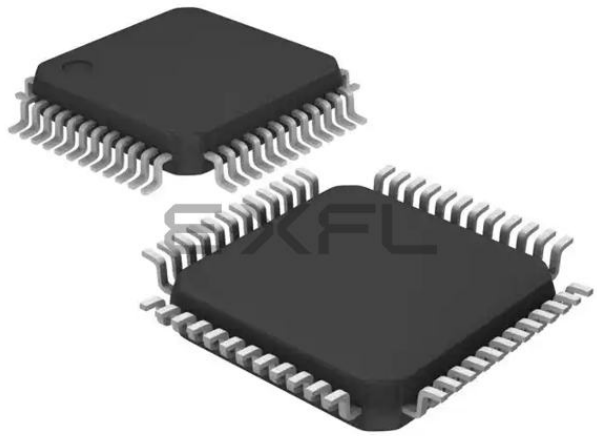

Welcome to E-XFL.COM

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 10x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°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/s912zvc64f0vlf

Chapter 11
Timer Module (TIM16B8CV3) Block Description

Chapter 12
Timer Module (TIM16B4CV3) Block Description

2. Select internal pullup device on HVI (PTPSL=1)
3. Enable function to force input buffer active on HVI in analog mode (PTTEL=1)
4. Verify PTIL=0 for a connected external pulldown device; read PTIL=1 for an open input

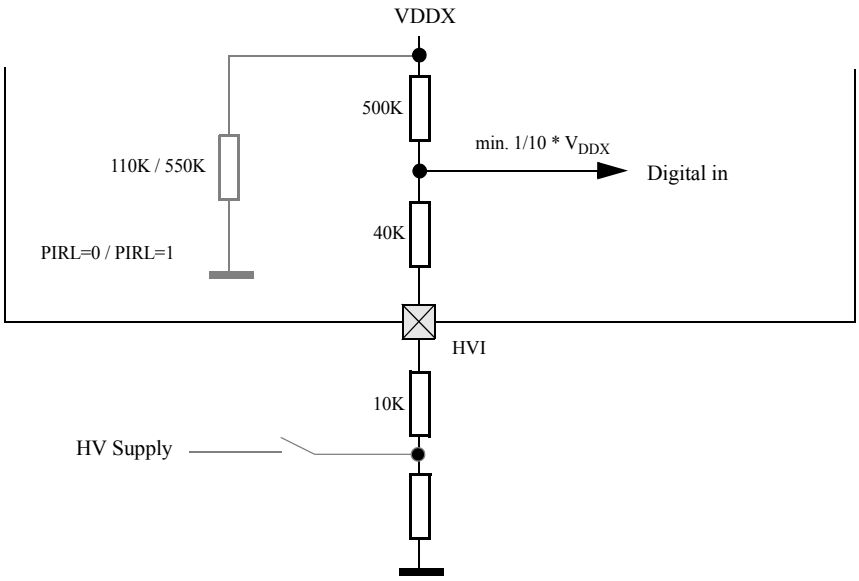

Figure 2-36. Digital Input Read with Pullup Enabled

External pullup device (Figure 2-37):

1. Enable analog function on HVI in non-direct mode (PTAENL=1, PTADIRL=0)
2. Select internal pulldown device on HVI (PTPSL=0)
3. Enable function to force input buffer active on HVI in analog mode (PTTEL=1)
4. Verify PTIL=1 for a connected external pullup device; read PTIL=0 for an open input

