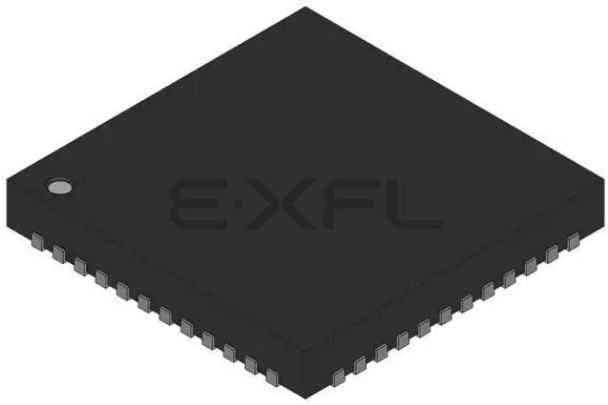


Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

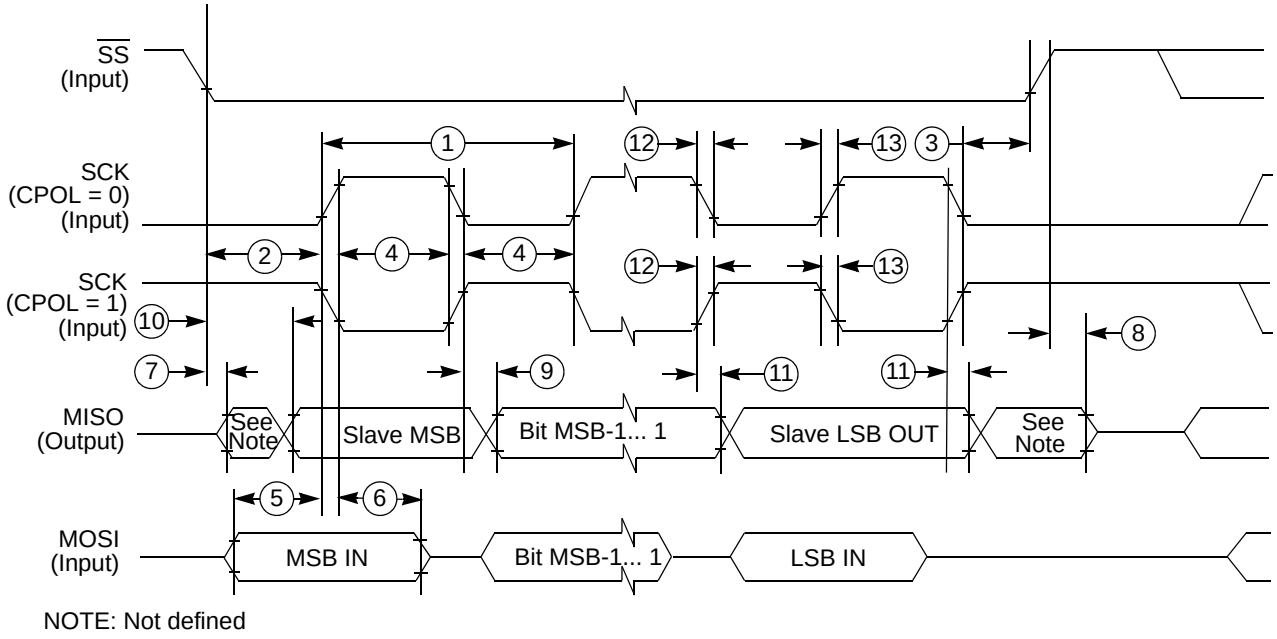
Details	
Product Status	Active
Applications	-
Core Processor	-
Program Memory Type	-
Controller Series	-
RAM Size	-
Interface	-
Number of I/O	-
Voltage - Supply	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mm912j637am2epr2

Table 45. SPI Master Mode Timing Characteristics

Num	C	Characteristic	Symbol	Min	Typ	Max	Unit
1	D	SCK Frequency	f_{SCK}	1/2048	—	1/2	f_{BUS}
1	D	SCK Period	t_{SCK}	2.0	—	2048	t_{BUS}
2	D	Enable Lead Time	t_{LEAD}	—	1/2	—	t_{SCK}
3	D	Enable Lag Time	t_{LAG}	—	1/2	—	t_{SCK}
4	D	Clock (SCK) High or Low Time	t_{WSCK}	—	1/2	—	t_{SCK}
5	D	Data Setup Time (inputs)	t_{SU}	8.0	—	—	ns
6	D	Data Hold Time (inputs)	t_{HI}	8.0	—	—	ns
9	D	Data Valid After SCK Edge	t_{VSCK}	—	—	29	ns
10	D	Data Valid After $\overline{SS}$ Fall (CPHA = 0)	t_{VSS}	—	—	15	ns
11	D	Data Hold Time (outputs)	t_{HO}	20	—	—	ns
12	D	Rise and Fall Time Inputs	t_{RFI}	—	—	8.0	ns
13	D	Rise and Fall Time Outputs	t_{RFO}	—	—	8.0	ns

4.6.2.5.2 Slave Mode

The timing diagram for slave mode with transmission format CPHA = 0 is depicted in Figure 10.



NOTE: Not defined

Figure 10. SPI Slave Timing (CPHA = 0)

The timing diagram for slave mode with transmission format CPHA = 1 is depicted in Figure 11.

