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	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	119
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	14K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xd256mag

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://freescale.com/

A full list of family members and options is included in the appendices.

Read page 29 first to understand the maskset specific chapters of this document

This document contains information for all constituent modules, with the exception of the S12X CPU. For S12X CPU information please refer to the CPU S12X Reference Manual.

Section Number	Title	Page

5.3.2.3 ATD Control Register 2 (ATDCTL2)

This register controls power down, interrupt and external trigger. Writes to this register will abort current conversion sequence but will not start a new sequence.

	7	6	5	4	3	2	1	0
R	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ASCIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-5. ATD Control Register 2 (ATDCTL2)

Read: Anytime

Write: Anytime

Table 5-5. ATDCTL2 Field Descriptions

Field	Description
7 ADPU	**ATD Power Up** — This bit provides on/off control over the ATD block allowing reduced MCU power consumption. Because analog electronic is turned off when powered down, the ATD requires a recovery time period after ADPU bit is enabled. 0 Power down ATD 1 Normal ATD functionality
6 AFFC	**ATD Fast Flag Clear All** 0 ATD flag clearing operates normally (read the status register ATDSTAT1 before reading the result register to clear the associate CCF flag). 1 Changes all ATD conversion complete flags to a fast clear sequence. Any access to a result register will cause the associate CCF flag to clear automatically.
5 AWAI	**ATD Power Down in Wait Mode** — When entering wait mode this bit provides on/off control over the ATD block allowing reduced MCU power. Because analog electronic is turned off when powered down, the ATD requires a recovery time period after exit from Wait mode. 0 ATD continues to run in Wait mode 1 Halt conversion and power down ATD during wait mode After exiting wait mode with an interrupt conversion will resume. But due to the recovery time the result of this conversion should be ignored.
4 ETRIGLE	**External Trigger Level/Edge Control** — This bit controls the sensitivity of the external trigger signal. See Table 5-6 for details.
3 ETRIGP	**External Trigger Polarity** — This bit controls the polarity of the external trigger signal. See Table 5-6 for details.
2 ETRIGE	**External Trigger Mode Enable** — This bit enables the external trigger on one of the AD channels or one of the ETRIG3–0 inputs as described in Table 5-4. If external trigger source is one of the AD channels, the digital input buffer of this channel is enabled. The external trigger allows to synchronize sample and ATD conversions processes with external events. 0 Disable external trigger 1 Enable external trigger **Note:** If using one of the AD channel as external trigger (ETRIGSEL = 0) the conversion results for this channel have no meaning while external trigger mode is enabled.

6.4 Functional Description

The core of the XGATE module is a RISC processor which is able to access the MCU's internal memories and peripherals (see Figure 6-1). The RISC processor always remains in an idle state until it is triggered by an XGATE request. Then it executes a code sequence that is associated with the request and optionally triggers an interrupt to the S12X_CPU upon completion. Code sequences are not interruptible. A new XGATE request can only be serviced when the previous sequence is finished and the RISC core becomes idle.

The XGATE module also provides a set of hardware semaphores which are necessary to ensure data consistency whenever RAM locations or peripherals are shared with the S12X_CPU.

The following sections describe the components of the XGATE module in further detail.

6.4.1 XGATE RISC Core

The RISC core is a 16 bit processor with an instruction set that is well suited for data transfers, bit manipulations, and simple arithmetic operations (see Section 6.8, "Instruction Set").

It is able to access the MCU's internal memories and peripherals without blocking these resources from the S12X_CPU[1]. Whenever the S12X_CPU and the RISC core access the same resource, the RISC core will be stalled until the resource becomes available again[1].

The XGATE offers a high access rate to the MCU's internal RAM. Depending on the bus load, the RISC core can perform up to two RAM accesses per S12X_CPU bus cycle.

Bus accesses to peripheral registers or flash are slower. A transfer rate of one bus access per S12X_CPU cycle can not be exceeded.

The XGATE module is intended to execute short interrupt service routines that are triggered by peripheral modules or by software.

