



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	HCS12X
Core Size	16-Bit
Speed	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs256j0val

2.3.38 Port M Pull Device Enable Register (PERM)

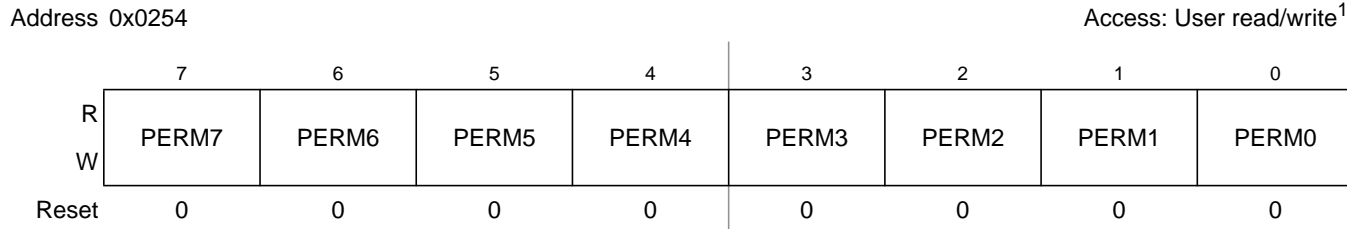


Figure 2-36. Port M Pull Device Enable Register (PERM)

¹ Read: Anytime
Write: Anytime

Table 2-34. PERM Register Field Descriptions

Field	Description
7-0 PERM	Port M pull device enable —Enable pull device on input pin or wired-or output pin This bit controls whether a pull device on the associated port input pin is active. If a pin is used as output this bit has only effect if used in wired-or mode. The polarity is selected by the related polarity select register bit. 1 Pull device enabled 0 Pull device disabled

2.3.39 Port M Polarity Select Register (PPSM)

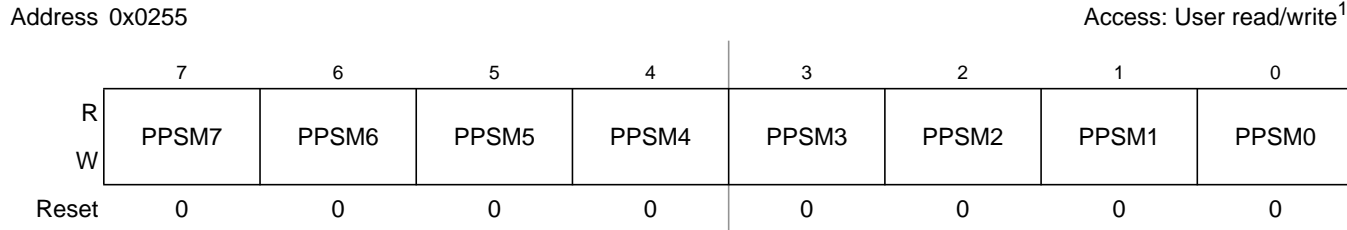


Figure 2-37. Port M Polarity Select Register (PPSM)

¹ Read: Anytime
Write: Anytime

Table 2-35. PPSM Register Field Descriptions

Field	Description
7-0 PPSM	Port M pull device select —Configure pull device polarity on input pin This bit selects a pull-up or a pull-down device if enabled on the associated port input pin. If CAN0 is active the selection of a pull-down device on the RXCAN input will have no effect. 1 A pull-down device selected 0 A pull-up device selected

2.3.57 Port H Interrupt Flag Register (PIFH)

Address 0x0267				Access: User read/write ¹				
	7	6	5	4	3	2	1	0
R	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-55. Port H Interrupt Flag Register (PIFH)

¹ Read: Anytime
Write: Anytime

Table 2-54. PIFH Register Field Descriptions

Field	Description
7-0 PIFH	<p>Port H interrupt flag— The flag bit is set after an active edge was applied to the associated input pin. This can be a rising or a falling edge based on the state of the polarity select register. Writing a logic “1” to the corresponding bit field clears the flag.</p> <p>1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set) 0 No active edge occurred</p>

2.3.58 Port J Data Register (PTJ)

Address 0x0268				Access: User read/write ¹				
	7	6	5	4	3	2	1	0
R	PTJ7	PTJ6	0	0	0	0	PTJ1	PTJ0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-56. Port J Data Register (PTJ)

¹ Read: Anytime, the data source depends on the data direction value
Write: Anytime

Table 2-55. PTJ Register Field Descriptions

Field	Description
7-6, 1-0 PTJ	<p>Port J general purpose input/output data—Data Register, pin interrupt input/output The associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read.</p> <ul style="list-style-type: none"> Pin interrupts can be generated if enabled in input or output mode.

