

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	HCS12
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TFQFN Exposed Pad
Supplier Device Package	48-QFN-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12p64j0cft

1.5 Device Memory Map

Table 1-2 shows the device register memory map.

Table 1-2. Device Register Memory Map

Address	Module	Size (Bytes)
0x0000–0x0009	PIM (port integration module)	10
0x000A–0x000B	MMC (memory map control)	2
0x000C–0x000D	PIM (port integration module)	2
0x000E–0x000F	Reserved	2
0x0010–0x0017	MMC (memory map control)	8
0x0018–0x0019	Reserved	2
0x001A–0x001B	Device ID register	2
0x001C–0x001F	PIM (port integration module)	4
0x0020–0x002F	DBG (debug module)	16
0x0030–0x0033	Reserved	4
0x0034–0x003F	CPMU (clock and power management)	12
0x0040–0x006F	TIM (timer module)	48
0x0070–0x009F	ATD (analog-to-digital converter 12 bit 10-channel)	48
0x00A0–0x00C7	PWM (pulse-width modulator 6 channels)	40
0x00C8–0x00CF	SCI (serial communications interface)	8
0x00D0–0x00D7	Reserved	8
0x00D8–0x00DF	SPI (serial peripheral interface)	8
0x00E0–0x00FF	Reserved	32
0x0100–0x0113	FTMRC control registers	20
0x0114–0x011F	Reserved	12
0x0120	INT (interrupt module)	1
0x0121–0x013F	Reserved	31
0x0140–0x017F	CAN	64
0x0180–0x023F	Reserved	192
0x0240–0x027F	PIM (port integration module)	64
0x0280–0x02BF	Reserved	64
0x02C0–0x02EF	Reserved	48
0x02F0–0x02FF	CPMU (clock and power management)	16
0x0300–0x03FF	Reserved	256

Table 1-12. Interrupt Vector Locations (Sheet 3 of 3)

Vector Address ⁽¹⁾	Interrupt Source	CCR Mask	Local Enable	Wake up from STOP	Wakeup from WAIT
Vector base + \$AE to Vector base + \$90	Reserved				
Vector base + \$8E	Port P interrupt	I bit	PIEP (PIEP7,PIEP5-PIEP0)	Yes	Yes
Vector base+ \$8C	PWM emergency shutdown	I bit	PWMSDN (PWMIE)	No	Yes
Vector base + \$8A	Low-voltage interrupt (LVI)	I bit	CPMUCTRL (LVIE)	No	Yes
Vector base + \$88	Autonomous periodical interrupt (API)	I bit	CPMUAPICTRL (APIE)	Yes	Yes
Vector base + \$86	High temperature interrupt	I bit	CPMUHTCL (HTIE)	No	Yes
Vector base + \$84	ATD compare interrupt	I bit	ATDCTL2 (ACMPIE)	Yes	Yes
Vector base + \$82	Reserved				
Vector base + \$80	Spurious interrupt	—	None	-	-

1. 16 bits vector address based

1.11.3 Effects of Reset

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

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

1.11.3.1 Flash Configuration Reset Sequence Phase

On each reset, the Flash module will hold CPU activity while loading Flash module registers from the Flash memory. If double faults are detected in the reset phase, Flash module protection and security may be active on leaving reset. This is explained in more detail in the Flash module section [13.6 Initialization](#).

1.11.3.2 Reset While Flash Command Active

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

1.11.3.3 I/O Pins

Refer to the PIM section for reset configurations of all peripheral module ports.

1.11.3.4 Memory

The RAM arrays are not initialized out of reset.

1. Read: Anytime
Write: Never, writes to this register have no effect.

Table 2-22. PTIS Register Field Descriptions

Field	Description
3-0 PTIS	Port S input data— A read always returns the buffered input state of the associated pin. It can be used to detect overload or short circuit conditions on output pins.

2.3.26 Port S Data Direction Register (DDRS)

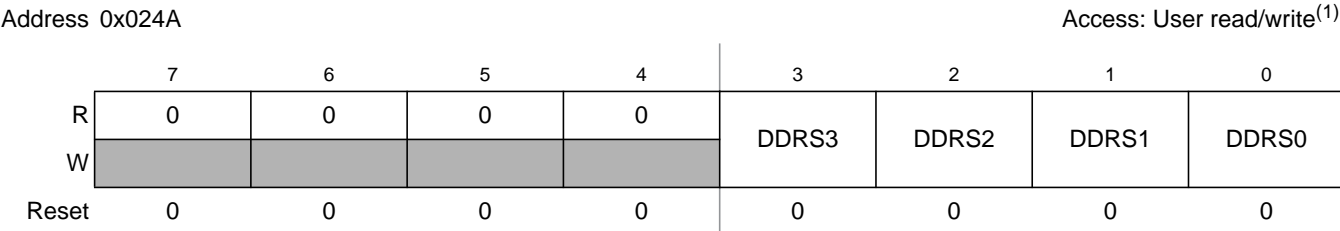


Figure 2-24. Port S Data Direction Register (DDRS)

1. Read: Anytime
Write: Anytime

Table 2-23. DDRS Register Field Descriptions

Field	Description
3-2 DDRS	Port S data direction— This bit determines whether the associated pin is an input or output. 1 Associated pin is configured as output 0 Associated pin is configured as input
1 DDRS	Port S data direction— This bit determines whether the associated pin is an input or output. Depending on the configuration of the enabled SCI the I/O state will be forced to be input or output. In this case the data direction bit will not change. 1 Associated pin is configured as output 0 Associated pin is configured as input
0 DDRS	Port S data direction— This bit determines whether the associated pin is an input or output. Depending on the configuration of the enabled SCI the I/O state will be forced to be input or output. In this case the data direction bit will not change. 1 Associated pin is configured as output 0 Associated pin is configured as input