6.4.2 Programmer's Model

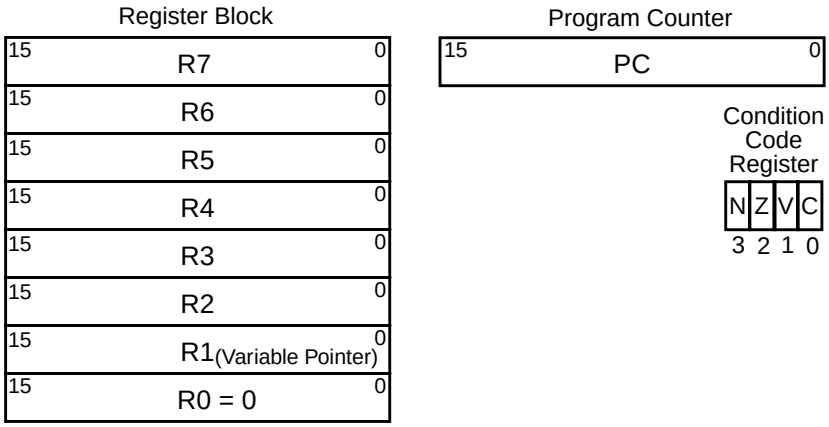

Figure 6-19. Programmer's Model

1. With the exception of PRR registers (see Section "S12X_MMC").

BFFO Bit Field Find First One BFFO

Operation

FirstOne (RS) ⇒ RD;

Searches the first “1” in register RS (from MSB to LSB) and writes the bit position into the destination register RD. The upper bits of RD are cleared. In case the content of RS is equal to $0000, RD will be cleared and the carry flag will be set. This is used to distinguish a “1” in position 0 versus no “1” in the whole RS register at all.

CCR Effects

N	Z	V	C
0	Δ	0	Δ

N: 0; cleared.
Z: Set if the result is $0000; cleared otherwise.
V: 0; cleared.
C: Set if RS = $0000[1]; cleared otherwise.

[1] Before executing the instruction

Code and CPU Cycles

Source Form	Address Mode	Machine Code											Cycles
BFFO RD, RS	DYA	0	0	0	0	1	RD	RS	1	0	0	0	P

	15	14	13	12	11	10	9	8
R	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-63. Timer Input Capture Holding Register 3 High (TC3H)

	7	6	5	4	3	2	1	0
R	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-64. Timer Input Capture Holding Register 3 Low (TC3H)

Read: Anytime

Write: Has no effect.

All bits reset to zero.

These registers are used to latch the value of the input capture registers TC0–TC3. The corresponding IOSx bits in TIOS should be cleared (see Section 7.4.1.1, “IC Channels”).

7.4 Functional Description

This section provides a complete functional description of the ECT block, detailing the operation of the design from the end user perspective in a number of subsections.

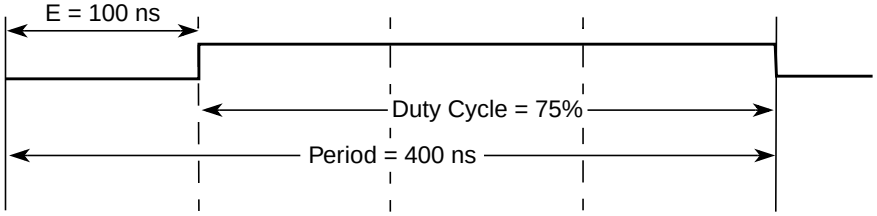

Figure 8-21. PWM Left Aligned Output Example Waveform

8.4.2.6 Center Aligned Outputs

For center aligned output mode selection, set the CAEx bit (CAEx = 1) in the PWMCAE register and the corresponding PWM output will be center aligned.

The 8-bit counter operates as an up/down counter in this mode and is set to up whenever the counter is equal to $00. The counter compares to two registers, a duty register and a period register as shown in the block diagram in Figure 8-19. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register changes the counter direction from an up-count to a down-count. When the PWM counter decrements and matches the duty register again, the output flip-flop changes state causing the PWM output to also change state. When the PWM counter decrements and reaches zero, the counter direction changes from a down-count back to an up-count and a load from the double buffer period and duty registers to the associated registers is performed, as described in Section 8.4.2.3, “PWM Period and Duty”. The counter counts from 0 up to the value in the period register and then back down to 0. Thus the effective period is PWMPERx*2.