3.4.3 Chip Bus Control

The MMC controls the address buses and the data buses that interface the S12X masters (CPU, BDM) with the rest of the system (master buses). In addition the MMC handles all CPU read data bus swapping operations. All internal resources are connected to specific target buses (see Figure 3-20).

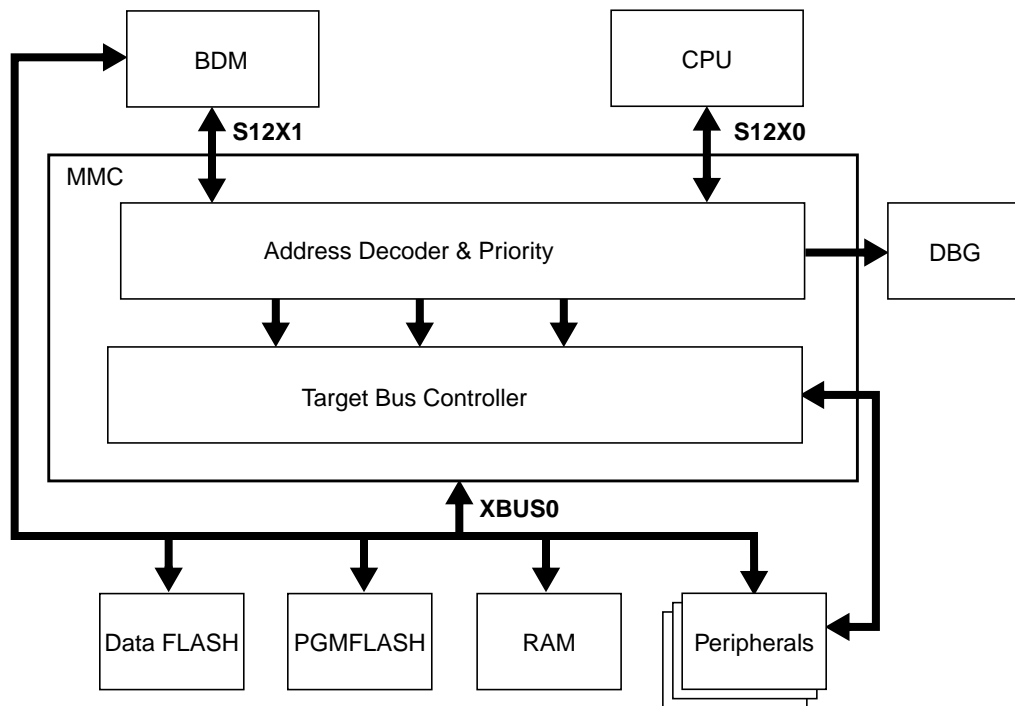


Figure 3-20. MMC Block Diagram

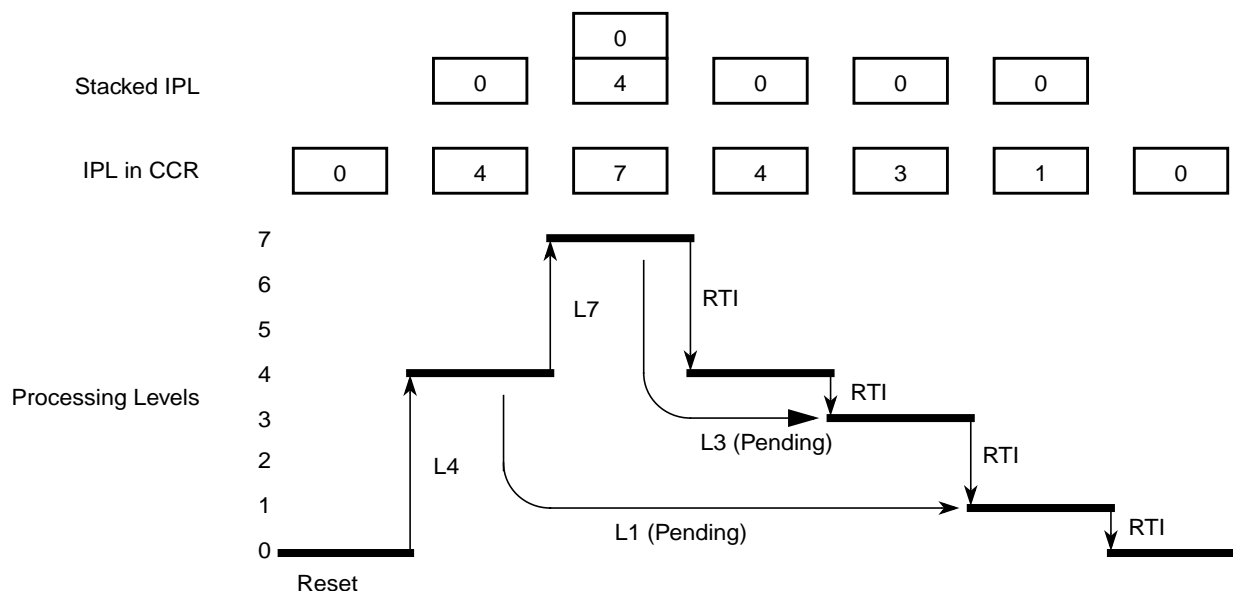


Figure 4-14. Interrupt Processing Example

4.5.3 Wake Up from Stop or Wait Mode

4.5.3.1 CPU Wake Up from Stop or Wait Mode

Only I bit maskable interrupt requests which are configured to be handled by the CPU are capable of waking the MCU from wait mode.

Since bus and core clocks are disabled in stop mode, only interrupt requests that can be generated without these clocks can wake the MCU from stop mode. These are listed in the device overview interrupt vector table. Only I bit maskable interrupt requests which are configured to be handled by the CPU are capable of waking the MCU from stop mode.

To determine whether an I bit maskable interrupt is qualified to wake up the CPU or not, the same settings as in normal run mode are applied during stop or wait mode:

- If the I bit in the CCR is set, all I bit maskable interrupts are masked from waking up the MCU.
- An I bit maskable interrupt is ignored if it is configured to a priority level below or equal to the current IPL in CCR.
- I bit maskable interrupt requests which are configured to be handled by the XGATE module are not capable of waking up the CPU.

The X bit maskable interrupt request can wake up the MCU from stop or wait mode at anytime, even if the X bit in CCR is set. If the X bit maskable interrupt request is used to wake-up the MCU with the X bit in the CCR set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAI or STOP instruction. This features works following the same rules like any interrupt request, i.e. care must be taken that the X interrupt request used for wake-up remains active at least until the system begins execution of the instruction following the WAI or STOP instruction; otherwise, wake-up may not occur.

Table 5-7. Firmware Commands