Table 63. Analog die Registers - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3

Offset[71]	Name		15 7	14 6	13 5	12 4	11 3	10 2	9 1	8 0
0x13	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x14	WD_RR	R	WDR[7:0]							
	Watchdog rearm register	W								
0x15	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x16	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x17	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x18	SCIBD (hi)	R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
	SCI Baud Rate Register	W								
0x19	SCIBD (lo)	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	SCI Baud Rate Register	W								
0x1A	SCIC1	R	LOOPS	0	RSRC	M	0	ILT	PE	PT
	SCI Control Register 1	W								
0x1B	SCIC2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	SCI Control Register 2	W								
0x1C	SCIS1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	SCI Status Register 1	W								
0x1D	SCIS2	R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
	SCI Status Register 2	W								
0x1E	SCIC3	R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
	SCI Control Register 3	W								
0x1F	SCID	R	R7	R6	R5	R4	R3	R2	R1	R0
	SCI Data Register	W	T7	T6	T5	T4	T3	T2	T1	T0
0x20	TIOS	R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
	Timer Input Capture/Output Compare Select	W								
0x21	CFORC	R	0	0	0	0	0	0	0	0
	Timer Compare Force Register	W					FOC3	FOC2	FOC1	FOC0
0x22	OC3M	R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
	Output Compare 3 Mask Register	W								
0x23	OC3D	R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
	Output Compare 3 Data Register	W								
0x24	TCNT (hi)	R	TCNT[15:0]							
	Timer Count Register	W								
0x25	TCNT (lo)	R								
	Timer Count Register	W								

Table 87. Interrupt Sources

IRQ	Description
CVMI	Current / Voltage Measurement Interrupt
LTC	Lifetime Counter Interrupt
CAL	Calibration Request Interrupt

5.3.4.1 Under-voltage Interrupt (UVI)

This maskable interrupt signalizes a under-voltage condition on the VSUP supply input.

Acknowledge the interrupt by writing a 1 into the UVF Bit in the PCR Status Register (PCR_SR (hi)). The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. The UVF Bit represents the current condition, and might not be set after an interrupt was signalized by the interrupt source registers.

See Section 5.2, “Analog Die - Power, Clock and Resets - PCR" for details on the PCR Status Register (PCR_SR (hi)), including masking information.

NOTE

The under-voltage interrupt is not active in devices with the Cranking mode enabled. For those devices, the under-voltage threshold is used to enable the high precision low voltage threshold during Stop/Sleep mode.

Once the device wakes up from cranking mode, the UVI flag is indicating the wake-up source.

5.3.4.2 High Temperature Interrupt (HTI)

This maskable interrupt signalizes a high temperature condition on the analog die. The sensing element is located close to the major thermal contributors, the system voltage regulators.

Acknowledge the interrupt by writing a 1 into the HTF Bit in the PCR Status Register (PCR_SR (hi)). The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. The HTF Bit represents the current condition and might not be set after an interrupt was signalized by the interrupt source registers.

See Section 5.2, “Analog Die - Power, Clock and Resets - PCR" for details on the PCR Status Register (PCR_SR (hi)), including masking information.

5.3.4.3 LIN Driver Over-temperature Interrupt (LTI)

Acknowledge the interrupt by reading the LIN Register - LINR. The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.11, “LIN" for details on the LIN Register, including masking information.

5.3.4.4 TIM Channel 0 Interrupt (CH0)

See Section 5.9, “Basic Timer Module - TIM (TIM16B4C)".

5.3.4.5 TIM Channel 1 Interrupt (CH1)

See Section 5.9, “Basic Timer Module - TIM (TIM16B4C)".

5.3.4.6 TIM Channel 2 Interrupt (CH2)

See Section 5.9, “Basic Timer Module - TIM (TIM16B4C)".

5.6 Temperature Measurement - TSENSE

5.6.1 Introduction

This chapter only gives a summary of the temperature sense module. Refer to Section 5.7, “Channel Acquisition" for the complete description of all acquisition channels, including the temperature measurement channel.

5.6.1.1 Features

- Internal on chip Temperature Sensor
- Optional External Temperature Sensor Input (VTEMP)
- Dedicated 16-Bit Sigma Delta ADC
- Programmable Gain and Offset Compensation
- Optional External Sensor Supply (TSUP) with selectable capacitor
- Optional Measurement during Low Power mode to trigger recalibration

5.6.1.2 Block Diagram

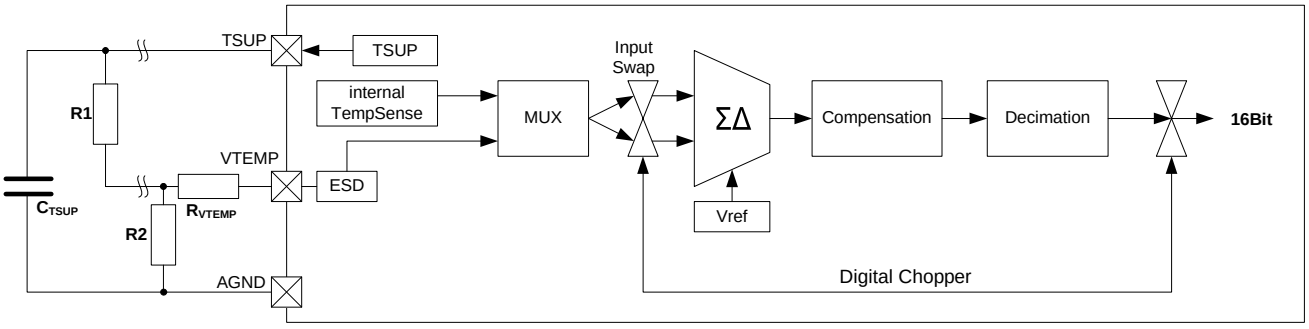


Figure 31. Temperature Measurement Channel

NOTE

To minimize ground shift effects while using the external sensor option, R2 must be placed as close to the AGND pin as possible.