NOTE

Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

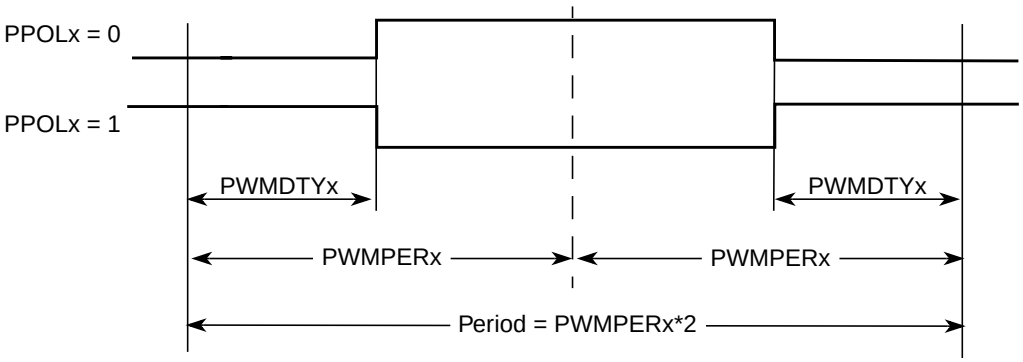

Figure 8-22. PWM Center Aligned Output Waveform

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

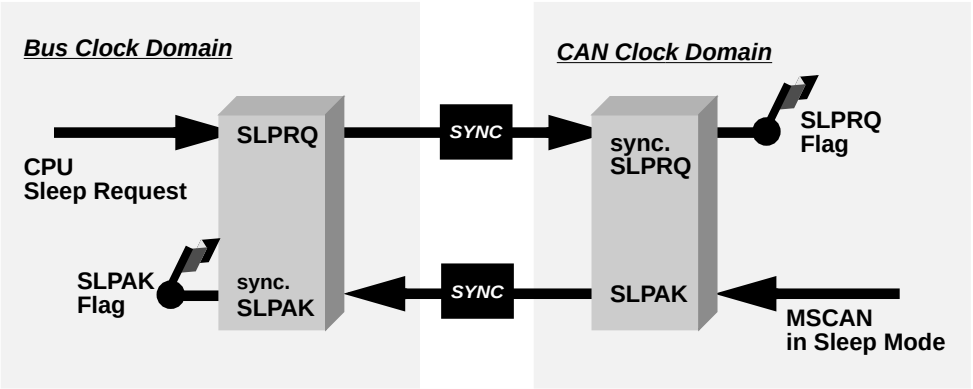

Figure 10-45. Sleep Request / Acknowledge Cycle

NOTE

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

If sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 10-45). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode.

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

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

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

If the WUPE bit in CANCTL0 is not asserted, the MSCAN will mask any activity it detects on CAN. The RXCAN pin is therefore held internally in a recessive state. This locks the MSCAN in sleep mode (Figure 10-46). 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

Table 12-6. Example SPI Baud Rate Selection (25 MHz Bus Clock)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	0	0	0	0	2	12.5 MHz
0	0	0	0	0	1	4	6.25 MHz
0	0	0	0	1	0	8	3.125 MHz
0	0	0	0	1	1	16	1.5625 MHz
0	0	0	1	0	0	32	781.25 kHz
0	0	0	1	0	1	64	390.63 kHz
0	0	0	1	1	0	128	195.31 kHz
0	0	0	1	1	1	256	97.66 kHz
0	0	1	0	0	0	4	6.25 MHz
0	0	1	0	0	1	8	3.125 MHz
0	0	1	0	1	0	16	1.5625 MHz
0	0	1	0	1	1	32	781.25 kHz
0	0	1	1	0	0	64	390.63 kHz
0	0	1	1	0	1	128	195.31 kHz
0	0	1	1	1	0	256	97.66 kHz
0	0	1	1	1	1	512	48.83 kHz
0	1	0	0	0	0	6	4.16667 MHz
0	1	0	0	0	1	12	2.08333 MHz
0	1	0	0	1	0	24	1.04167 MHz
0	1	0	0	1	1	48	520.83 kHz
0	1	0	1	0	0	96	260.42 kHz
0	1	0	1	0	1	192	130.21 kHz
0	1	0	1	1	0	384	65.10 kHz
0	1	0	1	1	1	768	32.55 kHz
0	1	1	0	0	0	8	3.125 MHz
0	1	1	0	0	1	16	1.5625 MHz
0	1	1	0	1	0	32	781.25 kHz
0	1	1	0	1	1	64	390.63 kHz
0	1	1	1	0	0	128	195.31 kHz
0	1	1	1	0	1	256	97.66 kHz
0	1	1	1	1	0	512	48.83 kHz
0	1	1	1	1	1	1024	24.41 kHz
1	0	0	0	0	0	10	2.5 MHz
1	0	0	0	0	1	20	1.25 MHz
1	0	0	0	1	0	40	625 kHz
1	0	0	0	1	1	80	312.5 kHz
1	0	0	1	0	0	160	156.25 kHz
1	0	0	1	0	1	320	78.13 kHz
1	0	0	1	1	0	640	39.06 kHz