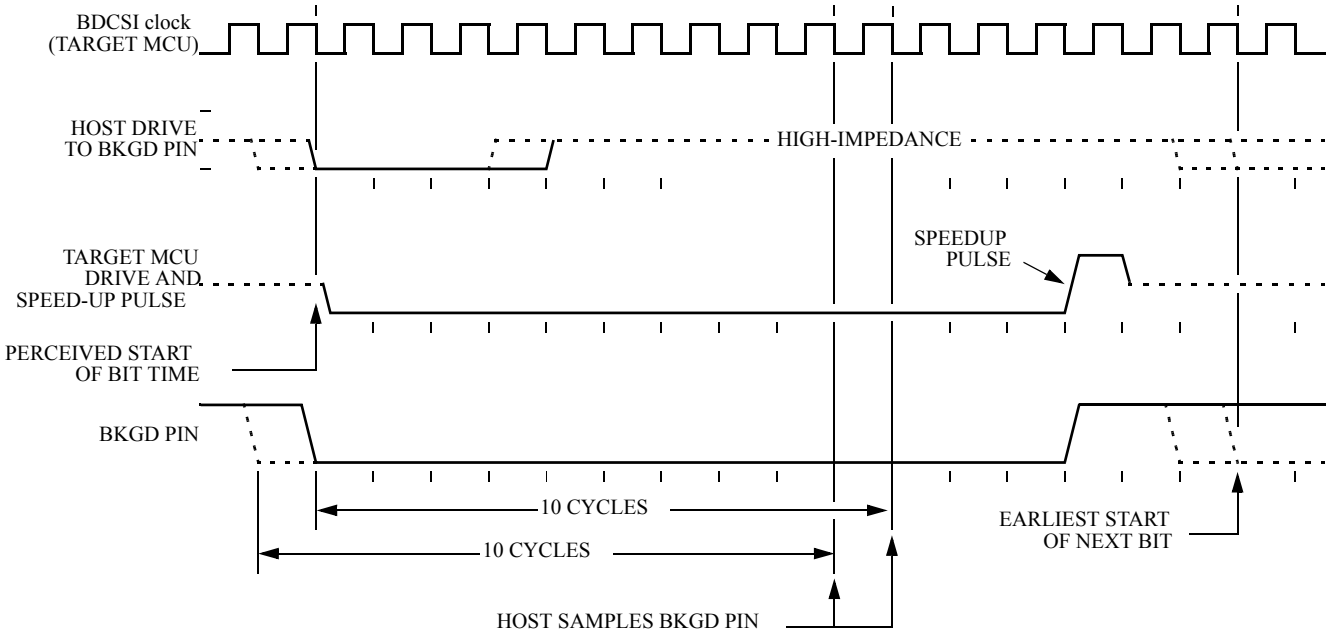

Figure 3-8. BDC Target-to-Host Serial Bit Timing (Logic 0)

3.4.7 Serial Interface Hardware Handshake (ACK Pulse) Protocol

BDC commands are processed internally at the device core clock rate. Since the BDCSI clock can be asynchronous relative to the bus frequency, a handshake protocol is provided so the host can determine when an issued command has been executed. This section describes the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when a BDC command has been executed by the target. This protocol is implemented by a low pulse (16 BDCSI clock cycles) followed by a brief speedup pulse on the BKGD pin, generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 3-9). This pulse is referred to as the ACK pulse. After the ACK pulse has finished, the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command.

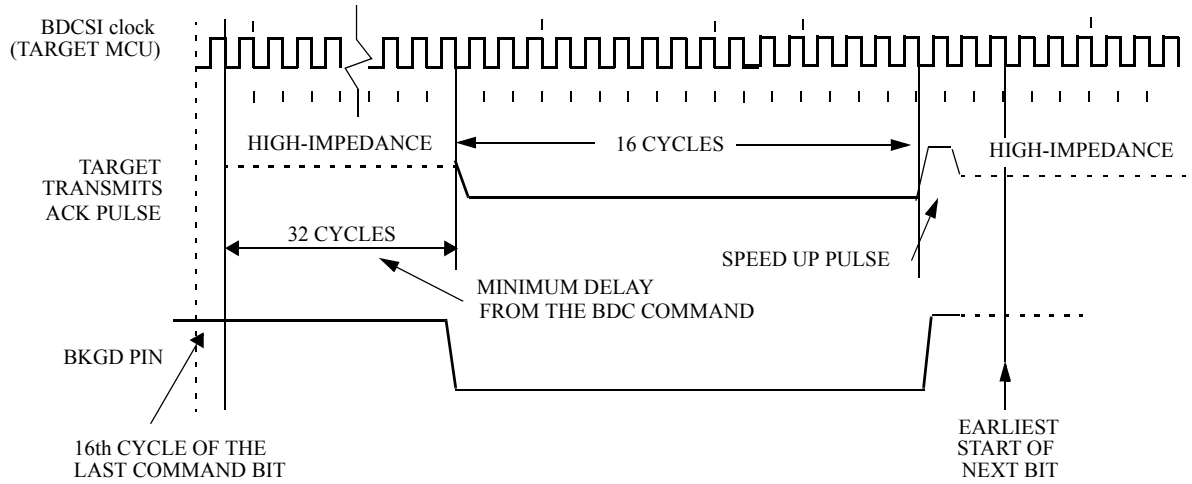

Figure 3-9. Target Acknowledge Pulse (ACK)

If the handshake protocol is disabled, the access is always independent of free cycles, whereby BDC has higher priority than CPU. Since at least 2 bytes (command byte + data byte) are transferred over BKGD the maximum intrusiveness is only once every few hundred cycles.

After decoding an internal access command, the BDC then awaits the next internal core clock cycle. The relationship between BDCSI clock and core clock must be considered. If the host retrieves the data immediately, then the BDCSI clock frequency must not be more than 4 times the core clock frequency, in order to guarantee that the BDC gains bus access within 16 the BDCSI cycle DLY period following an access command. If the BDCSI clock frequency is more than 4 times the core clock frequency, then the host must use a suitable delay time before retrieving data (see 3.5.1/3-150). Furthermore, for stretched read accesses to external resources via a device expanded bus (if implemented) the potential extra stretch cycles must be taken into consideration before attempting to obtain read data.

If the access does not succeed before the host starts data retrieval then the NORESP flag is set but the access is not aborted. The NORESP state can be used by the host to recognize an unexpected access conflict due to stretched expanded bus accesses. Although the NORESP bit is set when an access does not succeed before the start of data retrieval, the access may succeed in following bus cycles if the internal access has already been initiated.

3.4.10 Single Stepping

When a STEP1 command is issued to the BDC in active BDM, the CPU executes a single instruction in the user code and returns to active BDM. The STEP1 command can be issued repeatedly to step through the user code one instruction at a time.

If an interrupt is pending when a STEP1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. In this case the stacking counts as one instruction. The device re-enters active BDM with the program counter pointing to the first instruction in the interrupt service routine.