Table 2-30. DDRM Register Field Descriptions (continued)

Field	Description
2 DDRM	Port M data direction— This bit determines whether the associated pin is an input or output. Depending on the configuration of the enabled SPI the I/O state will be forced to be input or output. In this case the data direction bit will not change. 1 Associated pin is configured as output 0 Associated pin is configured as input
1 DDRM	Port M data direction— This bit determines whether the associated pin is an input or output. The enabled CAN forces the I/O state to be an output. In this case the data direction bit will not change. 1 Associated pin is configured as output 0 Associated pin is configured as input
0 DDRM	Port M data direction— This bit determines whether the associated pin is an input or output. The enabled CAN forces the I/O state to be an input. In this case the data direction bit will not change. 1 Associated pin is configured as output 0 Associated pin is configured as input

2.3.35 Port M Reduced Drive Register (RDRM)

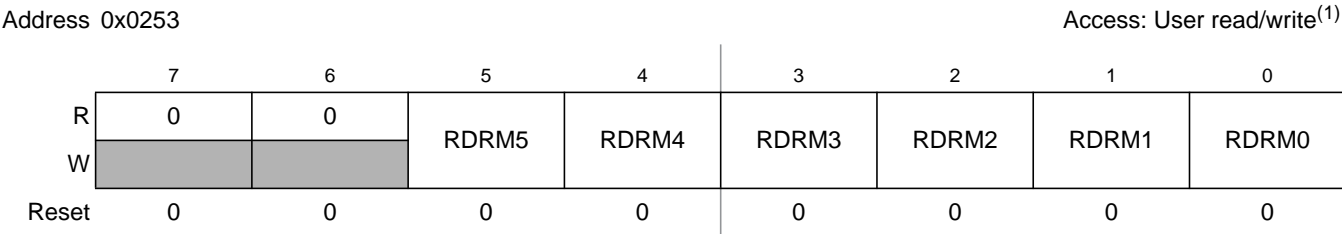


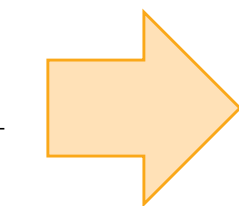
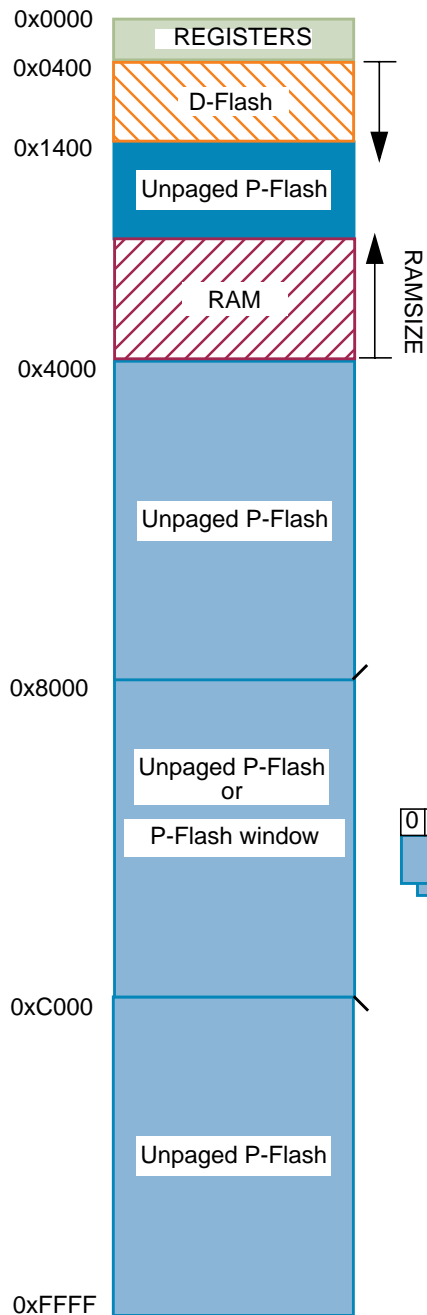
Figure 2-33. Port M Reduced Drive Register (RDRM)

1. Read: Anytime
Write: Anytime

Table 2-31. RDRM Register Field Descriptions

Field	Description
5-0 RDRM	Port M reduced drive— Select reduced drive for output pin This bit configures the drive strength of the associated output pin as either full or reduced. If a pin is used as input this bit has no effect. The reduced drive function is independent of which function is being used on a particular pin. 1 Reduced drive selected (approx. 1/5 of the full drive strength) 0 Full drive strength enabled