18.3.2.12 RAM Shared Region Upper Boundary Register (RAMSHU)

Address: 0x011F

	7	6	5	4	3	2	1	0
R	1	SHU6	SHU5	SHU4	SHU3	SHU2	SHU1	SHU0
W								
Reset	1	1	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 18-20. RAM Shared Region Upper Boundary Register (RAMSHU)

Read: Anytime

Write: Anytime when RWPE = 0

Table 18-18. RAMSHU Field Descriptions

Field	Description
6–0 SHU[6:0]	**RAM Shared Region Upper Boundary Bits 6–0** — These bits define the upper boundary of the shared memory in multiples of 256 bytes. The block selected by this register is included in the region. See Figure 18-25 for details.

19.4.4.1 Final State

On entering final state a trigger may be issued to the trace buffer according to the trace position control as defined by the TALIGN field (see Section 19.3.1.3, “Debug Trace Control Register (DBGTCR)”). If the TSOURCE bits in the trace control register DBGTCR are cleared then the trace buffer is disabled and the transition to final state can only generate a breakpoint request. In this case or upon completion of a tracing session when tracing is enabled, the ARM bit in the DBGC1 register is cleared, returning the module to the disarmed state0. If tracing is enabled a breakpoint request can occur at the end of the tracing session.

19.4.5 Trace Buffer Operation

The trace buffer is a 64 lines deep by 64-bits wide RAM array. The DBG module stores trace information in the RAM array in a circular buffer format. The CPU accesses the RAM array through a register window (DBGTBH:DBGTBL) using 16-bit wide word accesses. After each complete 64-bit trace buffer line is read via the CPU, an internal pointer into the RAM is incremented so that the next read will receive fresh information. Data is stored in the format shown in Table 19-39. After each store the counter register bits DBGCNT[6:0] are incremented. Tracing of CPU activity is disabled when the BDM is active but tracing of XGATE activity is still possible. Reading the trace buffer while the BDM is active returns invalid data and the trace buffer pointer is not incremented.

19.4.5.1 Trace Trigger Alignment

Using the TALIGN bits (see Section 19.3.1.3, “Debug Trace Control Register (DBGTCR)”) it is possible to align the trigger with the end, the middle or the beginning of a tracing session.

If end or mid tracing is selected, tracing begins when the ARM bit in DBGC1 is set and State1 is entered. The transition to final state if end is selected signals the end of the tracing session. The transition to final state if mid is selected signals that another 32 lines will be traced before ending the tracing session. Tracing with begin-trigger starts at the opcode of the trigger.

19.4.5.1.1 Storing with Begin-Trigger

Storing with begin-trigger, data is not stored in the trace buffer until the final state is entered. Once the trigger condition is met the DBG module will remain armed until 64 lines are stored in the trace buffer. If the trigger is at the address of the change-of-flow instruction the change of flow associated with the trigger will be stored in the trace buffer. Using begin-trigger together with tagging, if the tagged instruction is about to be executed then the trace is started. Upon completion of the tracing session the breakpoint is generated, thus the breakpoint does not occur at the tagged instruction boundary.

19.4.5.1.2 Storing with Mid-Trigger

Storing with mid-trigger, data is stored in the trace buffer as soon as the DBG module is armed. When the trigger condition is met, another 32 lines will be traced before ending the tracing session, irrespective of the number of lines stored before the trigger occurred, then the DBG module is disarmed and no more data is stored. If the trigger is at the address of a change of flow instruction the trigger event is not stored in the trace buffer. Using mid-trigger with tagging, if the tagged instruction is about to be executed then the trace

20.4.6.1 External Tagging using $\overline{\text{TAGHI}}$ and $\overline{\text{TAGLO}}$

External tagging using the external $\overline{\text{TAGHI}}$ and $\overline{\text{TAGLO}}$ pins can only be used to tag S12XCPU opcodes; tagging of XGATE code using these pins is not possible. An external tag triggers the state sequencer into state0 when the tagged opcode reaches the execution stage of the instruction queue.