Command ¹	Opcode (hex)	Data	Description
READ_NEXT ²	62	16-bit data out	Increment X index register by 2 ($X = X + 2$), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT<f-helvetica><st-superscript>	42	16-bit data in	Increment X index register by 2 ($X = X + 2$), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL ³	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

¹ If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

² When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.

³ System stop disables the ACK function and ignored commands will not have an ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge if CPU executes the wait or stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 5.4.7, “Serial Interface Hardware Handshake Protocol” last Note).

5.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

Chapter 10

Analog-to-Digital Converter (ADC12B16CV1)

Table 10-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.00	13 Oct. 2005		Initial version
V01.01	04 Mar. 2008		corrected reference to DJM bit

10.1 Introduction

The ADC12B16C is a 16-channel, 12-bit, multiplexed input successive approximation analog-to-digital converter. Refer to device electrical specifications for ATD accuracy.

10.1.1 Features

- 8-, 10-, or 12-bit resolution.
- Conversion in Stop Mode using internally generated clock
- Automatic return to low power after conversion sequence
- Automatic compare with interrupt for higher than or less/equal than programmable value
- Programmable sample time.
- Left/right justified result data.
- External trigger control.
- Sequence complete interrupt.
- Analog input multiplexer for 16 analog input channels.
- Special conversions for V_{RH} , V_{RL} , $(V_{RL}+V_{RH})/2$.
- 1-to-16 conversion sequence lengths.
- Continuous conversion mode.
- Multiple channel scans.
- Configurable external trigger functionality on any AD channel or any of four additional trigger inputs. The four additional trigger inputs can be chip external or internal. Refer to device specification for availability and connectivity.
- Configurable location for channel wrap around (when converting multiple channels in a sequence).