C_{TSUP} is optional. The supply output must be configured to operate with the capacitor.

5.7.6.3.14 Low Pass Filter Coefficient Number (ACQ_LPFC)

Table 135. Low Pass Filter Coefficient Number (ACQ_LPFC)

Offset[133] 0x6E — Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	LPFC[3:0]			
W								
Reset	0	0	0	0	1	1	1	0

Notes
133.Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 136. Low Pass Filter Coefficient Number (ACQ_LPFC) - Register Field Descriptions

Field	Description
3-0 LPFC[3:0]	Low Pass Filter Coefficient Number. Defines the highest coefficient Number used. 0000 - LPF used with Coefficient A0 0001 - LPF used with Coefficient A0...A1 .... 1111 - LPF used with Coefficient A0...A15

5.7.6.3.15 Low Power Trigger Current Measurement Period (ACQ_TCMP)

Table 137. Low Power Trigger Current Measurement Period (ACQ_TCMP)

Offset (134)(135) 0x70 — Access: User read / write

	7	6	5	4	3	2	1	0
R	TCMP[15:8]							
W								
Reset	0	0	0	0	0	0	0	0
R	TCMP[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes
134.Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
135.This Register is 16-Bit access only.

Table 138. Low Power Trigger Current Measurement Period (ACQ_TCMP) - Register Field Descriptions

Field	Description
15-0 TCMP[15:0]	Low power trigger current measurement period (Trigger counter based on ALFCLK). See Section 5.2.4.1.1, “Cyclic Current Acquisition / Calibration Temperature Check".

NOTE

The cyclic acquisition period must be greater than the acquisition time. See Section 5.7.3.5.2, “Latency and Throughput"” for estimation. A continuous acquisition is still possible by using TCMP=0.

Table 150. Low Power Ah Counter (ACQ_AHC1 (hi) / ACQ_AHC1 (lo) / ACQ_AHC0 (hi) / ACQ_AHC0 (lo)) - Register Field Descriptions

Field	Description
31-0 AHC[31:0]	Low power Ah counter (32-Bit signed integer, two's complement). Reading one 16-Bit part of the register will buffer the second. Reading the second will unlock the buffer. See Section 5.2.4.1.1.2, "Current Ampere Hour Threshold Wake-up".

5.7.6.3.22 Low Pass Filter Coefficient Ax (LPF_Ax (hi))

Table 151. Low Pass Filter Coefficient Ax (LPF_Ax (hi))

Offset[142] 0x80...0x9E — Access: User read/write

	7	6	5	4	3	2	1	0
R W	Ax[15:8]							
Reset	see Table 154							

Notes
142.Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

5.7.6.3.23 Low Pass Filter Coefficient Ax (LPF_Ax (lo))

Table 152. Low Pass Filter Coefficient Ax (LPF_Ax (lo))

Offset[143] 0x81...0x9F — Access: User read/write

	7	6	5	4	3	2	1	0
R W	Ax[7:0]							
Reset	see Table 154							

Notes
143.Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 153. Low Pass Filter Coefficient Ax - Register Field Descriptions

Field	Description
15-0 Ax[15:0]	Low Pass Filter Coefficient Value. x = 0...15. Data Format: MSB = Sign ("1" minus). [14:0] integer.

Table 154. Low Pass Filter Coefficient Ax - Reset Values

Field	Reset Value	Field	Reset Value
A0	0x00F5	A8	0x1021
A1	0x0312	A9	0x0E35
A2	0x051F	A10	0x0B44
A3	0x0852	A11	0x0852
A4	0x0B44	A12	0x051F
A5	0x0E35	A13	0x0312
A6	0x1021	A14	0x00F5
A7	0x10E5	A15	0x0000

Table 180. External Temp. Offset Compensation (COMP_ETO) - Register Field Descriptions

Field	Description
7-0 ETOC[7:0]	External Temperature Offset Compensation Buffer. This register contains the External Temperature Offset compensation as 8-bit signed char (two complement). Refer to Section 5.7.3.4, “Compensation" for details.

5.7.6.3.37 External Temp. Gain Compensation (COMP_ETG)

Table 181. External Temp. Gain Compensation (COMP_ETG)

Offset[163] 0xD3 Access: User read/write

	7	6	5	4	3	2	1	0
R W	ETGC[7:0]							
Reset	1	0	0	0	0	0	0	0

Notes

163.Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 182. External Temp. Gain Compensation (COMP_ETG) - Register Field Descriptions

Field	Description
7-0 ETGC[7:0]	External Temperature Gain Compensation Buffer. This register contains the External Temperature Gain compensation as 8-bit special coded value. Refer to Section 5.7.3.4, “Compensation" for details.

