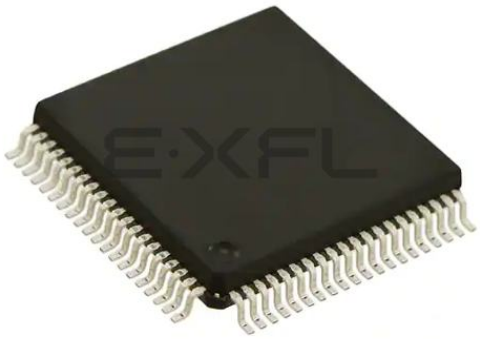

Welcome to E-XFL.COM

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	59
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs128j1caar

Appendix A
Electrical Characteristics

Appendix B
Package Information

Appendix C
PCB Layout Guidelines

2.3.35 Port M Input Register (PTIM)

Address 0x0251 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIM7	PTIM6	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
W								
Reset	u	u	u	u	u	u	u	u

= Unimplemented or Reserved u = Unaffected by reset

Figure 2-33. Port M Input Register (PTIM)

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

Table 2-31. PTIM Register Field Descriptions

Field	Description
7-0 PTIM	**Port M 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.

2.3.36 Port M Data Direction Register (DDRM)

Address 0x0252 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRM7	DDRM6	DDRM5	DDRM4	DDRM3	DDRM2	DDRM1	DDRM0
Reset	0	0	0	0	0	0	0	0

Figure 2-34. Port M Data Direction Register (DDRM)

[1] Read: Anytime
Write: Anytime

Table 2-39. PTP Register Field Descriptions

Field	Description
7 PTP	**Port P general purpose input/output data**—Data Register, PWM input/output, pin interrupt input/output When not used with the alternative function, 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. • The PWM function takes precedence over the general purpose I/O function if the related channel or the emergency shut-down feature is enabled. • Pin interrupts can be generated if enabled in input or output mode.
6-3 PTP	**Port P general purpose input/output data**—Data Register, PWM output, pin interrupt input/output When not used with the alternative function, 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. • The PWM function takes precedence over the general purpose I/O function if the related channel is enabled. • Pin interrupts can be generated if enabled in input or output mode.
2 PTP	**Port P general purpose input/output data**—Data Register, PWM output, routed TIM output, routed SCI1 TXD output, pin interrupt input/output When not used with the alternative function, 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. • The PWM function takes precedence over the TIM, SCI1 and general purpose I/O function if the related channel is enabled. • The TIM function takes precedence over SCI1 and the general purpose I/O function if the related channel is enabled. • The SCI1 function takes precedence over the general purpose I/O function if enabled. • Pin interrupts can be generated if enabled in input or output mode.

5.3.2.4 BDM Global Page Index Register (BDMGPR)

Register Global Address 0x7FFF08

	7	6	5	4	3	2	1	0
R W	BGAE	BGP6	BGP5	BGP4	BGP3	BGP2	BGP1	BGP0
Reset	0	0	0	0	0	0	0	0

Figure 5-6. BDM Global Page Register (BDMGPR)

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

Table 5-5. BDMGPR Field Descriptions

Field	Description
7 BGAE	**BDM Global Page Access Enable Bit** — BGAE enables global page access for BDM hardware and firmware read/write instructions The BDM hardware commands used to access the BDM registers (READ_BD_ and WRITE_BD_) can not be used for global accesses even if the BGAE bit is set. 0 BDM Global Access disabled 1 BDM Global Access enabled
6–0 BGP[6:0]	**BDM Global Page Index Bits 6–0** — These bits define the extended address bits from 22 to 16. For more detailed information regarding the global page window scheme, please refer to the S12X_MMC Block Guide.

5.3.3 Family ID Assignment

The family ID is a 8-bit value located in the firmware ROM (at global address: 0x7FFF0F). The read-only value is a unique family ID which is 0xC1 for S12X devices.

5.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands: hardware and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see Section 5.4.3, “BDM Hardware Commands”. Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see Section 5.4.4, “Standard BDM Firmware Commands”. The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted (see Section 5.4.3, “BDM Hardware Commands”) and in secure mode (see Section 5.4.1, “Security”). Firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

Table 6-4. DBGC1 Field Descriptions (continued)

Field	Description
3 DBGBRK	**S12XDBG Breakpoint Enable Bit** — The DBGBRK bit controls whether the debugger will request a breakpoint to S12XCPU upon reaching the state sequencer Final State. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. Please refer to Section 6.4.7 for further details. 0 No breakpoint on trigger. 1 Breakpoint on trigger
1–0 COMRV	**Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the S12XDBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which register is visible at the address 0x0027. See Table 6-5.