CPU and BDM
Local Memory Map



Global Memory Map

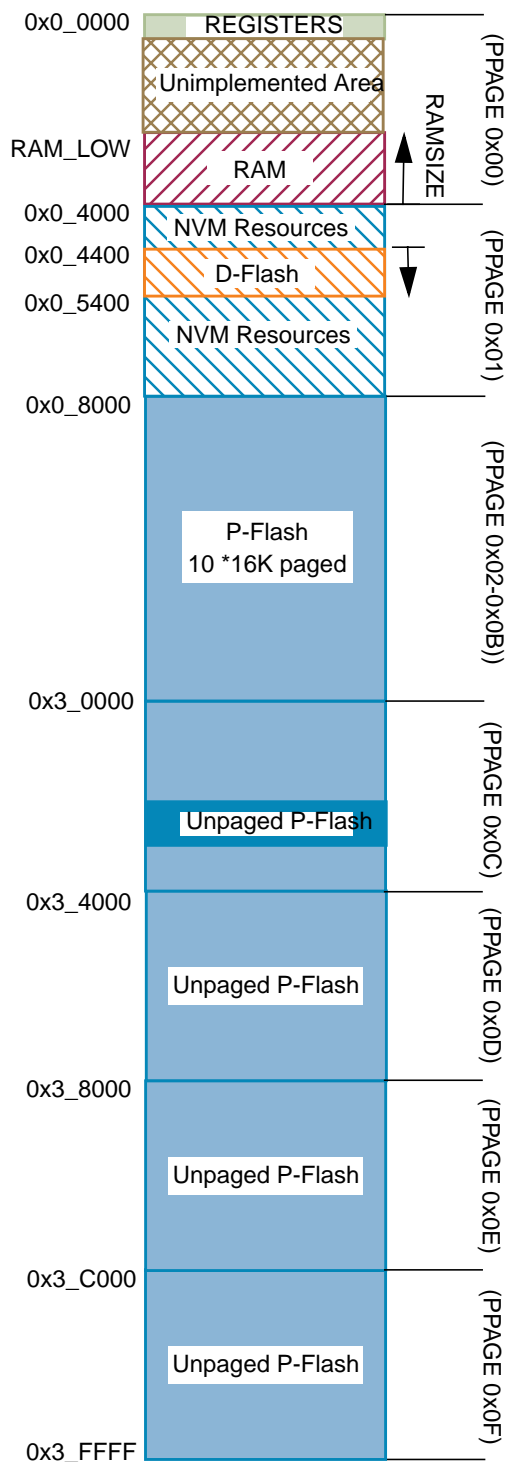


Figure 3-10. Local to Global Address Mapping

5.4.9 SYNC — Request Timed Reference Pulse

The SYNC command is unlike other BDM commands because the host does not necessarily know the correct communication speed to use for BDM communications until after it has analyzed the response to the SYNC command. To issue a SYNC command, the host should perform the following steps:

1. Drive the BKGD pin low for at least 128 cycles at the lowest possible BDM serial communication frequency (**The lowest serial communication frequency is determined by the settings for the VCO clock (CPMUSYNR). The BDM clock frequency is always VCO clock frequency divided by 8.**)
2. Drive BKGD high for a brief speedup pulse to get a fast rise time (this speedup pulse is typically one cycle of the host clock.)
3. Remove all drive to the BKGD pin so it reverts to high impedance.
4. Listen to the BKGD pin for the sync response pulse.

Upon detecting the SYNC request from the host, the target performs the following steps:

1. Discards any incomplete command received or bit retrieved.
2. Waits for BKGD to return to a logic one.
3. Delays 16 cycles to allow the host to stop driving the high speedup pulse.
4. Drives BKGD low for 128 cycles at the current BDM serial communication frequency.
5. Drives a one-cycle high speedup pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high impedance.

The host measures the low time of this 128 cycle SYNC response pulse and determines the correct speed for subsequent BDM communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next negative edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target and so an ACK response pulse will not be issued.

5.4.10 Instruction Tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

handshake protocol is enabled, the time out between a read command and the data retrieval is disabled. Therefore, the host could wait for more than 512 serial clock cycles and still be able to retrieve the data from an issued read command. However, once the handshake pulse (ACK pulse) is issued, the time-out feature is re-activated, meaning that the target will time out after 512 clock cycles. Therefore, the host needs to retrieve the data within a 512 serial clock cycles time frame after the ACK pulse had been issued. After that period, the read command is discarded and the data is no longer available for retrieval. Any negative edge in the BKGD pin after the time-out period is considered to be a new command or a SYNC request.

Note that whenever a partially issued command, or partially retrieved data, has occurred the time out in the serial communication is active. This means that if a time frame higher than 512 serial clock cycles is observed between two consecutive negative edges and the command being issued or data being retrieved is not complete, a soft-reset will occur causing the partially received command or data retrieved to be disregarded. The next negative edge in the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDM command, or the start of a SYNC request pulse.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x002E	DBGADHM	R W	Bit 15	14	13	12	11	10	9	Bit 8
0x002F	DBGADLM	R W	Bit 7	6	5	4	3	2	1	Bit 0

- ¹ This bit is visible at DBGCNT[7] and DBGSR[7]
² This represents the contents if the Comparator A control register is blended into this address.
³ This represents the contents if the Comparator B control register is blended into this address
⁴ This represents the contents if the Comparator C control register is blended into this address

Figure 6-2. Quick Reference to DBG Registers

6.3.2 Register Descriptions

This section consists of the DBG control and trace buffer register descriptions in address order. Each comparator has a bank of registers that are visible through an 8-byte window between 0x0028 and 0x002F in the DBG module register address map. When ARM is set in DBG1, the only bits in the DBG module registers that can be written are ARM, TRIG, and COMRV[1:0]