NOTE

Care must be taken to ensure that all interrupt requests remain active until the system begins execution of the applicable service routine; otherwise, the exception request may not get processed at all or the result may be a spurious interrupt request (vector at address (vector base + 0x0080)).

5.17.4.3 Reset Exception Requests

The INT module supports three system reset exception request types (Refer to the Clock and Reset generator module for details):

1. Pin reset, power-on reset or illegal address reset, low voltage reset (if applicable)
2. Clock monitor reset request
3. COP watchdog reset request

5.17.4.4 Exception Priority

The priority (from highest to lowest) and address of all exception vectors issued by the INT module upon request by the CPU is shown in Table 312.

Table 312. Exception Vector Map and Priority

Vector Address[240]	Source
0xFFFE	Pin reset, power-on reset, illegal address reset, low voltage reset (if applicable)
0xFFFC	Clock monitor reset
0xFFFA	COP watchdog reset
(Vector base + 0x00F8)	Unimplemented opcode trap
(Vector base + 0x00F6)	Software interrupt instruction (SWI) or BDM vector request
(Vector base + 0x00F4)	X bit maskable interrupt request (XIRQ or D2D error interrupt)[241]
(Vector base + 0x00F2)	IRQ or D2D interrupt request[242]
(Vector base + 0x00F0–0x0082)	Device specific I bit maskable interrupt sources (priority determined by the low byte of the vector address, in descending order)
(Vector base + 0x0080)	Spurious interrupt

Notes
240.16-bit vector address based
241.D2D error interrupt on MCUs featuring a D2D initiator module, otherwise XIRQ pin interrupt
242.D2D interrupt on MCUs featuring a D2D initiator module, otherwise IRQ pin interrupt

5.17.5 Initialization/Application Information

5.17.5.1 Initialization

After a system reset, the software should:

1. Initialize the interrupt vector base register, if the interrupt vector table is not located at the default location (0xFF80–0xFFF9).
2. Enable I bit maskable interrupts by clearing the I bit in the CCR.
3. Enable the X bit maskable interrupt by clearing the X bit in the CCR.

Table 342. State Control Register Access Encoding

COMRV	Visible State Control Register
00	DBGSCR1
01	DBGSCR2
10	DBGSCR3
11	DBGMFR

5.19.3.2.7.1 Debug State Control Register 1 (DBGSCR1)

Table 343. Debug State Control Register 1 (DBGSCR1)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

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

This register is visible at 0x0027 only with COMRV[1:0] = 00. The state control register 1 selects the targeted next state while in State1. The matches refer to the match channels of the comparator match control logic, as depicted in Figure 63 and described in Section 5.19.3.2.8.1, “Debug Comparator Control Register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 344. DBGSCR1 Field Descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state while in State1, based upon the match event.

Table 345. State1 Sequencer Next State Selection

SC[3:0]	Description (Unspecified matches have no effect)
0000	Any match to Final State
0001	Match1 to State3
0010	Match2 to State2
0011	Match1 to State2
0100	Match0 to State2..... Match1 to State3
0101	Match1 to State3.....Match0 to Final State
0110	Match0 to State2..... Match2 to State3
0111	Either Match0 or Match1 to State2
1000	Reserved
1001	Match0 to State3
1010	Reserved
1011	Reserved
1100	Reserved
1101	Either Match0 or Match2 to Final State.....Match1 to State2

If the TAG bit is clear (forced type match), 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 when the opcode is fetched from the memory, which precedes the instruction execution by an indefinite number of cycles due to instruction pipelining. For a comparator match of an opcode at an odd address when TAG = 0, the corresponding even address must be contained in the comparator register. Thus for an opcode at odd address (n), the comparator register must contain address (n–1).

Once a successful comparator match has occurred, the condition that caused the original match is not verified again on subsequent matches. If a particular data value is verified at a given address, this address may not still contain that data value when a subsequent match occurs.

Match[0, 1, 2] map directly to Comparators [A, B, C] respectively, except in range modes (see Section 5.19.3.2.4, “Debug Control Register2 (DBGC2)"). Comparator channel priority rules are described in the priority section (Section 5.19.4.3.4, “Channel Priorities").

5.19.4.2.1 Single Address Comparator Match

With range comparisons disabled, the match condition is an exact equivalence of address bus, with the value stored in the comparator address registers. Further qualification of the type of access (R/W, word/byte) and data bus contents is possible, depending on comparator channel.

5.19.4.2.1.1 Comparator C

Comparator C offers only address and direction (R/W) comparison. The exact address is compared, with the comparator address register loaded with address (n), a word access of address (n–1) also accesses (n) but does not cause a match.

Table 374. Comparator C Access Considerations

Condition For Valid Match	Comp C Address	RWE	RW	Examples
Read and write accesses of ADDR[n]	ADDR[n](252)	0	X	LDAA ADDR[n] STAA #$BYTE ADDR[n]
Write accesses of ADDR[n]	ADDR[n]	1	0	STAA #$BYTE ADDR[n]
Read accesses of ADDR[n]	ADDR[n]	1	1	LDAA #$BYTE ADDR[n]

Notes

252.A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match. The comparator address register must contain the exact address from the code.