Module Base + 0x0001

Access: User read/write⁽¹⁾

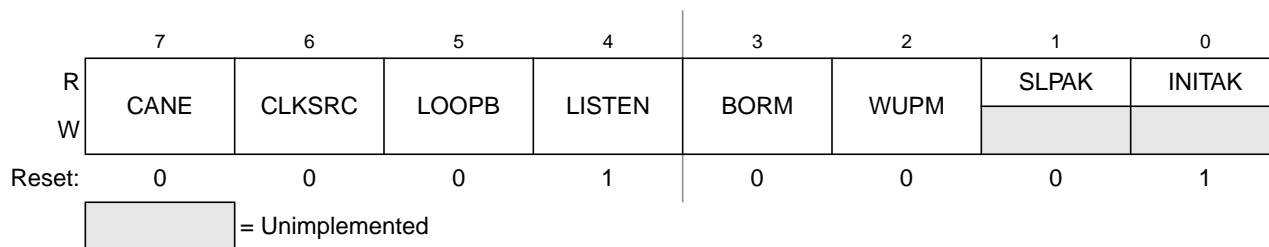


Figure 11-5. MSCAN Control Register 1 (CANCTL1)

1. Read: Anytime

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

Table 11-4. CANCTL1 Register Field Descriptions

Field	Description
7 CANE	MSCAN Enable 0 MSCAN module is disabled 1 MSCAN module is enabled
6 CLKSRC	MSCAN Clock Source — This bit defines the clock source for the MSCAN module (only for systems with a clock generation module; Section 11.4.3.2, "Clock System," and Section Figure 11-43., "MSCAN Clocking Scheme,"). 0 MSCAN clock source is the oscillator clock 1 MSCAN clock source is the bus clock
5 LOOPB	Loopback Self Test Mode — When this bit is set, the MSCAN performs an internal loopback which can be used for self test operation. The bit stream output of the transmitter is fed back to the receiver internally. The RXCAN input is ignored and the TXCAN output goes to the recessive state (logic 1). The MSCAN behaves as it does normally when transmitting and treats its own transmitted message as a message received from a remote node. In this state, the MSCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated. 0 Loopback self test disabled 1 Loopback self test enabled
4 LISTEN	Listen Only Mode — This bit configures the MSCAN as a CAN bus monitor. When LISTEN is set, all valid CAN messages with matching ID are received, but no acknowledgement or error frames are sent out (see Section 11.4.4.4, "Listen-Only Mode"). In addition, the error counters are frozen. Listen only mode supports applications which require "hot plugging" or throughput analysis. The MSCAN is unable to transmit any messages when listen only mode is active. 0 Normal operation 1 Listen only mode activated
3 BORM	Bus-Off Recovery Mode — This bit configures the bus-off state recovery mode of the MSCAN. Refer to Section 11.5.2, "Bus-Off Recovery," for details. 0 Automatic bus-off recovery (see Bosch CAN 2.0A/B protocol specification) 1 Bus-off recovery upon user request
2 WUPM	Wake-Up Mode — If WUPE in CANCTL0 is enabled, this bit defines whether the integrated low-pass filter is applied to protect the MSCAN from spurious wake-up (see Section 11.4.5.5, "MSCAN Sleep Mode"). 0 MSCAN wakes up on any dominant level on the CAN bus 1 MSCAN wakes up only in case of a dominant pulse on the CAN bus that has a length of T_{wup}

11.3.3.3 Data Length Register (DLR)

This register keeps the data length field of the CAN frame.

Module Base + 0x00XC

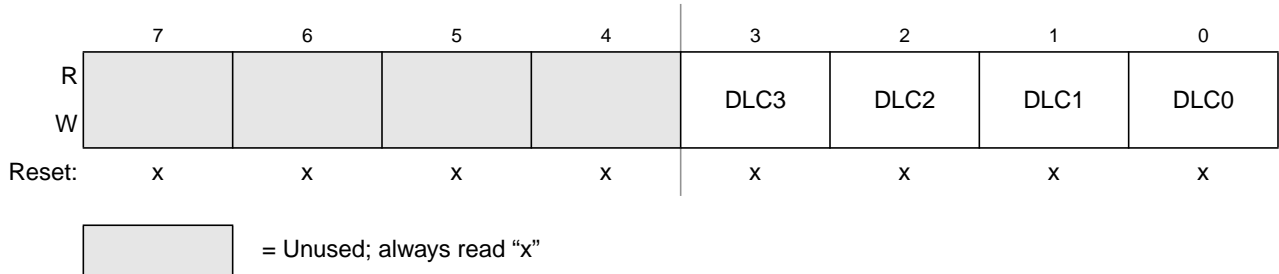


Figure 11-35. Data Length Register (DLR) — Extended Identifier Mapping

Table 11-34. DLR Register Field Descriptions

Field	Description
3-0 DLC[3:0]	Data Length Code Bits — The data length code contains the number of bytes (data byte count) of the respective message. During the transmission of a remote frame, the data length code is transmitted as programmed while the number of transmitted data bytes is always 0. The data byte count ranges from 0 to 8 for a data frame. Table 11-35 shows the effect of setting the DLC bits.