When stepping through the user code, the execution of the user code is done step by step but peripherals are free running. Some peripheral modules include a freeze feature, whereby their clocks are halted when the device enters active BDM. Timer modules typically include the freeze feature. Serial interface modules typically do not include the freeze feature. Hence possible timing relations between CPU code execution and occurrence of events of peripherals no longer exist.

If the handshake protocol is enabled and BDCCIS is set then stepping over the STOP instruction causes the Long-ACK pulse to be generated and the BDCCSR STOP flag to be set. When stop mode is exited due to an interrupt the device enters active BDM and the PC points to the start of the corresponding interrupt service routine. Stepping can be continued.

Stepping over a WAI instruction, the STEP1 command cannot be finished because active BDM cannot be entered after CPU starts to execute the WAI instruction.

Stepping over the WAI instruction causes the BDCCSR WAIT and NORESP flags to be set and, if the handshake protocol is enabled, then the Long-ACK pulse is generated. Then the device enters wait mode, clears the BDMACT bit and awaits an interrupt to leave wait mode. In this time non-intrusive BDC commands are possible, although the STEP1 has actually not finished. When an interrupt occurs the device leaves wait mode, enters active BDM and the PC points to the start of the corresponding interrupt service routine. A further ACK related to stepping over the WAI is not generated.

8.2 Signal Description

This section lists and describes the signals that connect off chip as well as internal supply nodes and special signals.

8.2.1 $\overline{\text{RESET}}$

Pin $\overline{\text{RESET}}$ is an active-low bidirectional pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates that an MCU-internal reset has been triggered.

8.2.2 EXTAL and XTAL

These pins provide the interface for a crystal to control the internal clock generator circuitry. EXTAL is the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. If XOSCLCP is enabled, the MCU internal OSCCLK_LCP is derived from the EXTAL input frequency. If OSCE=0, the EXTAL pin is pulled down by an internal resistor of approximately 200 kΩ and the XTAL pin is pulled down by an internal resistor of approximately 700 kΩ.

NOTE

NXP recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.
The loop controlled circuit (XOSCLCP) is not suited for overtone resonators and crystals.

8.2.3 VSUP — Regulator Power Input Pin

Pin VSUP is the power input of VREGAUTO. All currents sourced into the regulator loads flow through this pin.

A suitable reverse battery protection network can be used to connect VSUP to the car battery supply network.

8.2.4 VDDA, VSSA — Regulator Reference Supply Pins

Pins VDDA and VSSA are used to supply the analog parts of the regulator. Internal precision reference circuits are supplied from these signals.

An off-chip decoupling capacitor (220 nF(X7R ceramic)) between VDDA and VSSA is required and can improve the quality of this supply.

VDDA has to be connected externally to VDDX.

8.2.5 VDDX, VSSX — Pad Supply Pins

VDDX is the supply domain for the digital Pads. VDDX has to be connected externally to VDDA.

An off-chip decoupling capacitor (10µF plus 220 nF(X7R ceramic)) between VDDX and VSSX is required.

This supply domain is monitored by the Low Voltage Reset circuit.

8.2.6 .VDDC, VSSC — CANPHY Supply Pin

VDDC is the supply domain for the CANPHY.

An off-chip decoupling capacitor (10µF plus 220 nF(X7R ceramic)) between VDDC and VSSC is required.

This supply domain is monitored by the Low Voltage Reset circuit.

8.2.7 BCTL — Base Control Pin for external PNP

BCTL is the ballast connection for the on chip voltage regulator. It provides the base current of an external

BJT (PNP) of the VDDX and VDDA supplies. An additional 1KΩ resistor between emitter and base of the BJT is required.

8.2.8 BCTLC — Base Control Pin for external PNP for VDDC

BCTLC is the ballast connection for the on chip voltage regulator. It provides the base current of an external BJT (PNP) of the VDDC supply. An additional 1KΩ resistor between emitter and base of the BJT is required.

8.2.9 VSS — Core Logic Ground Pin

VSS is the core logic supply return pins. It must be grounded.

8.2.10 VDD — Internal Regulator Output Supply (Core Logic)

Node VDD is a device internal supply output of the voltage regulator that provides the power supply for the internal core logic.

This supply domain is monitored by the Low Voltage Reset circuit and The Power On Reset circuit.

8.2.11 VDDF — Internal Regulator Output Supply (NVM Logic)

Node VDDF is a device internal supply output of the voltage regulator that provides the power supply for the NVM logic.

This supply domain is monitored by the Low Voltage Reset circuit.

8.2.12 API_EXTCLK — API external clock output pin

This pin provides the signal selected via APIES and is enabled with APIEA bit. See the device specification if this clock output is available on this device and to which pin it might be connected.

8.5.5 Computer Operating Properly Watchdog (COP) Reset

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication that the software is no longer being executed in the intended sequence; thus COP reset is generated.