The pins operate independently, thus the state of one pin does not affect the function of the other. External tagging is possible in emulation modes only. The presence of logic level 0 on either pin at the rising edge of the external clock (ECLK) performs the function indicated in the Table 20-43. It is possible to tag both bytes of an instruction word. If a taghit occurs, a breakpoint can be generated as defined by the DBGBRK and BDM bits in DBGC1. Each time $\overline{\text{TAGHI}}$ or $\overline{\text{TAGLO}}$ are low on the rising edge of ECLK, the old tag is replaced by a new one.

Table 20-43. Tag Pin Function

$\overline{\text{TAGHI}}$	$\overline{\text{TAGLO}}$	Tag
1	1	No tag
1	0	Low byte
0	1	High byte
0	0	Both bytes

20.4.6.2 Unconditional Tagging Function

In emulation modes a low assertion of PE5/$\overline{\text{TAGLO}}$/MODA in the 7th or 8th bus cycle after reset enables the unconditional tagging function, allowing immediate tagging via $\overline{\text{TAGHI}}$/$\overline{\text{TAGLO}}$ with breakpoint to BDM independent of the ARM, BDM and DBGBRK bits. Conversely these bits are not affected by unconditional tagging. The unconditional tagging function remains enabled until the next reset. This function allows an immediate entry to BDM in emulation modes before user code execution. The $\overline{\text{TAGLO}}$ assertion must be in the 7th or 8th bus cycle following the end of reset, whereby the prior $\overline{\text{RESET}}$ pin assertion lasts the full 192 bus cycles.

20.4.7 Breakpoints

There are several ways to generate breakpoints to the XGATE and S12XCPU modules

- Through XGATE software breakpoint requests.
- From comparator channel triggers to final state.
- Using software to write to the TRIG bit in the DBGC1 register.
- From taghits generated using the external $\overline{\text{TAGHI}}$ and $\overline{\text{TAGLO}}$ pins.
- Through the auxilliary forced breakpoint input.

20.4.7.1 XGATE Software Breakpoints

The XGATE software breakpoint instruction BRK can request an S12XCPU breakpoint, via the S12XDBG module. In this case, if the XGSBPE bit is set, the S12XDBG module immediately generates a forced breakpoint request to the S12XCPU, the state sequencer is returned to state0 and tracing, if active, is terminated. If configured for BEGIN trigger and tracing has not yet been triggered from another source, the trace buffer contains no information. Breakpoint requests from the XGATE module do not depend

21.1.3 Block Diagram

Figure 21-1 is a block diagram of the XEBI with all related I/O signals.

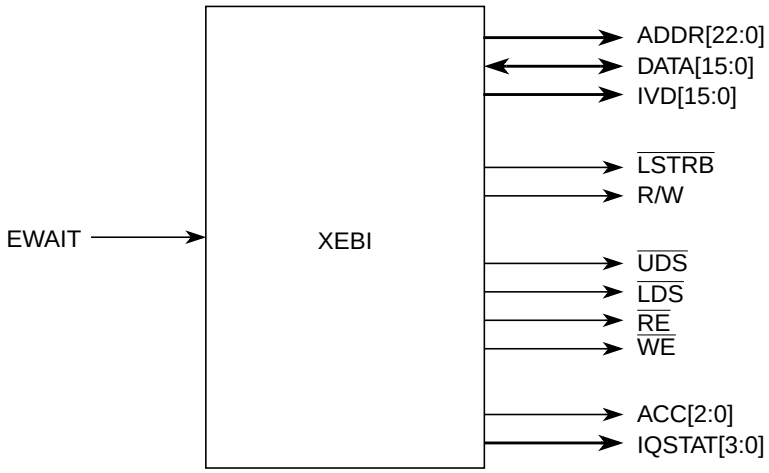

Figure 21-1. XEBI Block Diagram

21.2 External Signal Description

The user is advised to refer to the SoC section for port configuration and location of external bus signals.