5.19.4.2.1.2 Comparator B

Comparator B offers address, direction (R/W) and access size (word/byte) comparison. If the SZE bit is set, the access size (word or byte) is compared with the SZ bit value such that only the specified size of access causes a match. If configured for a byte access of a particular address, a word access covering the same address does not lead to match.

Assuming the access direction is not qualified (RWE=0), for simplicity, the size access considerations are shown in Table 375.

Table 375. Comparator B Access Size Considerations

Condition For Valid Match	Comp B Address	RWE	SZE	SZ8	Examples
Word and byte accesses of ADDR[n]	ADDR[n](253)	0	0	X	MOVB #$BYTE ADDR[n] MOVW #$WORD ADDR[n]
Word accesses of ADDR[n] only	ADDR[n]	0	1	0	MOVW #$WORD ADDR[n] LDD ADDR[n]
Byte accesses of ADDR[n] only	ADDR[n]	0	1	1	MOVB #$BYTE ADDR[n] LDAB ADDR[n]

Notes

253.A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match. The comparator address register must contain the exact address from the code.

Access direction can also be used to qualify a match for Comparator B in the same way, as described for Comparator C in Table 374.

5.19.4.2.1.3 Comparator A

Comparator A offers address, direction (R/W), access size (word/byte), and data bus comparison.

Table 376 lists access considerations with data bus comparison. On word accesses, the data byte of the lower address is mapped to DBGADH. Access direction can also be used to qualify a match for Comparator A in the same way as described for Comparator C in Table 374.

Table 376. Comparator A Matches When Accessing ADDR[n]

SZE	SZ	DBGADHM, DBGADLM	Access DH=DBGADH, DL=DBGADL	Comment
0	X	$0000	Byte Word	No databus comparison
0	X	$FF00	Byte, data(ADDR[n])=DH Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match data(ADDR[n])
0	X	$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match data(ADDR[n+1])
0	X	$00FF	Byte, data(ADDR[n])=X, data(ADDR[n+1])=DL	Possible unintended match
0	X	$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data(ADDR[n], ADDR[n+1])
0	X	$FFFF	Byte, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Possible unintended match
1	0	$0000	Word	No databus comparison
1	0	$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match only data at ADDR[n+1]
1	0	$FF00	Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match only data at ADDR[n]
1	0	$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data at ADDR[n] & ADDR[n+1]
1	1	$0000	Byte	No databus comparison
1	1	$FF00	Byte, data(ADDR[n])=DH	Match data at ADDR[n]

5.19.4.2.1.4 Comparator A Data Bus Comparison NDB Dependency

Comparator A features an NDB control bit, which allows data bus comparators to be configured to either trigger on equivalence or trigger on difference. This allows monitoring of a difference in the contents of an address location from an expected value.

When matching on an equivalence (NDB=0), each individual data bus bit position can be masked out by clearing the corresponding mask bit (DBGADHM/DBGADLM) so that it is ignored in the comparison. A match occurs when all data bus bits with corresponding mask bits set are equivalent. If all mask register bits are clear, then a match is based on the address bus only, the data bus is ignored.

When matching on a difference, mask bits can be cleared to ignore bit positions. A match occurs when any data bus bit with corresponding mask bit set is different. Clearing all mask bits causes all bits to be ignored and prevents a match because no difference can be detected. In this case, address bus equivalence does not cause a match.

Table 377. NDB and MASK Bit Dependency

NDB	DBGADHM[n] / DBGADLM[n]	Comment
0	0	Do not compare data bus bit.
0	1	Compare data bus bit. Match on equivalence.
1	0	Do not compare data bus bit.
1	1	Compare data bus bit. Match on difference.

5.19.5.2 Scenario 1

A trigger is generated if a given sequence of 3 code events is executed.

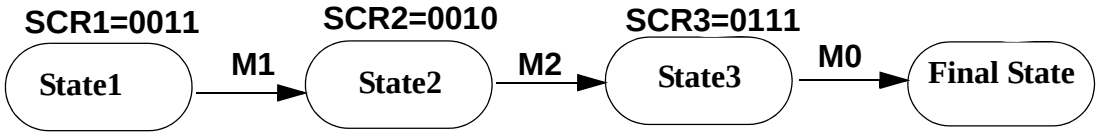


Figure 66. Scenario 1

Scenario 1 is possible with S12SDBGV1 SCR encoding.

5.19.5.3 Scenario 2

A trigger is generated if a given sequence of 2 code events is executed.

Figure 67. Scenario 2a

A trigger is generated if a given sequence of 2 code events is executed, whereby the first event is entry into a range (COMPA, COMPB configured for range mode). M1 is disabled in range modes.

Figure 68. Scenario 2b

A trigger is generated if a given sequence of 2 code events is executed, whereby the second event is entry into a range (COMPA, COMPB configured for range mode).

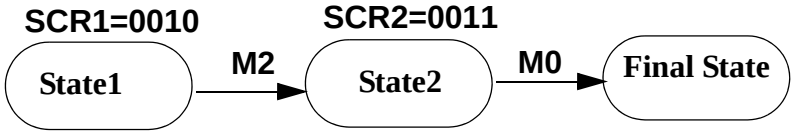


Figure 69. Scenario 2c

All 3 scenarios 2a, 2b, 2c are possible with the S12SDBGV1 SCR encoding.