6.3.2.1 Debug Control Register 1 (DBG1)

Address: 0x0020

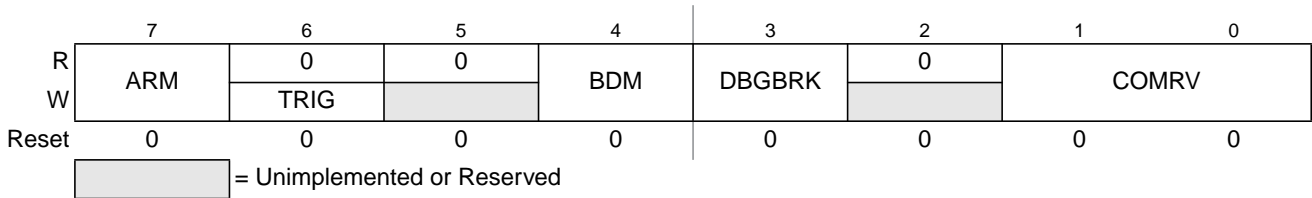


Figure 6-3. Debug Control Register (DBG1)

Read: Anytime

Write: Bits 7, 1, 0 anytime

Bit 6 can be written anytime but always reads back as 0.

Bits 4:3 anytime DBG is not armed.

NOTE

When disarming the DBG by clearing ARM with software, the contents of bits[4:3] are not affected by the write, since up until the write operation, ARM = 1 preventing these bits from being written. These bits must be cleared using a second write if required.

6.3.2.7.4 Debug Match Flag Register (DBGMFR)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	MC2	MC1	MC0
W								
Reset	0	0	0	0	0	0	0	0


 = Unimplemented or Reserved

Figure 6-12. Debug Match Flag Register (DBGMFR)

Read: If COMRV[1:0] = 11

Write: Never

DBGMFR is visible at 0x0027 only with COMRV[1:0] = 11. It features 3 flag bits each mapped directly to a channel. Should a match occur on the channel during the debug session, then the corresponding flag is set and remains set until the next time the module is armed by writing to the ARM bit. Thus the contents are retained after a debug session for evaluation purposes. These flags cannot be cleared by software, they are cleared only when arming the module. A set flag does not inhibit the setting of other flags. Once a flag is set, further comparator matches on the same channel in the same session have no affect on that flag.

6.3.2.8 Comparator Register Descriptions

Each comparator has a bank of registers that are visible through an 8-byte window in the DBG module register address map. Comparator A consists of 8 register bytes (3 address bus compare registers, two data bus compare registers, two data bus mask registers and a control register). Comparator B consists of four register bytes (three address bus compare registers and a control register). Comparator C consists of four register bytes (three address bus compare registers and a control register).

Each set of comparator registers can be accessed using the COMRV bits in the DBGIC1 register.

Unimplemented registers (e.g. Comparator B data bus and data bus masking) read as zero and cannot be written. The control register for comparator B differs from those of comparators A and C.

Table 6-21. Comparator Register Layout

0x0028	CONTROL	Read/Write	Comparators A,B and C
0x0029	ADDRESS HIGH	Read/Write	Comparators A,B and C
0x002A	ADDRESS MEDIUM	Read/Write	Comparators A,B and C
0x002B	ADDRESS LOW	Read/Write	Comparators A,B and C
0x002C	DATA HIGH COMPARATOR	Read/Write	Comparator A only
0x002D	DATA LOW COMPARATOR	Read/Write	Comparator A only
0x002E	DATA HIGH MASK	Read/Write	Comparator A only
0x002F	DATA LOW MASK	Read/Write	Comparator A only

6.3.2.8.1 Debug Comparator Control Register (DBGXCTL)

The contents of this register bits 7 and 6 differ depending upon which comparator registers are visible in the 8-byte window of the DBG module register address map.

Table 6-28. DBGADH Field Descriptions

Field	Description
7–0 Bits[15:8]	Comparator Data High Compare Bits — The Comparator data high compare bits control whether the selected comparator compares the data bus bits [15:8] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

6.3.2.8.6 Debug Comparator Data Low Register (DBGADL)

Address: 0x002D

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

Figure 6-20. Debug Comparator Data Low Register (DBGADL)

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

Table 6-29. DBGADL Field Descriptions

Field	Description
7–0 Bits[7:0]	Comparator Data Low Compare Bits — The Comparator data low compare bits control whether the selected comparator compares the data bus bits [7:0] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

6.3.2.8.7 Debug Comparator Data High Mask Register (DBGADHM)

Address: 0x002E

	7	6	5	4	3	2	1	0
R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
Reset	0	0	0	0	0	0	0	0

Figure 6-21. Debug Comparator Data High Mask Register (DBGADHM)

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

7.3.2.8 S12CPMU RTI Control Register (CPMURTI)

This register selects the time-out period for the Real Time Interrupt.

The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop Mode with PSTP=1 (Pseudo Stop Mode) and RTIOSCSEL=1 the RTI continues to run, else the RTI counter halts in Stop Mode.

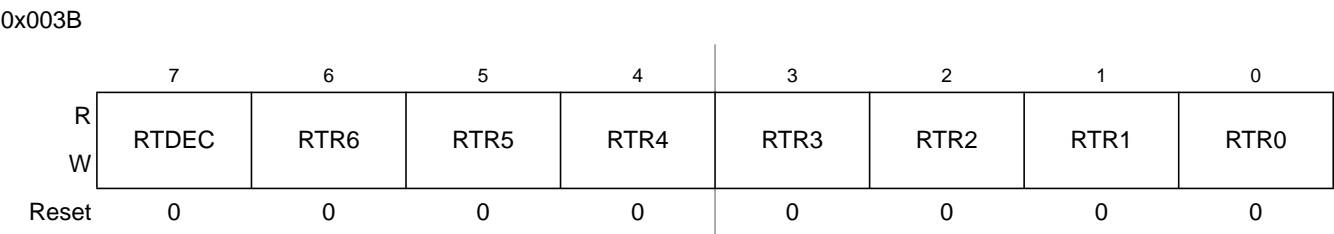


Figure 7-11. S12CPMU RTI Control Register (CPMURTI)

Read: Anytime

Write: Anytime

NOTE

A write to this register starts the RTI time-out period. A change of the RTIOSCSEL bit (writing a different value or loosing UPOSC status) re-starts the RTI time-out period.