NOTE

The following external bus related signals are described in other sections:
$\overline{CS2}$, $\overline{CS1}$, $\overline{CS0}$ (chip selects) — S12X_MMC section
ECLK, ECLKX2 (free-running clocks) — PIM section
$\overline{TAGHI}$, $\overline{TAGLO}$ (tag inputs) — PIM section, S12X_DBG section

Table 21-1 outlines the pin names and gives a brief description of their function. Refer to the SoC section and PIM section for reset states of these pins and associated pull-ups or pull-downs.

Table 22-16. ECLKCTL Field Descriptions (continued)

Field	Description
6 NCLKX2	**No ECLKX2** — This bit controls the availability of a free-running clock on the ECLKX2 pin. This clock has a fixed rate of twice the internal bus clock. Clock output is always active in emulation modes and if enabled in all other operating modes. 0 ECLKX2 is enabled 1 ECLKX2 is disabled
1–0 EDIV[1:0]	**Free-Running ECLK Divider** — These bits determine the rate of the free-running clock on the ECLK pin. The usage of the bits is shown in Table 22-17. Divider is always disabled in emulation modes and active as programmed in all other operating modes.

Table 22-17. Free-Running ECLK Clock Rate

EDIV[1:0]	Rate of Free-Running ECLK
00	ECLK = Bus clock rate
01	ECLK = Bus clock rate divided by 2
10	ECLK = Bus clock rate divided by 3
11	ECLK = Bus clock rate divided by 4

22.3.2.14 IRQ Control Register (IRQCR)

	7	6	5	4	3	2	1	0
R	IRQE	IRQEN	0	0	0	0	0	0
W								
Reset	0	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 22-16. IRQ Control Register (IRQCR)

Read: See individual bit descriptions below.

Write: See individual bit descriptions below.

Table 22-18. IRQCR Field Descriptions

Field	Description
7 IRQE	**IRQ Select Edge Sensitive Only** Special modes: Read or write anytime. Normal and emulation modes: Read anytime, write once. 0 $\overline{\text{IRQ}}$ configured for low level recognition. 1 $\overline{\text{IRQ}}$ configured to respond only to falling edges. Falling edges on the $\overline{\text{IRQ}}$ pin will be detected anytime IRQE = 1 and will be cleared only upon a reset or the servicing of the $\overline{\text{IRQ}}$ interrupt.
6 IRQEN	**External IRQ Enable** Read or write anytime. 0 External $\overline{\text{IRQ}}$ pin is disconnected from interrupt logic. 1 External $\overline{\text{IRQ}}$ pin is connected to interrupt logic.

Chapter 23
DQ256 Port Integration Module (S12XDQ256PIMV2)

Introduction

The S12XD family port integration module (below referred to as PIM) establishes the interface between the peripheral modules including the non-multiplexed external bus interface module (S12X_EBI) and the I/O pins for all ports. It controls the electrical pin properties as well as the signal prioritization and multiplexing on shared pins.

This document covers the description of:

- Port A, B used as address output of the S12X_EBI and Port C, D used as data I/O of the S12X_EBI
- Port E associated with the S12X_EBI control signals and the IRQ, XIRQ interrupt inputs
- Port K associated with address output and control signals of the S12X_EBI
- Port T connected to the Enhanced Capture Timer (ECT) module
- Port S associated with 2 SCI and 1 SPI modules
- Port M associated with 3 MSCAN modules
- Port P connected to the PWM and 2 SPI modules — inputs can be used as an external interrupt source
- Port H associated with 1 SCI module — inputs can be used as an external interrupt source
- Port J associated with 1 MSCAN, 1 SCI, and 1 IIC module — inputs can be used as an external interrupt source
- Port AD0 and AD1 associated with one 8-channel andone 16-channel ATD module

Most I/O pins can be configured by register bits to select data direction and drive strength, to enable and select pull-up or pull-down devices. Interrupts can be enabled on specific pins resulting in status flags.

The I/O’s of 2 MSCAN and 3 SPI modules can be routed from their default location to alternative port pins.

NOTE

The implementation of the PIM is device dependent. Therefore some functions are not available on certain derivatives or 112-pin and 80-pin package options.

23.0.1 Features

A full-featured PIM module includes these distinctive registers:

- Data and data direction registers for Ports A, B, C, D, E, K, T, S, M, P, H, J, AD0, and AD1 when used as general-purpose I/O

This register is associated with AD0 pins PAD[23:10]. These pins can also be used as general purpose I/O.

If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.