The clock source for the COP is either ACLK, IRCCLK or OSCCLK depending on the setting of the COPOSCSEL0 and COPOSCSEL1 bit.
Due to clock domain crossing synchronization there is a latency time to enter and exit Stop Mode if the COP clock source is ACLK and this clock is stopped in Stop Mode. This maximum total latency time is 4 ACLK cycles (2 ACLK cycles for Stop Mode entry and exit each) which must be added to the Stop Mode recovery time t_{STP_REC} from exit of current Stop Mode to entry of next Stop Mode. This latency time occurs no matter which Stop Mode (Full, Pseudo) is currently exited or entered next.
After exit from Stop Mode (Pseudo, Full) for this latency time of 2 ACLK cycles no Stop Mode request (STOP instruction) should be generated to make sure the COP counter can increment at each Stop Mode exit.
Table 8-34 gives an overview of the COP condition (run, static) in Stop Mode depending on legal configuration and status bit settings:

Table 8-34. COP condition (run, static) in Stop Mode

COPOSCSEL1	CSAD	PSTP	PCE	COPOSCSEL0	OSCE	UPOSC	COP counter behavior in Stop Mode (clock source)
1	0	x	x	x	x	x	Run (ACLK)
1	1	x	x	x	x	x	Static (ACLK)
0	x	1	1	1	1	1	Run (OSCCLK)
0	x	1	1	0	0	x	Static (IRCCLK)
0	x	1	1	0	1	x	Static (IRCCLK)
0	x	1	0	0	x	x	Static (IRCCLK)
0	x	1	0	1	1	1	Static (OSCCLK)
0	x	0	1	1	1	1	Static (OSCCLK)
0	x	0	1	0	1	x	Static (IRCCLK)
0	x	0	1	0	0	0	Static (IRCCLK)
0	x	0	0	1	1	1	Satic (OSCCLK)
0	x	0	0	0	1	1	Static (IRCCLK)
0	x	0	0	0	1	0	Static (IRCCLK)
0	x	0	0	0	0	0	Static (IRCCLK)

Three control bits in the CPMUCOP register allow selection of seven COP time-out periods.

9.6 Resets

At reset the ADC12B_LBA is disabled and in a power down state. The reset state of each individual bit is listed within Section 9.4.2, “Register Descriptions” which details the registers and their bit-fields.

9.7 Interrupts

The ADC supports three types of interrupts:

- Conversion Interrupt
- Sequence Abort Interrupt
- Error and Conversion Flow Control Issue Interrupt

Each of the interrupt types is associated with individual interrupt enable bits and interrupt flags.

9.7.1 ADC Conversion Interrupt

The ADC provides one conversion interrupt associated to 16 interrupt enable bits with dedicated interrupt flags. The 16 interrupt flags consist of:

- 15 conversion interrupt flags which can be associated to any conversion completion.
- One additional interrupt flag which is fixed to the “End Of List” conversion command type within the active CSL.

The association of the conversion number with the interrupt flag number is done in the conversion command.

9.7.2 ADC Sequence Abort Done Interrupt

The ADC provides one sequence abort done interrupt associated with the sequence abort request for conversion flow control. Hence, there is only one dedicated interrupt flag and interrupt enable bit for conversion sequence abort and it occurs when the sequence abort is done.

Table 11-19. Pin Action

PAMOD	PEDGE	Pin Action
0	0	Falling edge
0	1	Rising edge
1	0	Div. by 64 clock enabled with pin high level
1	1	Div. by 64 clock enabled with pin low level

NOTE

If the timer is not active (TEN = 0 in TSCR), there is no divide-by-64 because the ÷64 clock is generated by the timer prescaler.

Table 11-20. Timer Clock Selection

CLK1	CLK0	Timer Clock
0	0	Use timer prescaler clock as timer counter clock
0	1	Use PACLK as input to timer counter clock
1	0	Use PACLK/256 as timer counter clock frequency
1	1	Use PACLK/65536 as timer counter clock frequency

For the description of PACLK please refer Figure 11-30**.**

If the pulse accumulator is disabled (PAEN = 0), the prescaler clock from the timer is always used as an input clock to the timer counter. The change from one selected clock to the other happens immediately after these bits are written.

11.3.2.16 Pulse Accumulator Flag Register (PAFLG)

[1] .

Module Base + 0x0021

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

Unimplemented or Reserved

Figure 11-25. Pulse Accumulator Flag Register (PAFLG)

Read: Anytime

Write: Anytime

When the TFFCA bit in the TSCR register is set, any access to the PACNT register will clear all the flags in the PAFLG register. Timer module or Pulse Accumulator must stay enabled (TEN=1 or PAEN=1) while clearing these bits.

[1] The register is available only if corresponding channel exists.

12.3.2.1 Timer Input Capture/Output Compare Select (TIOS)

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R W	RESERVED	RESERVED	RESERVED	RESERVED	IOS3	IOS2	IOS1	IOS0
Reset	0	0	0	0	0	0	0	0

Figure 12-4. Timer Input Capture/Output Compare Select (TIOS)

Read: Anytime

Write: Anytime

Table 12-2. TIOS Field Descriptions

Note: Writing to unavailable bits has no effect. Reading from unavailable bits return a zero.

Field	Description
3:0 IOS[3:0]	**Input Capture or Output Compare Channel Configuration** 0 The corresponding implemented channel acts as an input capture. 1 The corresponding implemented channel acts as an output compare.

12.3.2.2 Timer Compare Force Register (CFORC)

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	RESERVED	RESERVED	RESERVED	RESERVED	FOC3	FOC2	FOC1	FOC0
Reset	0	0	0	0	0	0	0	0

Figure 12-5. Timer Compare Force Register (CFORC)

Read: Anytime but will always return 0x0000 (1 state is transient)

Write: Anytime

Table 12-3. CFORC Field Descriptions

Note: Writing to unavailable bits has no effect. Reading from unavailable bits return a zero.