Table 7-8. CPMURTI Field Descriptions

Field	Description
7 RTDEC	Decimal or Binary Divider Select Bit — RTDEC selects decimal or binary based prescaler values. 0 Binary based divider value. See Table 7-9 1 Decimal based divider value. See Table 7-10
6–4 RTR[6:4]	Real Time Interrupt Prescale Rate Select Bits — These bits select the prescale rate for the RTI. See Table 7-9 and Table 7-10.
3–0 RTR[3:0]	Real Time Interrupt Modulus Counter Select Bits — These bits select the modulus counter target value to provide additional granularity. Table 7-9 and Table 7-10 show all possible divide values selectable by the CPMURTI register.



7.3.2.21 S12CPMU Oscillator Register (CPMUOSC)

This registers configures the external oscillator (OSCLCP).

0x02FA



Figure 7-28. S12CPMU Oscillator Register (CPMUOSC)

Read: Anytime

Write: If PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register), then write anytime. Else write has no effect.

NOTE.

Write to this register clears the LOCK and UPOSC status bits.

NOTE.

If the chosen VCOCLK-to-OSCCLK ratio divided by two is not an integer number, then the filter can not be used and the OSCFILT[4:0] bits must be set to 0.

Table 7-22. CPMUOSC Field Descriptions

Field	Description
7 OSCE	Oscillator Enable Bit — This bit enables the external oscillator (OSCLCP). The UPOSC status bit in the CPMUFLG register indicates when the oscillation is stable and OSCCLK can be selected as Bus Clock or source of the COP or RTI. A loss of oscillation will lead to a clock monitor reset. 0 External oscillator is disabled. REFCLK for PLL is IRCCLK. 1 External oscillator is enabled.Clock monitor is enabled. REFCLK for PLL is the external oscillator clock divided by REFDIV. Note: When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit is already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator t_{UPOSC} before entering Pseudo Stop Mode.
6 OSCBW	Oscillator Filter Bandwidth Bit — If the VCOCLK frequency exceeds 25 MHz wide bandwidth must be selected.The Oscillator Filter is described in more detail at Section 7.4.5.2, "The Adaptive Oscillator Filter . 0 Oscillator filter bandwidth is narrow (window for expected OSCCLK edge is one VCOCLK cycle). 1 Oscillator filter bandwidth is wide (window for expected OSCCLK edge is three VCOCLK cycles).
4-0 OSCFILT	Oscillator Filter Bits — When using the oscillator a noise filter can be enabled, which filters noise from the OSCCLK and detects if the OSCCLK is qualified or not (quality status shown by bit UPOSC). The f_{VCO} -to- f_{OSC} ratio divided by two must be an integer value. The OSCFILT[4:0] bits must be set to the calculated integer value to enable the oscillator filter). 0x0000 Oscillator Filter disabled. else Oscillator Filter enabled:

Table 8-4. CANCTL1 Register Field Descriptions (continued)

Field	Description
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 8.4.5.5, “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}
1 SLPAK	Sleep Mode Acknowledge — This flag indicates whether the MSCAN module has entered sleep mode (see Section 8.4.5.5, “MSCAN Sleep Mode”). It is used as a handshake flag for the SLPRQ sleep mode request. Sleep mode is active when SLPRQ = 1 and SLPAK = 1. Depending on the setting of WUPE, the MSCAN will clear the flag if it detects activity on the CAN bus while in sleep mode. 0 Running — The MSCAN operates normally 1 Sleep mode active — The MSCAN has entered sleep mode
0 INITAK	Initialization Mode Acknowledge — This flag indicates whether the MSCAN module is in initialization mode (see Section 8.4.4.5, “MSCAN Initialization Mode”). It is used as a handshake flag for the INITRQ initialization mode request. Initialization mode is active when INITRQ = 1 and INITAK = 1. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0–CANIDAR7, and CANIDMR0–CANIDMR7 can be written only by the CPU when the MSCAN is in initialization mode. 0 Running — The MSCAN operates normally 1 Initialization mode active — The MSCAN has entered initialization mode

8.3.2.3 MSCAN Bus Timing Register 0 (CANBTR0)

The CANBTR0 register configures various CAN bus timing parameters of the MSCAN module.