5.21.3.2.2 BDM Program Page Index Register (BDMPPR)

Table 397. BDM Program Page Register (BDMPPR)

Register Global Address 0x3_FF08	7	6	5	4	3	2	1	0
R	BPAE	0	0	0	BPP3	BPP2	BPP1	BPP0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented, Reserved

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

Table 398. BDMPPR Field Descriptions

Field	Description
7 BPAE	**BDM Program Page Access Enable Bit** — BPAE enables program 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 Program Paging disabled 1 BDM Program Paging enabled
3–0 BPP[3:0]	**BDM Program Page Index Bits 3–0** — These bits define the selected program page. For more detailed information regarding the program page window scheme, refer to the S12S_MMC Block Guide.

5.21.3.3 Family ID Assignment

The family ID is an 8-bit value located in the BDM ROM in active BDM (at global address: 0x3_FF0F). The read-only value is a unique family ID which is 0xC2 for devices with an HCS12S core.

5.21.4 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 5.21.4.3, “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 5.21.4.4, “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 5.21.4.3, “BDM Hardware Commands") and in secure mode (see Section 5.21.4.1, “Security"). BDM firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

5.21.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip Flash EEPROM is erased. This being the case, the UNSEC and ENBDM bits will get set. The BDM program jumps to the start of the standard BDM firmware, the secured mode BDM firmware is turned off, and all BDM commands are allowed. If the Flash does not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the Flash.

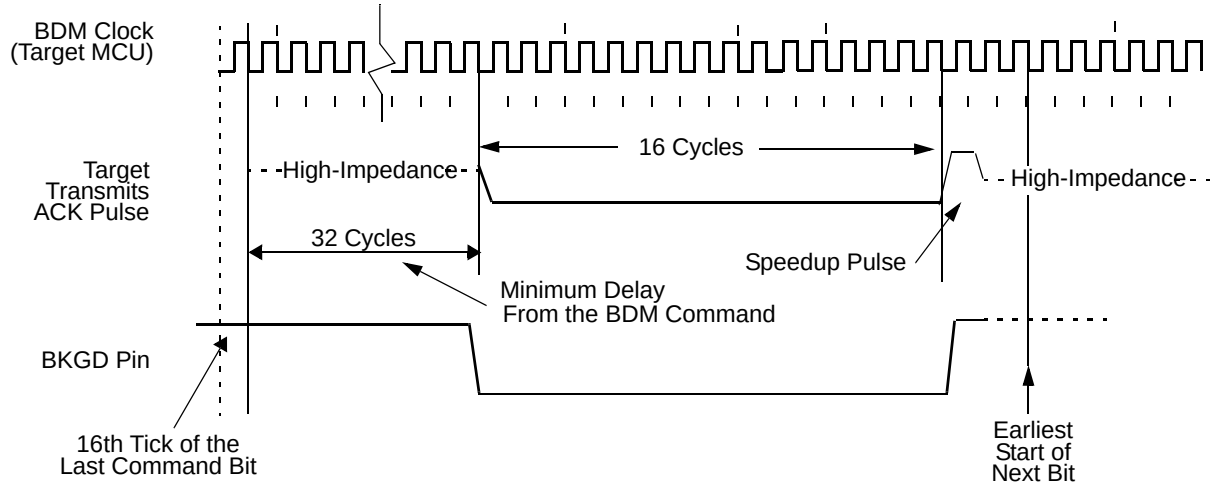


Figure 86. Target Acknowledge Pulse (ACK)

NOTE

If the ACK pulse was issued by the target, the host assumes the previous command was executed. If the CPU enters wait or stop prior to executing a hardware command, the ACK pulse will not be issued meaning that the BDM command was not executed. After entering wait or stop mode, the BDM command is no longer pending.

Figure 87 shows the ACK handshake protocol in a command level timing diagram. The READ_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word, and the host needs to determine which is the appropriate byte, based on whether the address was odd or even.

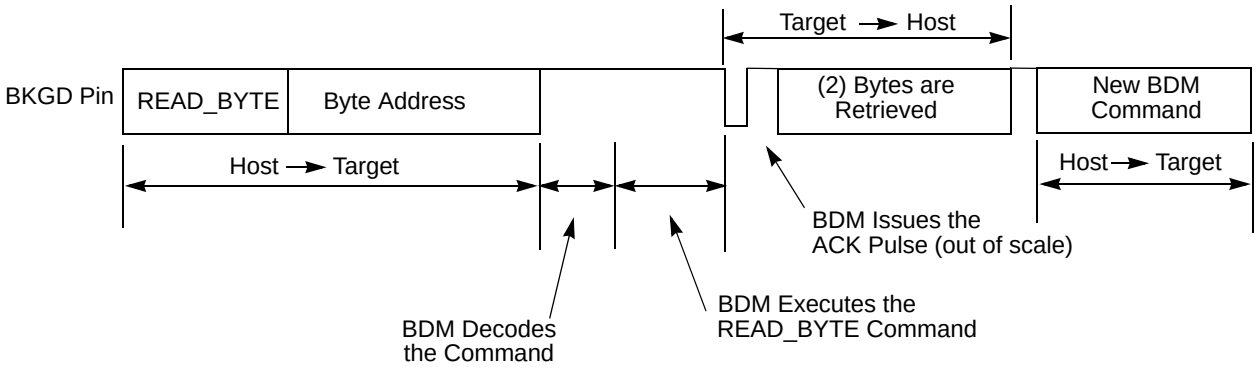


Figure 87. Handshake Protocol at Command Level

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge on the BKGD pin. The hardware handshake protocol in Figure 86 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint to avoid the risk of an electrical conflict on the BKGD pin.