Table 6-5. COMRV Encoding

COMRV	Visible Comparator	Visible Register at 0x0027
00	Comparator A	DBGSCR1
01	Comparator B	DBGSCR2
10	Comparator C	DBGSCR3
11	Comparator D	DBGMFR

6.3.2.2 Debug Status Register (DBGSR)

Address: 0x0021

	7	6	5	4	3	2	1	0
R	TBF	0	0	0	0	SSF2	SSF1	SSF0
W								
Reset	—	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 6-4. Debug Status Register (DBGSR)

Read: Anytime

Write: Never

Table 6-6. DBGSR Field Descriptions

Field	Description
7 TBF	**Trace Buffer Full** — The TBF bit indicates that the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines will be valid data, regardless of the value of DBGCNT bits CNT[6:0]. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit.
2–0 SSF[2:0]	**State Sequencer Flag Bits** — The SSF bits indicate in which state the State Sequencer is currently in. During a debug session on each transition to a new state these bits are updated. If the debug session is ended by software clearing the ARM bit, then these bits retain their value to reflect the last state of the state sequencer before disarming. If a debug session is ended by an internal trigger, then the state sequencer returns to state0 and these bits are cleared to indicate that state0 was entered during the session. On arming the module the state sequencer enters state1 and these bits are forced to SSF[2:0] = 001. See Table 6-7.

6.3.2.8.3 Debug Comparator Address Mid Register (DBGXAM)

Address: 0x002A

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

Figure 6-16. Debug Comparator Address Mid Register (DBGXAM)

Read: Anytime. See Table 6-26 for visible register encoding.

Write: If DBG not armed. See Table 6-26 for visible register encoding.

Table 6-30. DBGXAM Field Descriptions

Field	Description
7–0 Bit[15:8]	**Comparator Address Mid Compare Bits**— The Comparator address mid compare bits control whether the selected comparator will compare the address bus bits [15:8] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

6.3.2.8.4 Debug Comparator Address Low Register (DBGXAL)

Address: 0x002B

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

Figure 6-17. Debug Comparator Address Low Register (DBGXAL)

Read: Anytime. See Table 6-26 for visible register encoding.

Write: If DBG not armed. See Table 6-26 for visible register encoding.

Table 6-31. DBGXAL Field Descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Address Low Compare Bits** — The Comparator address low compare bits control whether the selected comparator will compare the address bus bits [7:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

6.4.1 S12XDBG Operation

Arming the S12XDBG module by setting ARM in DBGC1 allows triggering, and storing of data in the trace buffer and can be used to cause breakpoints to the CPU12X . The DBG module is made up of four main blocks, the comparators, control logic, the state sequencer, and the trace buffer.

The comparators monitor the bus activity of the CPU12X . Comparators can be configured to monitor address and databus. Comparators can also be configured to mask out individual data bus bits during a compare and to use R/W and word/byte access qualification in the comparison. When a match with a comparator register value occurs the associated control logic can trigger the state sequencer to another state (see Figure 6-22). Either forced or tagged triggers are possible. Using a forced trigger, the trigger is generated immediately on a comparator match. Using a tagged trigger, at a comparator match, the instruction opcode is tagged and only if the instruction reaches the execution stage of the instruction queue is a trigger generated. In the case of a transition to Final State, bus tracing is triggered and/or a breakpoint can be generated.

Independent of the state sequencer, a breakpoint can be triggered by writing to the TRIG bit in the DBGC1 control register.

The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads.

6.4.2 Comparator Modes

The S12XDBG contains four comparators, A, B, C, and D. Each comparator compares the selected address bus with the address stored in DBGXAH, DBGXAM, and DBGXAL. Furthermore, comparators A and C also compare the data buses to the data stored in DBGXDH, DBGXDL and allow masking of individual data bus bits.

S12X comparator matches are disabled in BDM and during BDM accesses.

The comparator match control logic configures comparators to monitor the buses for an exact address or an address range. The comparator configuration is controlled by the control register contents and the range control by the DBGC2 contents.