Table 11-35. Data Length Codes

Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8

11.3.3.4 Transmit Buffer Priority Register (TBPR)

This register defines the local priority of the associated message buffer. The local priority is used for the internal prioritization process of the MSCAN and is defined to be highest for the smallest binary number. The MSCAN implements the following internal prioritization mechanisms:

- All transmission buffers with a cleared TXEx flag participate in the prioritization immediately before the SOF (start of frame) is sent.

13.2.1 PWM7 — PWM Channel 7

This pin serves as waveform output of PWM channel 7 and as an input for the emergency shutdown feature.

13.2.2 PWM6 — PWM Channel 6

This pin serves as waveform output of PWM channel 6.

13.2.3 PWM5 — PWM Channel 5

This pin serves as waveform output of PWM channel 5.

13.2.4 PWM4 — PWM Channel 4

This pin serves as waveform output of PWM channel 4.

13.2.5 PWM3 — PWM Channel 3

This pin serves as waveform output of PWM channel 3.

13.2.6 PWM3 — PWM Channel 2

This pin serves as waveform output of PWM channel 2.

13.2.7 PWM3 — PWM Channel 1

This pin serves as waveform output of PWM channel 1.

13.2.8 PWM3 — PWM Channel 0

This pin serves as waveform output of PWM channel 0.

13.3 Memory Map and Register Definition

This section describes in detail all the registers and register bits in the PWM module.

The special-purpose registers and register bit functions that are not normally available to device end users, such as factory test control registers and reserved registers, are clearly identified by means of shading the appropriate portions of address maps and register diagrams. Notes explaining the reasons for restricting access to the registers and functions are also explained in the individual register descriptions.

13.3.1 Module Memory Map

This section describes the content of the registers in the PWM module. The base address of the PWM module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset. The figure below shows the registers associated

Clock A is used as an input to an 8-bit down counter. This down counter loads a user programmable scale value from the scale register (PWMSCLA). When the down counter reaches one, a pulse is output and the 8-bit counter is re-loaded. The output signal from this circuit is further divided by two. This gives a greater range with only a slight reduction in granularity. Clock SA equals clock A divided by two times the value in the PWMSCLA register.

NOTE

$$\text{Clock SA} = \text{Clock A} / (2 * \text{PWMSCLA})$$

When PWMSCLA = \$00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Similarly, clock B is used as an input to an 8-bit down counter followed by a divide by two producing clock SB. Thus, clock SB equals clock B divided by two times the value in the PWMSCLB register.

NOTE

$$\text{Clock SB} = \text{Clock B} / (2 * \text{PWMSCLB})$$

When PWMSCLB = \$00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

As an example, consider the case in which the user writes \$FF into the PWMSCLA register. Clock A for this case will be E divided by 4. A pulse will occur at a rate of once every 255×4 E cycles. Passing this through the divide by two circuit produces a clock signal at an E divided by 2040 rate. Similarly, a value of \$01 in the PWMSCLA register when clock A is E divided by 4 will produce a clock at an E divided by 8 rate.

Writing to PWMSCLA or PWMSCLB causes the associated 8-bit down counter to be re-loaded. Otherwise, when changing rates the counter would have to count down to \$01 before counting at the proper rate. Forcing the associated counter to re-load the scale register value every time PWMSCLA or PWMSCLB is written prevents this.

NOTE

Writing to the scale registers while channels are operating can cause irregularities in the PWM outputs.

13.4.1.3 Clock Select

Each PWM channel has the capability of selecting one of two clocks. For channels 0, 1, 4, and 5 the clock choices are clock A or clock SA. For channels 2, 3, 6, and 7 the choices are clock B or clock SB. The clock selection is done with the PCLKx control bits in the PWMCLK register.

NOTE

Changing clock control bits while channels are operating can cause irregularities in the PWM outputs.

Figure 14-17 shows two cases of break detect. In trace RXD_1 the break symbol starts with the start bit, while in RXD_2 the break starts in the middle of a transmission. If BRKDFE = 1, in RXD_1 case there will be no byte transferred to the receive buffer and the RDRF flag will not be modified. Also no framing error or parity error will be flagged from this transfer. In RXD_2 case, however the break signal starts later during the transmission. At the expected stop bit position the byte received so far will be transferred to the receive buffer, the receive data register full flag will be set, a framing error and if enabled and appropriate a parity error will be set. Once the break is detected the BRKDIF flag will be set.