Field	Description
3:0 FOC[3:0]	**Note: Force Output Compare Action for Channel 3:0 —** A write to this register with the corresponding data bit(s) set causes the action which is programmed for output compare “x” to occur immediately. The action taken is the same as if a successful comparison had just taken place with the TCx register except the interrupt flag does not get set. If forced output compare on any channel occurs at the same time as the successful output compare then forced output compare action will take precedence and interrupt flag won’t get set.

In freeze mode there is a software programmable option to disable the input clock to the prescaler. This is useful for emulation.

Wait: The prescaler keeps on running, unless PSWAI in PWMCTL is set to 1.

Freeze: The prescaler keeps on running, unless PFRZ in PWMCTL is set to 1.

13.1.3 Block Diagram

Figure 13-1 shows the block diagram for the 8-bit up to 8-channel scalable PWM block.

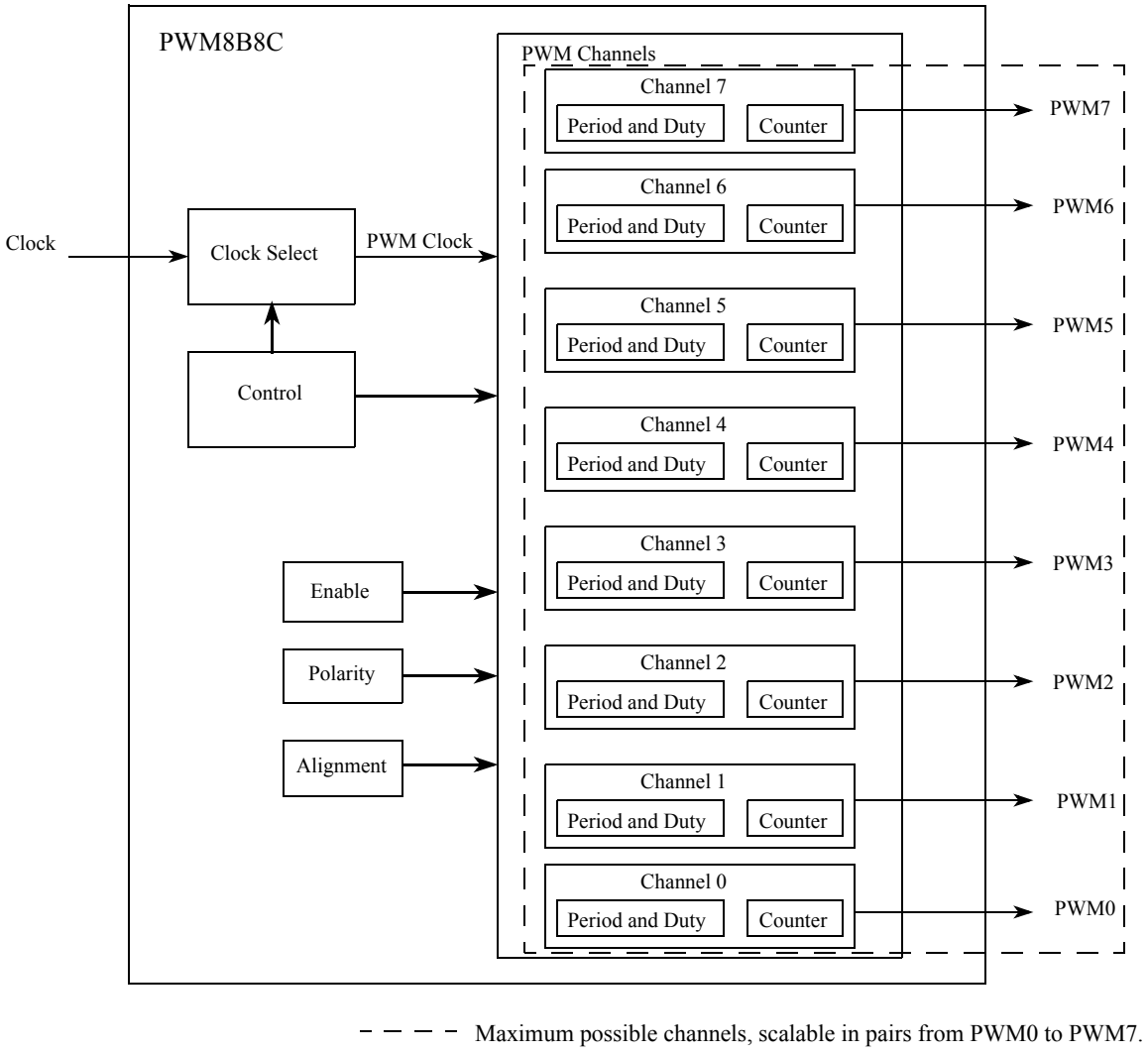

Figure 13-1. Scalable PWM Block Diagram

13.2 External Signal Description

The scalable PWM module has a selected number of external pins. Refer to device specification for exact number.