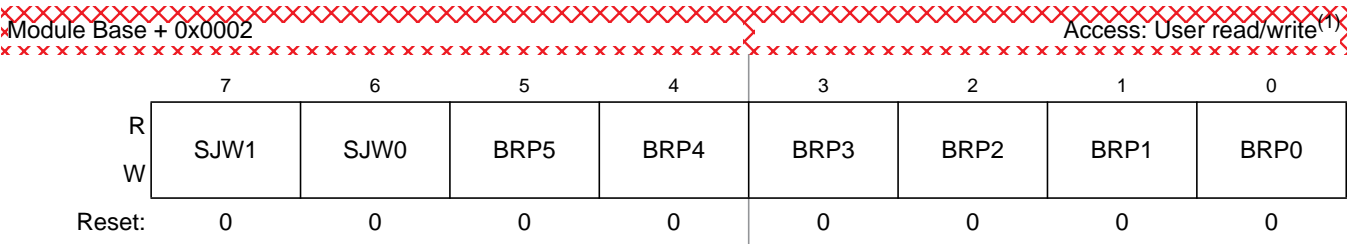


Figure 8-6. MSCAN Bus Timing Register 0 (CANBTR0)

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

Table 8-5. CANBTR0 Register Field Descriptions

Field	Description
7-6 SJW[1:0]	Synchronization Jump Width — The synchronization jump width defines the maximum number of time quanta (Tq) clock cycles a bit can be shortened or lengthened to achieve resynchronization to data transitions on the CAN bus (see Table 8-6).
5-0 BRP[5:0]	Baud Rate Prescaler — These bits determine the time quanta (Tq) clock which is used to build up the bit timing (see Table 8-7).

Table 8-6. Synchronization Jump Width

SJW1	SJW0	Synchronization Jump Width
0	0	1 Tq clock cycle

8.3.2.10 MSCAN Transmitter Message Abort Acknowledge Register (CANTAABK)

The CANTAABK register indicates the successful abort of a queued message, if requested by the appropriate bits in the CANTARQ register.

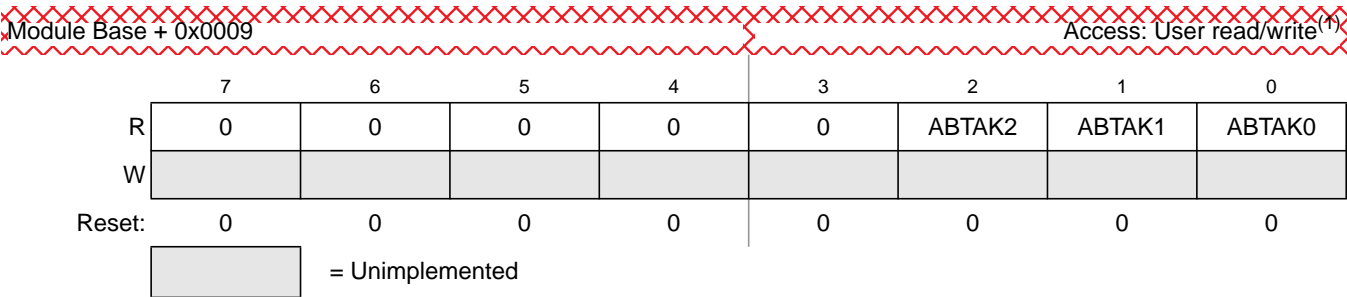


Figure 8-13. MSCAN Transmitter Message Abort Acknowledge Register (CANTAABK)

1. Read: Anytime
- Write: Unimplemented

NOTE

The CANTAABK register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1).

Table 8-16. CANTAABK Register Field Descriptions

Field	Description
2-0 ABTAK[2:0]	Abort Acknowledge — This flag acknowledges that a message was aborted due to a pending abort request from the CPU. After a particular message buffer is flagged empty, this flag can be used by the application software to identify whether the message was aborted successfully or was sent anyway. The ABTAKx flag is cleared whenever the corresponding TXE flag is cleared. 0 The message was not aborted. 1 The message was aborted.

8.3.2.11 MSCAN Transmit Buffer Selection Register (CANTBSEL)

The CANTBSEL register allows the selection of the actual transmit message buffer, which then will be accessible in the CANTXFG register space.

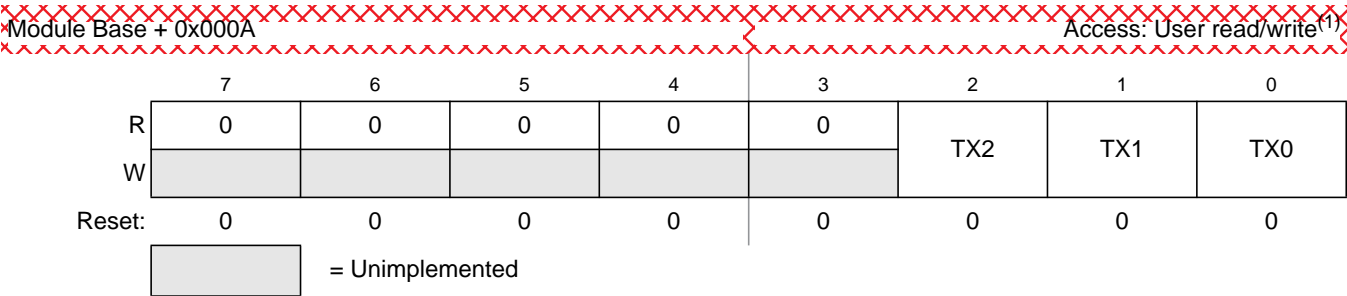


Figure 8-14. MSCAN Transmit Buffer Selection Register (CANTBSEL)

1. Read: Find the lowest ordered bit set to 1, all other bits will be read as 0
- Write: Anytime when not in initialization mode

9.3.2.3 ATD Control Register 2 (ATDCTL2)

Writes to this register will abort current conversion sequence.