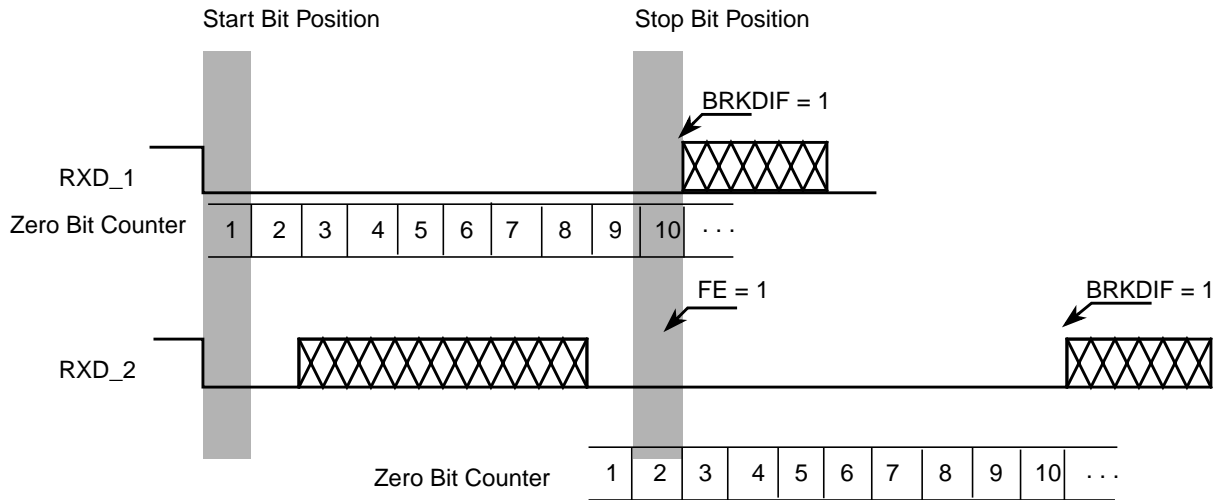


Figure 14-17. Break Detection if BRKDFE = 1 (M = 0)

14.4.5.4 Idle Characters

An idle character (or preamble) contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCI control register 1 (SCICR1). The preamble is a synchronizing idle character that begins the first transmission initiated after writing the TE bit from 0 to 1.

If the TE bit is cleared during a transmission, the TXD pin becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the frame currently being transmitted.

NOTE

When queueing an idle character, return the TE bit to logic 1 before the stop bit of the current frame shifts out through the TXD pin. Setting TE after the stop bit appears on TXD causes data previously written to the SCI data register to be lost. Toggle the TE bit for a queued idle character while the TDRE flag is set and immediately before writing the next byte to the SCI data register.

If the TE bit is clear and the transmission is complete, the SCI is not the master of the TXD pin

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

15.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after \overline{SS} has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

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

After $2n^1$ (last) SCK edges:

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

Figure 15-12 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The \overline{SS} pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

¹. n depends on the selected transfer width, please refer to Section 15.3.2.2, "SPI Control Register 2 (SPICR2)"

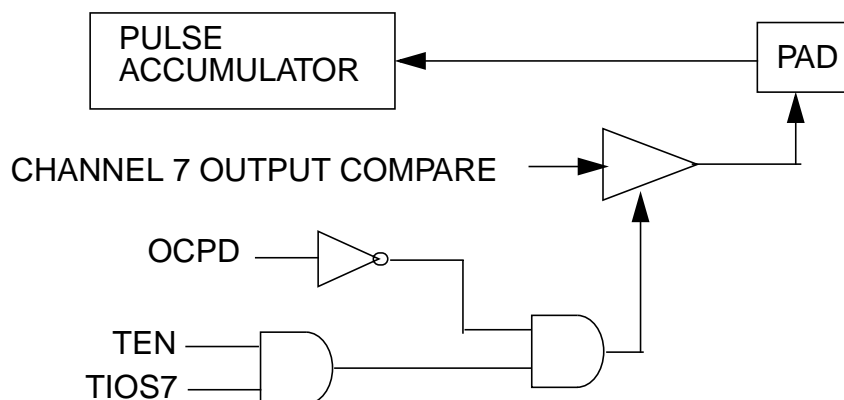


Figure 16-4. Channel 7 Output Compare/Pulse Accumulator Logic

16.2 External Signal Description

The TIM16B8CV2 module has a total of eight external pins.

16.2.1 IOC7 — Input Capture and Output Compare Channel 7 Pin

This pin serves as input capture or output compare for channel 7. This can also be configured as pulse accumulator input.