23.0.5.63 Port AD0 Data Direction Register 1 (DDR1AD0)

	7	6	5	4	3	2	1	0
R W	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
Reset	0	0	0	0	0	0	0	0

Figure 23-65. Port AD0 Data Direction Register 1 (DDR1AD0)

Read: Anytime.

Write: Anytime.

This register configures pins PAD[07:00] as either input or output.

Table 23-58. DDR1AD0 Field Descriptions

Field	Description
7–0 DDR1AD0[7:0]	**Data Direction Port AD0 Register 1** 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTAD01 register, when changing the DDR1AD0 register. **Note:** To use the digital input function on port AD0 the ATD0 digital input enable register (ATD0DIEN) has to be set to logic level “1”.

23.0.5.64 Port AD0 Reduced Drive Register 1 (RDR1AD0)

	7	6	5	4	3	2	1	0
R W	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
Reset	0	0	0	0	0	0	0	0

Figure 23-66. Port AD0 Reduced Drive Register 1 (RDR1AD0)

Read: Anytime.

Write: Anytime.

This register configures the drive strength of each output pin PAD[07:00] as either full or reduced. If the port is used as input this bit is ignored.

Chapter 24
DG128 Port Integration Module (S12XDG128PIMV2)

Introduction

The S12XD family port integration module (below referred to as PIM) establishes the interface between the peripheral modules including the non-multiplexed external bus interface module (S12X_EBI) and the I/O pins for all ports. It controls the electrical pin properties as well as the signal prioritization and multiplexing on shared pins.

This document covers the description of:

- Port A, B
- Port E associated with the IRQ, XIRQ interrupt inputs
- Port K
- Port T connected to the Enhanced Capture Timer (ECT) module
- Port S associated with 2 SCI and 1 SPI modules
- Port M associated with 2 MSCAN modules and 1 SCI module
- Port P connected to the PWM and 1 SPI module — inputs can be used as an external interrupt source
- Port H associated with 2 SCI modules — inputs can be used as an external interrupt source
- Port J associated with 1 MSCAN, 1 SCI, and 1 IIC module — inputs can be used as an external interrupt source
- Port AD1 associated with one 16-channel ATD module

Most I/O pins can be configured by register bits to select data direction and drive strength, to enable and select pull-up or pull-down devices. Interrupts can be enabled on specific pins resulting in status flags.

The I/O's of 2 MSCAN and 2 SPI modules can be routed from their default location to alternative port pins.

NOTE

The implementation of the PIM is device dependent. Therefore some functions are not available on certain derivatives or80-pin package options.

24.0.1 Features

A full-featured PIM module includes these distinctive registers:

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

= Unimplemented or Reserved

Figure 27-22. RESERVED3

All bits read 0 and are not writable.

27.3.2.14 RESERVED4

This register is reserved for factory testing and is not accessible.

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

= Unimplemented or Reserved

Figure 27-23. RESERVED4

All bits read 0 and are not writable.

27.4 Functional Description

27.4.1 Flash Command Operations

Write operations are used to execute program, erase, erase verify, erase abort, and data compress algorithms described in this section. The program and erase algorithms are controlled by a state machine whose timebase, FCLK, is derived from the oscillator clock via a programmable divider. The command register, as well as the associated address and data registers, operate as a buffer and a register (2-stage FIFO) so that a second command along with the necessary data and address can be stored to the buffer while the first command is still in progress. This pipelined operation allows a time optimization when programming more than one word on a specific row in the Flash block as the high voltage generation can be kept active in between two programming commands. The pipelined operation also allows a simplification of command launching. Buffer empty as well as command completion are signalled by flags in the Flash status register with corresponding interrupts generated, if enabled.

The next sections describe:

1. How to write the FCLKDIV register
2. Command write sequences to program, erase, erase verify, erase abort, and data compress operations on the Flash memory
3. Valid Flash commands
4. Effects resulting from illegal Flash command write sequences or aborting Flash operations

29.4.2.2.1 Data Compress Operation

The Flash module contains a 16-bit multiple-input signature register (MISR) to generate a 16-bit signature based on selected Flash array data. The final 16-bit signature, found in the FDATA registers after the data compress operation has completed, is based on the following logic equation which is executed on every data compression cycle during the operation:

MISR[15:0] = {MISR[14:0], ^MISR[15,4,2,1]} ^ DATA[15:0] ***Eqn. 29-1***

where MISR is the content of the internal signature register and DATA is the data to be compressed as shown in Figure 29-25.

