layer implements a TxD-dominant timeout mechanism. When the LPTxD signal has been dominant for more than t_{DTLIM} the transmitter is disabled and the LPDT status flag and the LPDTIF interrupt flag are set.

In order to re-enable the transmitter again, the following prerequisites must be met:

1) TxD-dominant condition is over (LPDT=0)

2) LPTxD is recessive or the LIN Physical Layer is in shutdown or receive only mode for a minimum of a transmit bit time

To re-enable the transmitter then, the LPDTIF flag must be cleared (by writing a 1).

NOTE

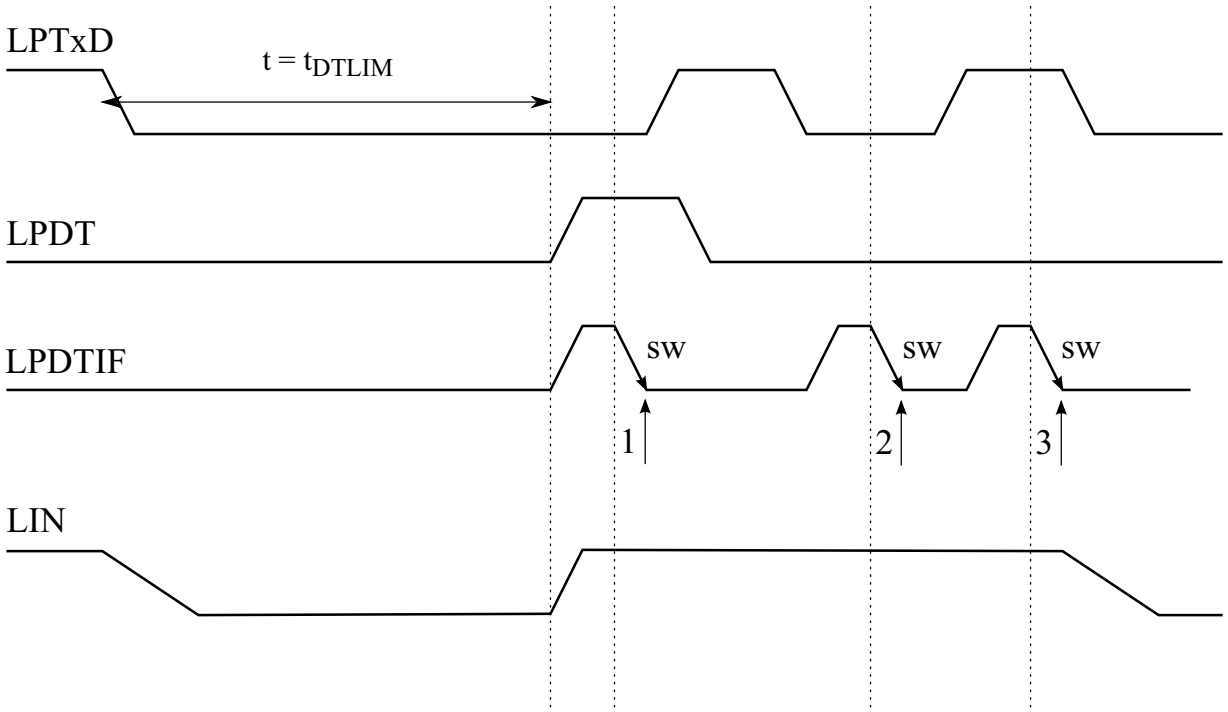
Please make sure that LPDTIF=1 before trying to clear it. It is not allowed to try to clear LPDTIF if LPDTIF=0 already.

After clearing LPDTIF, if the TxD-dominant timeout condition is still present or the LPTxD pin is dominant while being in normal mode, the transmitter remains disabled and the LPDTIF flag is set after a time again to indicate that the attempt to re-enable has failed. This time is equal to:

- minimum 1 IRC period (1 us) + 2 bus periods
- maximum 2 IRC periods (2 us) + 3 bus periods

If the bit LPDTIE is set in the LPIE register, an interrupt is requested.