On a match a trigger can initiate a transition to another state sequencer state (see Section 6.4.3”). The comparator control register also allows the type of access to be included in the comparison through the use of the RWE, RW, SZE, and SZ bits. The RWE bit controls whether read or write comparison is enabled for the associated comparator and the RW bit selects either a read or write access for a valid match. Similarly the SZE and SZ bits allows the size of access (word or byte) to be considered in the compare. Only comparators B and D feature SZE and SZ.

The TAG bit in each comparator control register is used to determine the triggering condition. By setting TAG, the comparator will qualify a match with the output of opcode tracking logic and a trigger occurs before the tagged instruction executes (tagged-type trigger). Whilst tagging, the RW, RWE, SZE, and SZ bits are ignored and the comparator register must be loaded with the exact opcode address.

If the TAG bit is clear (forced type trigger) a comparator match is generated when the selected address appears on the system address bus. If the selected address is an opcode address, the match is generated

be executed then the trace is continued for another 32 lines. Upon tracing completion the breakpoint is generated, thus the breakpoint does not occur at the tagged instruction boundary.

6.4.5.1.3 Storing with End-Trigger

Storing with End-Trigger, data is stored in the Trace Buffer until the Final State is entered, at which point the S12XDBG module will become disarmed and no more data will be stored. If the trigger is at the address of a change of flow instruction the trigger event will not be stored in the Trace Buffer.

6.4.5.2 Trace Modes

The S12XDBG module can operate in four trace modes. The mode is selected using the TRCMOD bits in the DBGTCR register. The modes are described in the following subsections. The trace buffer organization is shown in Table 6-40.

6.4.5.2.1 Normal Mode

In Normal Mode, change of flow (COF) program counter (PC) addresses will be stored.

COF addresses are defined as follows :

- Source address of taken conditional branches (long, short, bit-conditional, and loop primitives)
- Destination address of indexed JMP, JSR, and CALL instruction
- Destination address of RTI, RTS, and RTC instructions.
- Vector address of interrupts, except for SWI and BDM vectors

LBRA, BRA, BSR, BGND as well as non-indexed JMP, JSR, and CALL instructions are not classified as change of flow and are not stored in the trace buffer.

Change-of-flow addresses stored include the full 23-bit address bus of CPU12X and an information byte, which contains a source/destination bit to indicate whether the stored address was a source address or destination address.

NOTE

When an CPU12X COF instruction with destination address is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets exectuted after the interrupt service routine.