Module Base + 0x0002

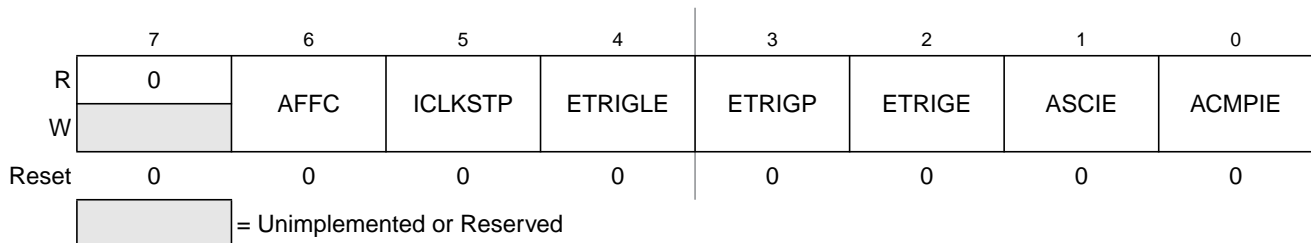


Figure 9-5. ATD Control Register 2 (ATDCTL2)

Read: Anytime

Write: Anytime

Table 9-6. ATDCTL2 Field Descriptions

Field	Description
6 AFFC	ATD Fast Flag Clear All 0 ATD flag clearing done by write 1 to respective CCF[n] flag. 1 Changes all ATD conversion complete flags to a fast clear sequence. For compare disabled (CMPE[n]=0) a read access to the result register will cause the associated CCF[n] flag to clear automatically. For compare enabled (CMPE[n]=1) a write access to the result register will cause the associated CCF[n] flag to clear automatically.
5 ICLKSTP	Internal Clock in Stop Mode Bit — This bit enables A/D conversions in stop mode. When going into stop mode and ICLKSTP=1 the ATD conversion clock is automatically switched to the internally generated clock ICLK. Current conversion sequence will seamless continue. Conversion speed will change from prescaled bus frequency to the ICLK frequency (see ATD Electrical Characteristics in device description). The prescaler bits PRS4-0 in ATDCTL4 have no effect on the ICLK frequency. For conversions during stop mode the automatic compare interrupt or the sequence complete interrupt can be used to inform software handler about changing A/D values. External trigger will not work while converting in stop mode. For conversions during transition from Run to Stop Mode or vice versa the result is not written to the results register, no CCF flag is set and no compare is done. When converting in Stop Mode (ICLKSTP=1) an ATD Stop Recovery time $t_{\text{ATDSTPRCV}}$ is required to switch back to bus clock based ATDCLK when leaving Stop Mode. Do not access ATD registers during this time. 0 If A/D conversion sequence is ongoing when going into stop mode, the actual conversion sequence will be aborted and automatically restarted when exiting stop mode. 1 A/D continues to convert in stop mode using internally generated clock (ICLK)
4 ETRIGLE	External Trigger Level/Edge Control — This bit controls the sensitivity of the external trigger signal. See Table 9-7 for details.
3 ETRIGP	External Trigger Polarity — This bit controls the polarity of the external trigger signal. See Table 9-7 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 9-5 . 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 the start of conversion with external events. External trigger will not work while converting in stop mode. 0 Disable external trigger 1 Enable external trigger

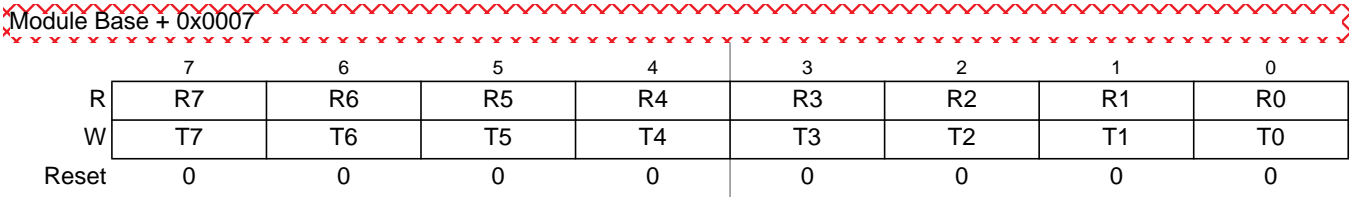


Figure 11-13. SCI Data Registers (SCIDRL)

Read: Anytime; reading accesses SCI receive data register

Write: Anytime; writing accesses SCI transmit data register; writing to R8 has no effect

Table 11-13. SCIDRH and SCIDRL Field Descriptions

Field	Description
SCIDRH 7 R8	Received Bit 8 — R8 is the ninth data bit received when the SCI is configured for 9-bit data format (M = 1).
SCIDRH 6 T8	Transmit Bit 8 — T8 is the ninth data bit transmitted when the SCI is configured for 9-bit data format (M = 1).
SCIDRL 7:0 R[7:0] T[7:0]	R7:R0 — Received bits seven through zero for 9-bit or 8-bit data formats T7:T0 — Transmit bits seven through zero for 9-bit or 8-bit formats

NOTE

If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten.The same value is transmitted until T8 is rewritten

In 8-bit data format, only SCI data register low (SCIDRL) needs to be accessed.

When transmitting in 9-bit data format and using 8-bit write instructions, write first to SCI data register high (SCIDRH), then SCIDRL.

11.4 Functional Description

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

Figure 11-14 shows the structure of the SCI module. The SCI allows full duplex, asynchronous, serial communication between the CPU and remote devices, including other CPUs. The SCI transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

11.5.3.1.8 BKDIF Description

The BKDIF interrupt is set when a break signal was received. Clear BKDIF by writing a “1” to the SCIASR1 SCI alternative status register 1. This flag is also cleared if break detect feature is disabled.