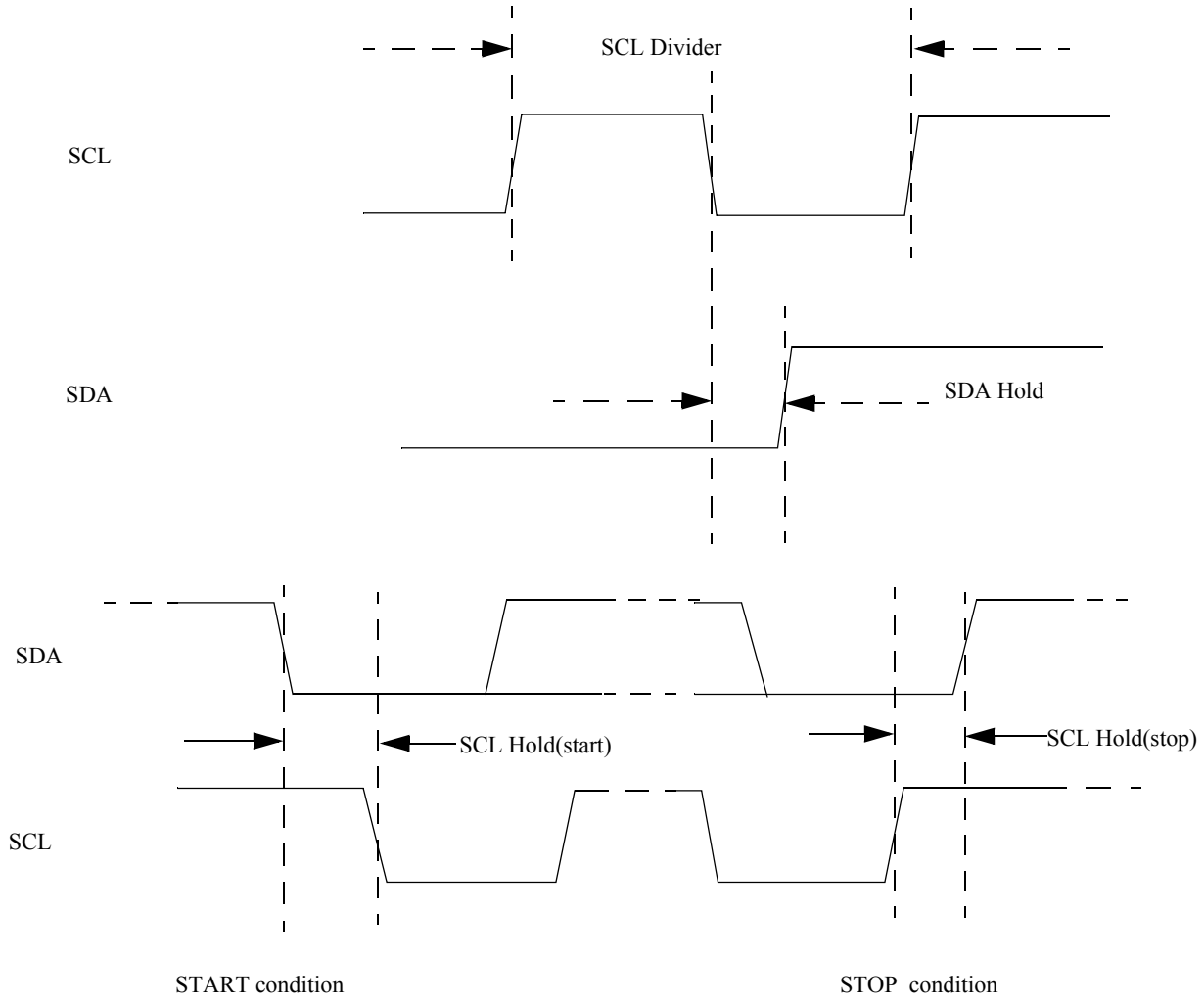

Figure 16-5. SCL Divider and SDA Hold

The equation used to generate the divider values from the IBFD bits is:

SCL Divider = MUL x {2 x (scl2tap + [(SCL_Tap -1) x tap2tap] + 2)}

The SDA hold delay is equal to the CPU clock period multiplied by the SDA Hold value shown in Table 16-7. The equation used to generate the SDA Hold value from the IBFD bits is:

SDA Hold = MUL x {scl2tap + [(SDA_Tap - 1) x tap2tap] + 3}

The equation for SCL Hold values to generate the start and stop conditions from the IBFD bits is:

SCL Hold(start) = MUL x [scl2start + (SCL_Tap - 1) x tap2tap]

SCL Hold(stop) = MUL x [scl2stop + (SCL_Tap - 1) x tap2tap]

Table 16-7. IIC Divider and Hold Values (Sheet 1 of 6)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=1				

16.4.1.11 General Call Address

To broadcast using a general call, a device must first generate the general call address($00), then after receiving acknowledge, it must transmit data.

In communication, as a slave device, provided the GCEN is asserted, a device acknowledges the broadcast and receives data until the GCEN is disabled or the master device releases the bus or generates a new transfer. In the broadcast, slaves always act as receivers. In general call, IAAS is also used to indicate the address match.

In order to distinguish whether the address match is the normal address match or the general call address match, IBDR should be read after the address byte has been received. If the data is $00, the match is general call address match. The meaning of the general call address is always specified in the first data byte and must be dealt with by S/W, the IIC hardware does not decode and process the first data byte.