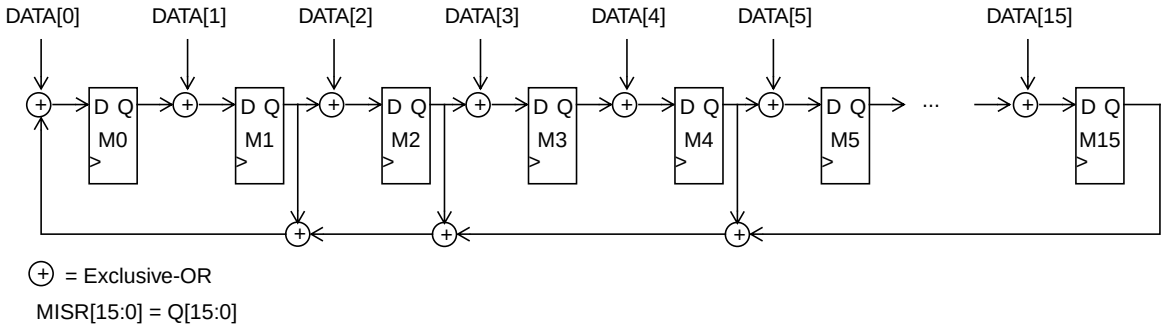

Figure 29-25. 16-Bit MISR Diagram

During the data compress operation, the following steps are executed:

1. MISR is reset to 0xFFFF.
2. Initialized DATA equal to 0xFFFF is compressed into the MISR which results in the MISR containing 0x0001.
3. DATA equal to the selected Flash array data range is read and compressed into the MISR with addresses incrementing.
4. DATA equal to the selected Flash array data range is read and compressed into the MISR with addresses decrementing.
5. DATA equal to the contents of the MISR is compressed into the same MISR.
6. The contents of the MISR are written to the FDATA registers.

E.7 Pinout explanations:

- A/D is the number of modules/total number of A/D channels.
- I/O is the sum of ports capable to act as digital input or output.

  – 144 Pin Packages:
  Port A = 8, B = 8, C=8, D=8, E = 6 + 2 input only,
  H = 8, J = 7, K = 8, M = 8, P = 8, S = 8, T = 8, PAD = 24
  25 inputs provide Interrupt capability (H =8, P= 8, J = 7, IRQ, XIRQ)

  – 112 Pin Packages:
  Port A = 8, B = 8, E = 6 + 2 input only, H = 8, J = 4, K = 7, M = 8, P = 8, S = 8, T = 8, PAD = 16
  22 inputs provide Interrupt capability (H =8, P= 8, J = 4, IRQ, XIRQ)

  – 80 Pin Packages:
  Port A = 8, B = 8, E = 6 + 2 input only, J = 2, M = 6, P = 7, S = 4, T = 8, PAD = 8
  11 inputs provide Interrupt capability (P= 7, J = 2, IRQ, XIRQ)

- CAN0 can be routed under software control from PM[1:0] to pins PM[3:2] or PM[5:4] or PJ[7:6].
- CAN4 pins are shared between IIC0 pins.
- CAN4 can be routed under software control from PJ[7:6] to pins PM[5:4] or PM[7:6].
- Versions with 4 CAN modules will have CAN0, CAN1, CAN2 and CAN4
- Versions with 3 CAN modules will have CAN0, CAN1 and CAN4.
- Versions with 2 CAN modules will have CAN0 and CAN4.
- Versions with 1 CAN modules will have CAN0
- Versions with 2 SPI modules will have SPI0 and SPI1.
- Versions with 4 SCI modules will have SCI0, SCI1, SCI2 and SCI4.
- Versions with 2 SCI modules will have SCI0 and SCI1.
- Versions with 1 IIC module will have IIC0.
- SPI0 can be routed to either Ports PS[7:4] or PM[5:2].
- SPI1 pins are shared with PWM[3:0]; In 144 and 112-pin versions, SPI1 can be routed under software control to PH[3:0].
- SPI2 pins are shared with PWM[7:4]; In 144 and 112-pin versions, SPI2 can be routed under software control to PH[7:4]. In 80-pin packages, $\overline{SS}$-signal of SPI2 is not bonded out!