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ; IRQ interrupt occurs during execution of this
MARK2   NOP                             ;
```

- System Reset generation from the following possible sources:
  - Power on reset
  - Low voltage reset
  - Illegal address reset
  - COP reset
  - Loss of clock reset
  - External pin reset
- Real-Time Interrupt (RTI)

8.1.2 Modes of Operation

This subsection lists and briefly describes all operating modes supported by the S12XECRG.

- Run Mode

  All functional parts of the S12XECRG are running during normal Run Mode. If RTI or COP functionality is required the individual bits of the associated rate select registers (COPCTL, RTICTL) have to be set to a non zero value.
- Wait Mode

  In this mode the IPLL can be disabled automatically depending on the PLLWAI bit.
- Stop Mode

  Depending on the setting of the PSTP bit Stop Mode can be differentiated between Full Stop Mode (PSTP = 0) and Pseudo Stop Mode (PSTP = 1).
  - Full Stop Mode

    The oscillator is disabled and thus all system and core clocks are stopped. The COP and the RTI remain frozen.
  - Pseudo Stop Mode

    The oscillator continues to run and most of the system and core clocks are stopped. If the respective enable bits are set the COP and RTI will continue to run, else they remain frozen.
- Self Clock Mode

  Self Clock Mode will be entered if the Clock Monitor Enable Bit (CME) and the Self Clock Mode Enable Bit (SCME) are both asserted and the clock monitor in the oscillator block detects a loss of clock. As soon as Self Clock Mode is entered the S12XECRG starts to perform a clock quality check. Self Clock Mode remains active until the clock quality check indicates that the required quality of the incoming clock signal is met (frequency and amplitude). Self Clock Mode should be used for safety purposes only. It provides reduced functionality to the MCU in case a loss of clock is causing severe system conditions.

8.1.3 Block Diagram

Figure 8-1 shows a block diagram of the S12XECRG.

10.3.2.12 ATD Conversion Result Registers (ATDDR*n*)

The A/D conversion results are stored in 16 result registers. Results are always in unsigned data representation. Left and right justification is selected using the DJM control bit in ATDCTL3.

If automatic compare of conversions results is enabled (CMPE[*n*]=1 in ATDCMPE), these registers must be written with the compare values in left or right justified format depending on the actual value of the DJM bit. In this case, as the ATDDR*n* register is used to hold the compare value, the result will not be stored there at the end of the conversion but is lost.

Read: Anytime

Write: Anytime

NOTE

For conversions not using automatic compare, results are stored in the result registers after each conversion. In this case avoid writing to ATDDRn except for initial values, because an A/D result might be overwritten.

10.3.2.12.1 Left Justified Result Data (DJM=0)

Module Base +
0x0010 = ATDDR0, 0x0012 = ATDDR1, 0x0014 = ATDDR2, 0x0016 = ATDDR3
0x0018 = ATDDR4, 0x001A = ATDDR5, 0x001C = ATDDR6, 0x001E = ATDDR7
0x0020 = ATDDR8, 0x0022 = ATDDR9, 0x0024 = ATDDR10, 0x0026 = ATDDR11
0x0028 = ATDDR12, 0x002A = ATDDR13, 0x002C = ATDDR14, 0x002E = ATDDR15

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

Figure 10-15. Left justified ATD conversion result register (ATDDR*n*)

10.3.2.12.2 Right Justified Result Data (DJM=1)

Module Base +
0x0010 = ATDDR0, 0x0012 = ATDDR1, 0x0014 = ATDDR2, 0x0016 = ATDDR3
0x0018 = ATDDR4, 0x001A = ATDDR5, 0x001C = ATDDR6, 0x001E = ATDDR7
0x0020 = ATDDR8, 0x0022 = ATDDR9, 0x0024 = ATDDR10, 0x0026 = ATDDR11
0x0028 = ATDDR12, 0x002A = ATDDR13, 0x002C = ATDDR14, 0x002E = ATDDR15

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bi1 1	Bit 0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-16. Right justified ATD conversion result register (ATDDR*n*)

Table 10-16 shows how depending on the A/D resolution the conversion result is transferred to the ATD result registers. Compare is always done using all 12 bits of both the conversion result and the compare value in ATDDRn.

When the transmit shift register is not transmitting a frame, the TXD pin goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCICR2), the transmitter enable signal goes low and the transmit signal goes idle.

If software clears TE while a transmission is in progress (TC = 0), the frame in the transmit shift register continues to shift out. To avoid accidentally cutting off the last frame in a message, always wait for TDRE to go high after the last frame before clearing TE.

To separate messages with preambles with minimum idle line time, use this sequence between messages:

1. Write the last byte of the first message to SCIDRH/L.
2. Wait for the TDRE flag to go high, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting the TE bit.
4. Write the first byte of the second message to SCIDRH/L.

14.4.5.3 Break Characters

Writing a logic 1 to the send break bit, SBK, in SCI control register 2 (SCICR2) loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCI control register 1 (SCICR1). As long as SBK is at logic 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next frame.