When one byte transfer is done, the received data can be read from IBDR. The user can control the procedure by enabling or disabling GCEN.

16.4.2 Operation in Run Mode

This is the basic mode of operation.

16.4.3 Operation in Wait Mode

IIC operation in wait mode can be configured. Depending on the state of internal bits, the IIC can operate normally when the CPU is in wait mode or the IIC clock generation can be turned off and the IIC module enters a power conservation state during wait mode. In the later case, any transmission or reception in progress stops at wait mode entry.

16.4.4 Operation in Stop Mode

The IIC is inactive in stop mode for reduced power consumption. The STOP instruction does not affect IIC register states.

16.5 Resets

The reset state of each individual bit is listed in Section 16.3, “Memory Map and Register Definition,” which details the registers and their bit-fields.

16.6 Interrupts

IICV3 uses only one interrupt vector.

Table 16-11. Interrupt Summary

Interrupt	Offset	Vector	Priority	Source	Description
IIC Interrupt	—	—	—	IBAL, TCF, IAAS bits in IBSR register	When either of IBAL, TCF or IAAS bits is set may cause an interrupt based on arbitration lost, transfer complete or address detect conditions

[1] Read: Anytime
Write: Anytime when not in initialization mode

NOTE

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 18-13. CANTIER Register Field Descriptions

Field	Description
2-0 TXEIE[2:0]	**Transmitter Empty Interrupt Enable** 0 No interrupt request is generated from this event. 1 A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.

18.3.2.9 MSCAN Transmitter Message Abort Request Register (CANTARQ)

The CANTARQ register allows abort request of queued messages as described below.

Module Base + 0x0008 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 18-12. MSCAN Transmitter Message Abort Request Register (CANTARQ)

[1] Read: Anytime
Write: Anytime when not in initialization mode

NOTE

The CANTARQ register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 18-14. CANTARQ Register Field Descriptions

Field	Description
2-0 ABTRQ[2:0]	**Abort Request** — The CPU sets the ABTRQx bit to request that a scheduled message buffer (TXEx = 0) be aborted. The MSCAN grants the request if the message has not already started transmission, or if the transmission is not successful (lost arbitration or error). When a message is aborted, the associated TXE (see Section 18.3.2.7, “MSCAN Transmitter Flag Register (CANTFLG)”) and abort acknowledge flags (ABTAK, see Section 18.3.2.10, “MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)”) are set and a transmit interrupt occurs if enabled. The CPU cannot reset ABTRQx. ABTRQx is reset whenever the associated TXE flag is set. 0 No abort request 1 Abort request pending

Module Base + 0x00X2

	7	6	5	4	3	2	1	0
R W	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
Reset:	x	x	x	x	x	x	x	x

Figure 18-28. Identifier Register 2 (IDR2) — Extended Identifier Mapping

Table 18-28. IDR2 Register Field Descriptions — Extended

Field	Description
7-0 ID[14:7]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
Reset:	x	x	x	x	x	x	x	x

Figure 18-29. Identifier Register 3 (IDR3) — Extended Identifier Mapping

Table 18-29. IDR3 Register Field Descriptions — Extended

Field	Description
7-1 ID[6:0]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
0 RTR	**Remote Transmission Request** — This flag reflects the status of the remote transmission request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame

If the WUPE bit in CANCTL0 is not asserted, the MSCAN will mask any activity it detects on CAN. RXCAN is therefore held internally in a recessive state. This locks the MSCAN in sleep mode. WUPE must be set before entering sleep mode to take effect.

The MSCAN is able to leave sleep mode (wake up) only when:

- CAN bus activity occurs and WUPE = 1

  or
- the CPU clears the SLPRQ bit

NOTE

The CPU cannot clear the SLPRQ bit before sleep mode (SLPRQ = 1 and SLPAK = 1) is active.

After wake-up, the MSCAN waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, if the MSCAN is woken-up by a CAN frame, this frame is not received.

The receive message buffers (RxFG and RxBG) contain messages if they were received before sleep mode was entered. All pending actions will be executed upon wake-up; copying of RxBG into RxFG, message aborts and message transmissions. If the MSCAN remains in bus-off state after sleep mode was exited, it continues counting the 128 occurrences of 11 consecutive recessive bits.

18.4.5.6 MSCAN Power Down Mode

The MSCAN is in power down mode (Table 18-37) when

- CPU is in stop mode

  or
- CPU is in wait mode and the CSWAI bit is set

When entering the power down mode, the MSCAN immediately stops all ongoing transmissions and receptions, potentially causing CAN protocol violations. To protect the CAN bus system from fatal consequences of violations to the above rule, the MSCAN immediately drives TXCAN into a recessive state.

NOTE

The user is responsible for ensuring that the MSCAN is not active when power down mode is entered. The recommended procedure is to bring the MSCAN into Sleep mode before the STOP or WAI instruction (if CSWAI is set) is executed. Otherwise, the abort of an ongoing message can cause an error condition and impact other CAN bus devices.

In power down mode, all clocks are stopped and no registers can be accessed. If the MSCAN was not in sleep mode before power down mode became active, the module performs an internal recovery cycle after powering up. This causes some fixed delay before the module enters normal mode again.

when entering normal mode after leaving shutdown mode. During this time the comparator output path to all subsequent logic (ACO, ACIF, timer link) is held in its current state. Refer to I_{ACMP_run} for current consumption of ACMP during operation.