Figure 19-13 shows the different scenarios of TxD-dominant timeout interrupt handling.



1: Flag cleared, transmitter re-enable not successful because TxD-dominant timeout condition is still present
2: Flag cleared, transmitter re-enable not successful because LPTxD is dominant
3: Flag cleared, transmitter re-enable successful

Figure 19-13. TxD-dominant timeout interrupt handling

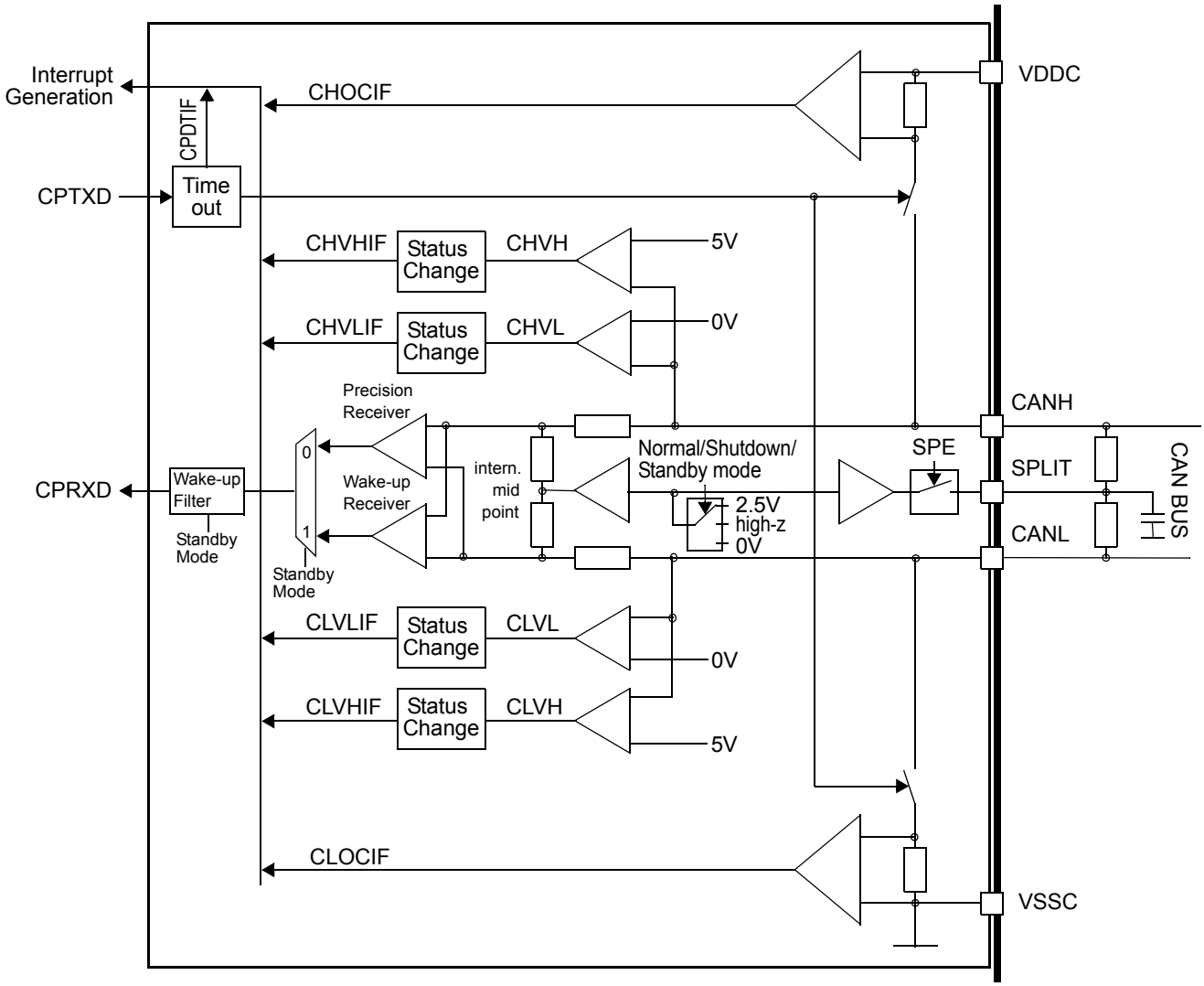


Figure 21-1. CAN Physical Layer Block Diagram

21.2 External Signal Description

Table 21-2 shows the external pins associated with the CAN Physical Layer.