NOTE

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other “highs” are pulled rather than driven. The time of the speedup pulse can become lengthy at low rates, and so the potential conflict time becomes longer as well.

Table 418. RTI Frequency Divide Rates for RTDEC = 0

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	**001 (2^{10})**	**010 (2^{11})**	**011 (2^{12})**	**100 (2^{13})**	**101 (2^{14})**	**110 (2^{15})**	**111 (2^{16})**
1110 (÷15)	OFF	$15x2^{10}$	$15x2^{11}$	$15x2^{12}$	$15x2^{13}$	$15x2^{14}$	$15x2^{15}$	$15x2^{16}$
1111 (÷16)	OFF	$16x2^{10}$	$16x2^{11}$	$16x2^{12}$	$16x2^{13}$	$16x2^{14}$	$16x2^{15}$	$16x2^{16}$

Notes
267.Denotes the default value out of reset.This value should be used to disable the RTI to ensure future backwards compatibility.

Table 419. RTI Frequency Divide Rates for RTDEC=1

RTR[3:0]	RTR[6:4] =							
	000 ($1x10^3$)	**001 ($2x10^3$)**	**010 ($5x10^3$)**	**011 ($10x10^3$)**	**100 ($20x10^3$)**	**101 ($50x10^3$)**	**110 ($100x10^3$)**	**111 ($200x10^3$)**
0000 (÷1)	$1x10^3$	$2x10^3$	$5x10^3$	$10x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$
0001 (÷2)	$2x10^3$	$4x10^3$	$10x10^3$	$20x10^3$	$40x10^3$	$100x10^3$	$200x10^3$	$400x10^3$
0010 (÷3)	$3x10^3$	$6x10^3$	$15x10^3$	$30x10^3$	$60x10^3$	$150x10^3$	$300x10^3$	$600x10^3$
0011 (÷4)	$4x10^3$	$8x10^3$	$20x10^3$	$40x10^3$	$80x10^3$	$200x10^3$	$400x10^3$	$800x10^3$
0100 (÷5)	$5x10^3$	$10x10^3$	$25x10^3$	$50x10^3$	$100x10^3$	$250x10^3$	$500x10^3$	$1x10^6$
0101 (÷6)	$6x10^3$	$12x10^3$	$30x10^3$	$60x10^3$	$120x10^3$	$300x10^3$	$600x10^3$	$1.2x10^6$
0110 (÷7)	$7x10^3$	$14x10^3$	$35x10^3$	$70x10^3$	$140x10^3$	$350x10^3$	$700x10^3$	$1.4x10^6$
0111 (÷8)	$8x10^3$	$16x10^3$	$40x10^3$	$80x10^3$	$160x10^3$	$400x10^3$	$800x10^3$	$1.6x10^6$
1000 (÷9)	$9x10^3$	$18x10^3$	$45x10^3$	$90x10^3$	$180x10^3$	$450x10^3$	$900x10^3$	$1.8x10^6$
1001 (÷10)	$10 x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$	$500x10^3$	$1x10^6$	$2x10^6$
1010 (÷11)	$11 x10^3$	$22x10^3$	$55x10^3$	$110x10^3$	$220x10^3$	$550x10^3$	$1.1x10^6$	$2.2x10^6$
1011 (÷12)	$12x10^3$	$24x10^3$	$60x10^3$	$120x10^3$	$240x10^3$	$600x10^3$	$1.2x10^6$	$2.4x10^6$
1100 (÷13)	$13x10^3$	$26x10^3$	$65x10^3$	$130x10^3$	$260x10^3$	$650x10^3$	$1.3x10^6$	$2.6x10^6$
1101 (÷14)	$14x10^3$	$28x10^3$	$70x10^3$	$140x10^3$	$280x10^3$	$700x10^3$	$1.4x10^6$	$2.8x10^6$
1110 (÷15)	$15x10^3$	$30x10^3$	$75x10^3$	$150x10^3$	$300x10^3$	$750x10^3$	$1.5x10^6$	$3x10^6$
1111 (÷16)	$16x10^3$	$32x10^3$	$80x10^3$	$160x10^3$	$320x10^3$	$800x10^3$	$1.6x10^6$	$3.2x10^6$

Table 421. CPMUCOP Field Descriptions (continued)

Field	Description
5 WRTMASK	**Write Mask for WCOP and CR[2:0] Bit** — This write-only bit serves as a mask for the WCOP and CR[2:0] bits while writing the CPMUCOP register. It is intended for BDM writing the RSBCK without changing the content of WCOP and CR[2:0]. 0 Write of WCOP and CR[2:0] has an effect with this write of CPMUCOP 1 Write of WCOP and CR[2:0] has no effect with this write of CPMUCOP. (Does not count for “write once”.)
2–0 CR[2:0]	**COP Watchdog Timer Rate Select** — These bits select the COP timeout rate (see Table 422). Writing a nonzero value to CR[2:0] enables the COP counter and starts the timeout period. A COP counter timeout causes a System Reset. This can be avoided by periodically (before timeout) initializing the COP counter via the CPMUARMCOP register. While all of the following four conditions are true the CR[2:0], WCOP bits are ignored and the COP operates at highest timeout period (2^{24} cycles) in normal COP mode (Window COP mode disabled): 1) COP is enabled (CR[2:0] is not 000) 2) BDM mode active 3) RSBCK = 0 4) Operation in Special mode