16.2.2 IOC6 — Input Capture and Output Compare Channel 6 Pin

This pin serves as input capture or output compare for channel 6.

16.2.3 IOC5 — Input Capture and Output Compare Channel 5 Pin

This pin serves as input capture or output compare for channel 5.

16.2.4 IOC4 — Input Capture and Output Compare Channel 4 Pin

This pin serves as input capture or output compare for channel 4. Pin

16.2.5 IOC3 — Input Capture and Output Compare Channel 3 Pin

This pin serves as input capture or output compare for channel 3.

16.2.6 IOC2 — Input Capture and Output Compare Channel 2 Pin

This pin serves as input capture or output compare for channel 2.

Table 17-10. Selectable Autonomous Periodical Interrupt Periods

APICLK	APIR[15:0]	Selected Period
0	0000	0.2 ms ¹
0	0001	0.4 ms ¹
0	0002	0.6 ms ¹
0	0003	0.8 ms ¹
0	0004	1.0 ms ¹
0	0005	1.2 ms ¹
0
0	FFFD	13106.8 ms ¹
0	FFFE	13107.0 ms ¹
0	FFFF	13107.2 ms ¹
1	0000	2 * bus clock period
1	0001	4 * bus clock period
1	0002	6 * bus clock period
1	0003	8 * bus clock period
1	0004	10 * bus clock period
1	0005	12 * bus clock period
1
1	FFFD	131068 * bus clock period
1	FFFE	131070 * bus clock period
1	FFFF	131072 * bus clock period

¹ When trimmed within specified accuracy. See electrical specifications for details.

The period can be calculated as follows depending of APICLK:

$$\text{Period} = 2 * (\text{APIR}[15:0] + 1) * 0.1 \text{ ms} \quad \text{or} \quad \text{period} = 2 * (\text{APIR}[15:0] + 1) * \text{bus clock period}$$

Table 18-48. Erase P-Flash 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 18-28)
		Set if an invalid global address [22:16] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the selected P-Flash sector 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

18.4.2.10 Unsecure Flash Command

The Unsecure Flash command will erase the entire P-Flash and D-Flash memory space and, if the erase is successful, will release security.

Table 18-49. Unsecure Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and D-Flash memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 18-50. Unsecure Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 18-28)
	FPVIOL	Set if any area of the P-Flash or D-Flash 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

18.4.2.11 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 18-9). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see

keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 18-10) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash block 0 will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 18.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 18.4.2.11
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x7F_FF00–0x7F_FF07 in the Flash configuration field.

The security as defined in the Flash security byte (0x7F_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x7F_FF00–0x7F_FF07 are unaffected by the Verify Backdoor Access Key command sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte (0x7F_FF0F). The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

18.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

The MCU can be unsecured in special single chip mode by erasing the P-Flash and D-Flash memory by one of the following methods:

- Reset the MCU into special single chip mode, delay while the erase test is performed by the BDM, send BDM commands to disable protection in the P-Flash and D-Flash memory, and execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.
- Reset the MCU into special expanded wide mode, disable protection in the P-Flash and D-Flash memory and run code from external memory to execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.

After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode. The BDM will execute the Erase Verify All Blocks command write sequence to verify that the P-Flash and D-Flash memory is erased. If the P-Flash and D-Flash memory are verified as

Table 19-4. Program IFR Fields

Global Address (PGMIFRON)	Size (Bytes)	Field Description
0x40_0000 – 0x40_0007	8	Device ID
0x40_0008 – 0x40_00E7	224	Reserved
0x40_00E8 – 0x40_00E9	2	Version ID
0x40_00EA – 0x40_00FF	22	Reserved
0x40_0100 – 0x40_013F	64	Program Once Field Refer to Section 19.4.2.6, “Program Once Command”
0x40_0140 – 0x40_01FF	192	Reserved

Table 19-5. D-Flash and Memory Controller Resource Fields

Global Address	Size (Bytes)	Description
0x10_0000 – 0x10_1FFF	8,192	D-Flash Memory
0x10_2000 – 0x11_FFFF	122,880	Reserved
0x12_0000 – 0x12_007F	128	D-Flash Nonvolatile Information Register (DFIFRON ¹ = 1)
0x12_0080 – 0x12_0FFF	3,968	Reserved
0x12_1000 – 0x12_1FFF	4,096	Reserved
0x12_2000 – 0x12_3CFF	7,242	Reserved
0x12_3D00 – 0x12_3FFF	768	Memory Controller Scratch RAM (MGRAMON ¹ = 1)
0x12_4000 – 0x12_E7FF	43,008	Reserved
0x12_E800 – 0x12_FFFF	6,144	Reserved
0x13_0000 – 0x13_FFFF	65,536	Reserved