The SCI recognizes a break character when there are 10 or 11(M = 0 or M = 1) consecutive zero received. Depending if the break detect feature is enabled or not receiving a break character has these effects on SCI registers.

If the break detect feature is disabled (BKDFE = 0):

- Sets the framing error flag, FE
- Sets the receive data register full flag, RDRF
- Clears the SCI data registers (SCIDRH/L)
- May set the overrun flag, OR, noise flag, NF, parity error flag, PE, or the receiver active flag, RAF (see 3.4.4 and 3.4.5 SCI Status Register 1 and 2)

If the break detect feature is enabled (BKDFE = 1) there are two scenarios[1]

The break is detected right from a start bit or is detected during a byte reception.

- Sets the break detect interrupt flag, BKDIF
- Does not change the data register full flag, RDRF or overrun flag OR
- Does not change the framing error flag FE, parity error flag PE.
- Does not clear the SCI data registers (SCIDRH/L)
- May set noise flag NF, or receiver active flag RAF.

1. A Break character in this context are either 10 or 11 consecutive zero received bits

15.3.2.5 SPI Data Register (SPIDR = SPIDRH:SPIDRL)

Module Base +0x0004

	7	6	5	4	3	2	1	0
R	R15	R14	R13	R12	R11	R10	R9	R8
W	T15	T14	T13	T12	T11	T10	T9	T8
Reset	0	0	0	0	0	0	0	0

Figure 15-7. SPI Data Register High (SPIDRH)

Module Base +0x0005

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Figure 15-8. SPI Data Register Low (SPIDRL)

Read: Anytime; read data only valid when SPIF is set

Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows data to be queued and transmitted. For an SPI configured as a master, queued data is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data.

Received data in the SPIDR is valid when SPIF is set.

If SPIF is cleared and data has been received, the received data is transferred from the receive shift register to the SPIDR and SPIF is set.

If SPIF is set and not serviced, and a second data value has been received, the second received data is kept as valid data in the receive shift register until the start of another transmission. The data in the SPIDR does not change.

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced before the start of a third transmission, the data in the receive shift register is transferred into the SPIDR and SPIF remains set (see Figure 15-9).

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced after the start of a third transmission, the data in the receive shift register has become invalid and is not transferred into the SPIDR (see Figure 15-10).

Table 18-20. P-Flash Protection Lower Address Range

FPLS[1:0]	Global Address Range	Protected Size
00	0x7F_8000–0x7F_83FF	1 Kbyte
01	0x7F_8000–0x7F_87FF	2 Kbytes
10	0x7F_8000–0x7F_8FFF	4 Kbytes
11	0x7F_8000–0x7F_9FFF	8 Kbytes

All possible P-Flash protection scenarios are shown in Figure 18-14. Although the protection scheme is loaded from the Flash memory at global address 0x7F_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 19-29. P-Flash Commands

FCMD	Command	Function on P-Flash Memory
0x01	Erase Verify All Blocks	Verify that all P-Flash (and D-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that a P-Flash block is erased.
0x03	Erase Verify P-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x04	Read Once	Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block 0 that was previously programmed using the Program Once command.
0x06	Program P-Flash	Program a phrase in a P-Flash block.
0x07	Program Once	Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block 0 that is allowed to be programmed only once.
0x08	Erase All Blocks	Erase all P-Flash (and D-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a P-Flash (or D-Flash) block. An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command.
0x0A	Erase P-Flash Sector	Erase all bytes in a P-Flash sector.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all P-Flash (and D-Flash) blocks and verifying that all P-Flash (and D-Flash) blocks are erased.
0x0C	Verify Backdoor Access Key	Supports a method of releasing MCU security by verifying a set of security keys.
0x0D	Set User Margin Level	Specifies a user margin read level for all P-Flash blocks.
0x0E	Set Field Margin Level	Specifies a field margin read level for all P-Flash blocks (special modes only).

19.4.1.5 D-Flash Commands

Table 19-30 summarizes the valid D-Flash commands along with the effects of the commands on the D-Flash block.

Table 19-30. D-Flash Commands

FCMD	Command	Function on D-Flash Memory
0x01	Erase Verify All Blocks	Verify that all D-Flash (and P-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that the D-Flash block is erased.
0x08	Erase All Blocks	Erase all D-Flash (and P-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a D-Flash (or P-Flash) block. An erase of the full D-Flash block is only possible when DPOPEN bit in the DFPROT register is set prior to launching the command.

Table 19-59. Erase Verify D-Flash Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x10	Global address [22:16] to identify the D-Flash block
001	Global address [15:0] of the first word to be verified	
010	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify D-Flash Section command, the Memory Controller will verify the selected section of D-Flash memory is erased. The CCIF flag will set after the Erase Verify D-Flash Section operation has completed.