2. Shutdown Mode
The ACMP is held in shutdown mode either when disabled (ACE=0) or during STOP mode. When leaving normal mode the current state of the comparator will be maintained. ACMPO drives zero in shutdown mode if not inverted (ACOPS=0). In this mode the current consumption is reduced to I_{ACMP_off}.

20.6 Memory Map and Register Definition

20.6.1 Register Map

Table 20-2 shows the ACMP register map.

NOTE

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

Table 20-2. ACMP Register Map

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	ACMPC0	R	ACE	ACOPE	ACOPS	ACDLY	ACHYS1-0		ACMOD1-0	
		W								
0x0001	ACMPC1	R	0	0	ACPSEL1-0		0	0	ACNSEL1-0	
		W								
0x0002	ACMPC2	R	0	0	0	0	0	0	0	ACIE
		W								
0x0003	ACMPS	R	ACO	0	0	0	0	0	0	ACIF
		W								
0x0004–0x0007	Reserved	R	0	0	0	0	0	0	0	0
		W								

- Protection scheme to prevent accidental program or erase of EEPROM memory
- Ability to program up to four words in a burst sequence

22.1.2.3 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

22.1.3 Block Diagram

The block diagram of the Flash module is shown in Figure 22-1.

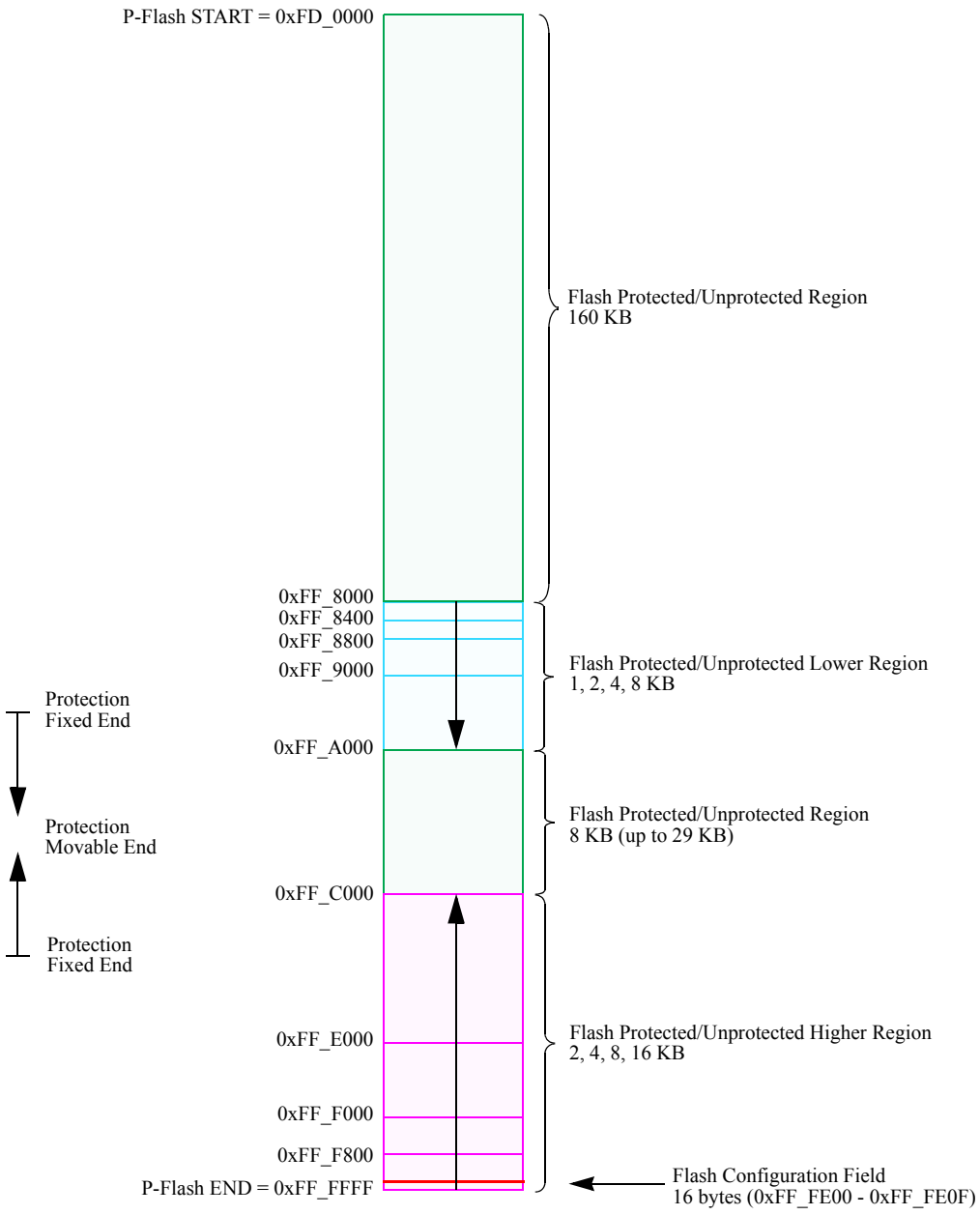

Figure 22-2. P-Flash Memory Map