¹ MMCCTL1 register bit

Chapter 20

64 KByte Flash Module (S12XFTMR64K1V1)

Table 20-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.04	03 Jan 2008		- Cosmetic changes
V01.05	19 Dec 2008	20.1/20-607 20.4.2.4/20-642 20.4.2.6/20-644 20.4.2.11/20-648 20.4.2.11/20-648 20.4.2.11/20-648	- Clarify single bit fault correction for P-Flash phrase - Add statement concerning code runaway when executing Read Once, Program Once, and Verify Backdoor Access Key commands from Flash block containing associated fields - Relate Key 0 to associated Backdoor Comparison Key address - Change “power down reset” to “reset” in Section 20.4.2.11
V01.06	25 Sep 2009	20.3.2/20-615 20.3.2.1/20-617 20.4.1.2/20-636 20.6/20-656	The following changes were made to clarify module behavior related to Flash register access during reset sequence and while Flash commands are active: - Add caution concerning register writes while command is active - Writes to FCLKDIV are allowed during reset sequence while CCIF is clear - Add caution concerning register writes while command is active - Writes to FCCOBIX, FCCOBHI, FCCOBLO registers are ignored during reset sequence

20.1 Introduction

The FTMR64K1 module implements the following:

- 64 Kbytes of P-Flash (Program Flash) memory
- 4 Kbytes of D-Flash (Data Flash) memory

The Flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller that executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

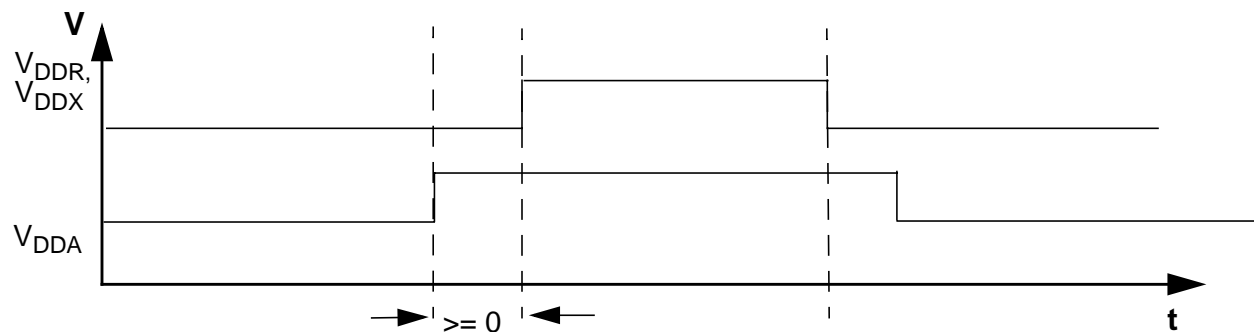


Figure A-3. S12XS family Power Sequencing

During power sequencing V_{DDA} can be powered up before V_{DDR} , V_{DDX} .
 V_{DDR} and V_{DDX} must be powered up together adhering to the operating conditions differential.
 V_{RH} power up must follow V_{DDA} to avoid current injection.

0x0100–0x0113 NVM Control Register (FTMR) Map (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0104	FCNFG	R	CCIE	0	0	IGNSF	0	0	DFDIE	FSFD
		W								
0x0105	FERCNFG	R	0	0	0	0	0	0	DFDIE	SFDIE
		W								
0x0106	FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
		W								
0x0107	FERSTAT	R	0	0	0	0	0	0	DFDIF	SFDIF
		W								
0x0108	FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
		W								
0x0109	DFPROT	R	DPOPEN	0	0	DPS4	DPS3	DPS2	DPS1	DPS0
		W								
0x010A	FCCOBHI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
		W								
0x010B	FCCOBLO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
		W								
0x010C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x010D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x010E	FECCRHI	R	ECCR15	ECCR14	ECCR13	ECCR12	ECCR11	ECCR10	ECCR9	ECCR8
		W								
0x010F	FECCRL0	R	ECCR7	ECCR6	ECCR5	ECCR4	ECCR3	ECCR2	ECCR1	ECCR0
		W								
0x0110	FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
		W								
0x0111	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0112	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0113	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0114–0x011F Reserved Register Space

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0114-0x011F	Reserved	R	0	0	0	0	0	0	0	0
		W								