Table 19-60. Erase Verify D-Flash Section Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 19-28)
		Set if an invalid global address [22:0] is supplied
		Set if a misaligned word address is supplied (global address [0] != 0)
		Set if the requested section breaches the end of the D-Flash block
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

19.4.2.15 Program D-Flash Command

The Program D-Flash operation programs one to four previously erased words in the D-Flash block. The Program D-Flash operation will confirm that the targeted location(s) were successfully programmed upon completion.

CAUTION

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

Table 19-61. Program D-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x11	Global address [22:16] to identify the D-Flash block
001	Global address [15:0] of word to be programmed	
010	Word 0 program value	
011	Word 1 program value, if desired	
100	Word 2 program value, if desired	

Table 20-17. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 20-18 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x7F_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 20-19. The FPHS bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x7F_8000. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 20-20. The FPLS bits can only be written to while the FPLDIS bit is set.

Table 20-18. 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 20-19 and Table 20-20.

Table 20-19. P-Flash Protection Higher Address Range

FPHS[1:0]	Global Address Range	Protected Size
00	0x7F_F800–0x7F_FFFF	2 Kbytes
01	0x7F_F000–0x7F_FFFF	4 Kbytes
10	0x7F_E000–0x7F_FFFF	8 Kbytes
11	0x7F_C000–0x7F_FFFF	16 Kbytes

Table 20-20. P-Flash Protection Lower Address Range

FPLS[1:0]	Global Address Range	Protected Size
00	0x7F_8000–0x7F_83FF	1 Kbyte
01	0x7F_8000–0x7F_87FF	2 Kbytes
10	0x7F_8000–0x7F_8FFF	4 Kbytes
11	0x7F_8000–0x7F_9FFF	8 Kbytes

All possible P-Flash protection scenarios are shown in Figure 20-14. Although the protection scheme is loaded from the Flash memory at global address 0x7F_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 A-13. Pseudo Stop and Full Stop Current

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Pseudo stop current (API, RTI, and COP disabled) PLL off, LCP mode							
10a	C	–40°C	I_{DDPS}	—	155	300	µA
	C	27°C		—	171	400	
	C	70°C		—	199	—	
	C	85°C		—	216	—	
	C	105°C		—	233	—	
	C	110°C		—	270	—	
	C	130°C		—	350	—	
	C	150°C		—	452	—	
Pseudo stop current (API, RTI, and COP disabled) PLL off, FSP mode							
10b	P	–40°C	I_{DDPS}	—	60	80	µA
	P	27°C		—	70	100	
	P	110°C		—	160	2400	
	P	130°C		—	210	2400	
	P	150°C		—	400	2400	
Pseudo stop current (API, RTI, and COP enabled) PLL off, LCP mode							
11	C	27°C	I_{DDPS}	—	186	—	µA
	C	70°C		—	209	—	
	C	85°C		—	245	—	
	C	105°C		—	270	—	
	C	125°C		—	383	—	
	C	150°C		—	487	—	
Stop Current							
12	P	–40°C	I_{DDS}	—	20	60	µA
	P	27°C		—	25	80	
	C	70°C		—	40	—	
	C	85°C		—	65	—	
	C	105°C		—	80	—	
	C	110°C		—	95	—	
	C	125°C		—	220	—	
	C	130°C		—	250	—	
	P	150°C		—	380	2000	
Stop Current (API active)							
13	T	–40°C	I_{DDS}	—	25	—	µA
	T	27°C		—	40	—	
	T	85°C		—	70	—	
	T	110°C		—	100	—	
	T	130°C		—	255	—	
Stop Current (ATD active)							
14	T	27°C	I_{DDS}	—	190	—	µA
	T	85°C		—	230	—	
	T	125°C		—	400	—	

A.7 MSCAN

Table A-25. MSCAN Wake-up Pulse Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	MSCAN wakeup dominant pulse filtered	t_{WUP}	—	—	1.5	µs
2	P	MSCAN wakeup dominant pulse pass	t_{WUP}	5	—	—	µs

0x02C0–0x02EF Analog-to-Digital Converter 12-Bit 16-Channel (ATD0) Map (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02CF	ATD0CMPHTL	R	CMPHT7	CMPHT6	CMPHT5	CMPHT4	CMPHT3	CMPHT2	CMPHT1	CMPHT0
		W								
0x02D0	ATD0DR0H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02D1	ATD0DR0L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02D2	ATD0DR1H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02D3	ATD0DR1L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02D4	ATD0DR2H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02D5	ATD0DR2L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02D6	ATD0DR3H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02D7	ATD0DR3L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02D8	ATD0DR4H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02D9	ATD0DR4L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02DA	ATD0DR5H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02DB	ATD0DR5L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02DC	ATD0DR6H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02DD	ATD0DR6L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02DE	ATD0DR7H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02DF	ATD0DR7L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02E0	ATD0DR8H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02E1	ATD0DR8L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02E2	ATD0DR9H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02E3	ATD0DR9L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02E4	ATD0DR10H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02E5	ATD0DR10L	R	Bit7	Bit6	0	0	0	0	0	0
		W								