11.5.4 Recovery from Wait Mode

The SCI interrupt request can be used to bring the CPU out of wait mode.

11.5.5 Recovery from Stop Mode

An active edge on the receive input can be used to bring the CPU out of stop mode.

Table 13-17. P-Flash Protection Function

FPOPEN	FPHDIS	FPLDIS	Function ⁽¹⁾
1	1	1	No P-Flash Protection
1	1	0	Protected Low Range
1	0	1	Protected High Range
1	0	0	Protected High and Low Ranges
0	1	1	Full P-Flash Memory Protected
0	1	0	Unprotected Low Range
0	0	1	Unprotected High Range
0	0	0	Unprotected High and Low Ranges

1. For range sizes, refer to [Table 13-18](#) and [Table 13-19](#).

Table 13-18. P-Flash Protection Higher Address Range

FPHS[1:0]	Global Address Range	Protected Size
00	0x3_F800–0x3_FFFF	2 Kbytes
01	0x3_F000–0x3_FFFF	4 Kbytes
10	0x3_E000–0x3_FFFF	8 Kbytes
11	0x3_C000–0x3_FFFF	16 Kbytes

Table 13-19. P-Flash Protection Lower Address Range

FPLS[1:0]	Global Address Range	Protected Size
00	0x3_8000–0x3_83FF	1 Kbyte
01	0x3_8000–0x3_87FF	2 Kbytes
10	0x3_8000–0x3_8FFF	4 Kbytes
11	0x3_8000–0x3_9FFF	8 Kbytes

All possible P-Flash protection scenarios are shown in [Figure 13-14](#). Although the protection scheme is loaded from the Flash memory at global address 0x3_FF0C during the reset sequence, it can be changed by the user. The P-Flash protection scheme can be used by applications requiring reprogramming in single chip mode while providing as much protection as possible if reprogramming is not required.

Table 13-55. Set User Margin Level 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 13-27)
		Set if an invalid global address [17:16] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

13.4.5.13 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of the P-Flash or D-Flash block.

Table 13-56. Set Field Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Global address [17:16] to identify the Flash block
001	Margin level setting	

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the field margin level for the targeted block and then set the CCIF flag.

NOTE

When the D-Flash block is targeted, the D-Flash field margin levels are applied only to the D-Flash reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and D-Flash reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in [Table 13-57](#).

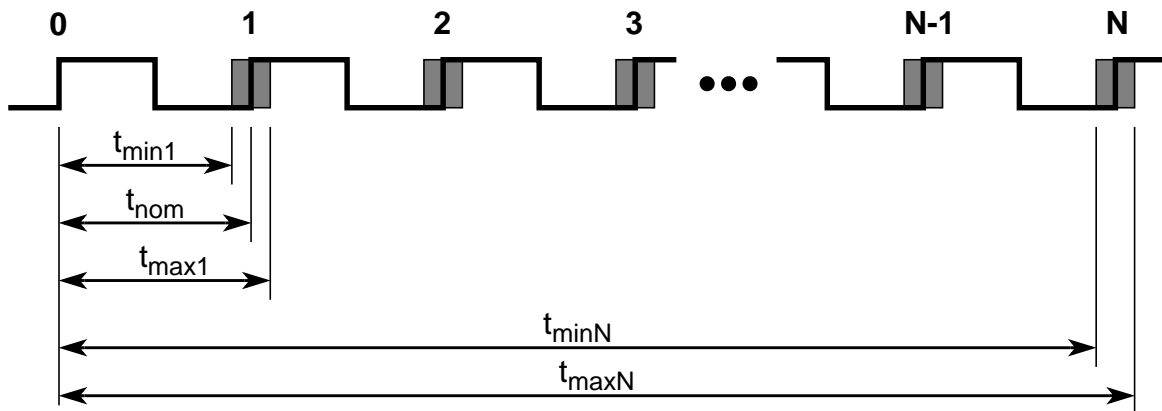


Figure A-2. Jitter Definitions

The relative deviation of t_{nom} is at its maximum for one clock period, and decreases towards zero for larger number of clock periods (N).

Defining the jitter as:

$$J(N) = \max\left(\left|1 - \frac{t_{max}^{(N)}}{N \cdot t_{nom}}\right|, \left|1 - \frac{t_{min}^{(N)}}{N \cdot t_{nom}}\right|\right)$$

For $N < 100$, the following equation is a good fit for the maximum jitter:

$$J(N) = \frac{j_1}{\sqrt{N}}$$

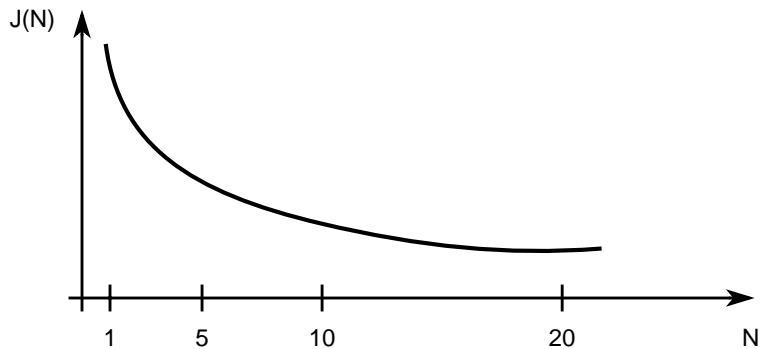


Figure A-3. Maximum Bus Clock Jitter Approximation

NOTE

On timers and serial modules a prescaler will eliminate the effect of the jitter to a large extent.