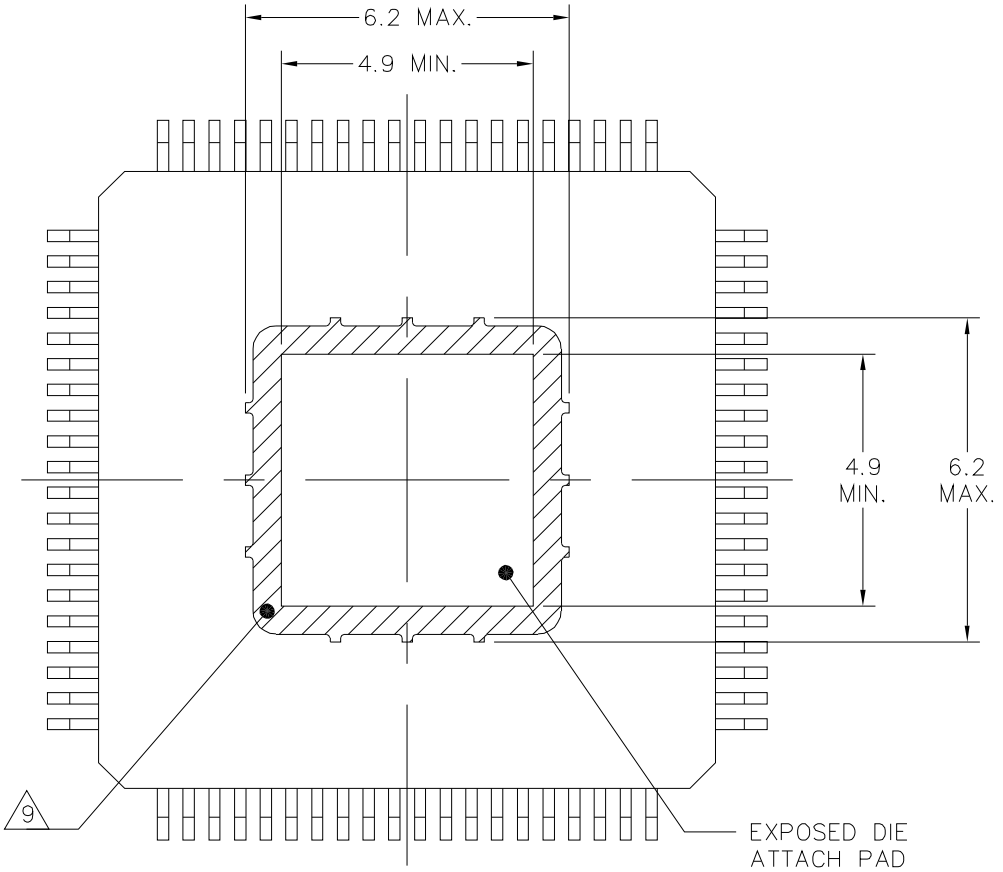
Table 21-2. CAN Physical Layer Signal Properties

Name	Function
CANH	CAN Bus High Pin
SPLIT	2.5 V Termination Pin
CANL	CAN Bus Low Pin
VDDC	Supply Pin for CAN Physical Layer
VSSC	Ground Pin for CAN Physical Layer

freescale

MECHANICAL OUTLINE

DO NOT SCALE THIS DRAWING



VIEW J–J

TITLE: LQFP, 12 X 12 X 1.4 PKG, 0.5 PITCH, 80LD, 5.6 X 5.6 EXPOSED PAD	DOCUMENT NO: 98ASA00505D REV: X2
	STANDARD: NON–JEDEC
	SHEET: 3