Table 422. COP Watchdog Rates

CR2	CR1	CR0	COPCLK Cycles to Time-out (COPCLK is either IRCCLK or OSCCLK depending on the COPOSCSEL bit)
0	0	0	COP disabled
0	0	1	2^{14}
0	1	0	2^{16}
0	1	1	2^{18}
1	0	0	2^{20}
1	0	1	2^{22}
1	1	0	2^{23}
1	1	1	2^{24}

5.22.3.2.10 Reserved Register CPMUTEST0

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special mode can alter the S12CPMU’s functionality.

Table 423. Reserved Register (CPMUTEST0)

0x003D

	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

Read: Anytime

Write: Only in Special mode

5.22.3.2.16 S12CPMU Oscillator Register (CPMUOSC)

This register configures the external oscillator (OSCLCP).

Table 434. S12CPMU Oscillator Register (CPMUOSC)

0x02FA

	7	6	5	4	3	2	1	0
R	OSCE	OSCBW	OSCPINS_EN	OSCFILT[4:0]				
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register), 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 ((f_{VCO} / f_{OSC})/2) is not an integer number, the filter can not be used and the OSCFILT[4:0] bits must be set to 0.

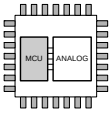
NOTE

The frequency modulation (FM1 and FM0) can not be used if the Adaptive Oscillator Filter is enabled.

Table 435. 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 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 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 in Section 5.22.4.5.2, “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).
5 OSCPINS_EN	Oscillator Pins EXTAL and XTAL Enable Bit If OSCE=1 this read-only bit is set. It can only be cleared with the next reset. Enabling the external oscillator reserves the EXTAL and XTAL pins exclusively for oscillator application. 0 EXTAL and XTAL pins are not reserved for oscillator. 1 EXTAL and XTAL pins exclusively reserved for oscillator.
4-0 OSCFILT	**Oscillator Filter Bits** — When using the oscillator a noise filter can be enabled, which filters noise from the incoming external oscillator clock and detects if the external oscillator clock is qualified or not (quality status shown by bit UPOSC). The VCOCLK-to-OSCCLK ratio divided by two ((f_{VCO} / f_{OSC})/2) must be an integer value. This value must be written to the OSCFILT[4:0] bits to enable the Adaptive Oscillator Filter. 0x0000 Adaptive Oscillator Filter disabled, else Adaptive Oscillator Filter enabled]

5.24 128 kByte Flash Module (S12FTMRC128K1V1)



5.24.1 Introduction

The FTMRC128K1 module implements the following:

- 128 kbytes of P-Flash (Program Flash) memory
- 4.0 kbytes of D-Flash (Data Flash) memory

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

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

The Flash memory may be read as bytes, aligned words, or misaligned words. Read access time is one bus cycle for bytes and aligned words, and two bus cycles for misaligned words. For Flash memory, an erased bit reads 1 and a programmed bit reads 0.

It is possible to read from P-Flash memory while some commands are executing on D-Flash memory. It is not possible to read from D-Flash memory while a command is executing on P-Flash memory. Simultaneous P-Flash and D-Flash operations are discussed in Section 5.24.4.4, “Allowed Simultaneous P-Flash and D-Flash Operations".

Both P-Flash and D-Flash memories are implemented with Error Correction Codes (ECC) that can resolve single bit faults and detect double bit faults. For P-Flash memory, the ECC implementation requires that programming be done on an aligned 8-byte basis (a Flash phrase). Since P-Flash memory is always read by half-phrase, only one single bit fault in an aligned 4-byte half-phrase containing the byte or word accessed will be corrected.

5.24.1.1 Glossary

Command Write Sequence — An MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

D-Flash Memory — The D-Flash memory constitutes the nonvolatile memory store for data.

D-Flash Sector — The D-Flash sector is the smallest portion of the D-Flash memory that can be erased. The D-Flash sector consists of four 64-byte rows for a total of 256-bytes.

NVM Command Mode — An NVM mode using the CPU to setup the FCCOB register to pass parameters required for Flash command execution.

Phrase — An aligned group of four 16-bit words within the P-Flash memory. Each phrase includes two sets of aligned double words with each set, including 7 ECC bits for single bit fault correction and double bit fault detection within each double word.

P-Flash Memory — The P-Flash memory constitutes the main nonvolatile memory store for applications.

P-Flash Sector — The P-Flash sector is the smallest portion of the P-Flash memory that can be erased. Each P-Flash sector contains 512-bytes.

Program IFR — Nonvolatile information register located in the P-Flash block that contains the Device ID, Version ID, and the Program Once field.

Table 465. D-Flash and Memory Controller Resource Fields

Global Address	Size (Bytes)	Description
0x0_4000 – 0x0_43FF	1,024	Reserved
0x0_4400 – 0x0_53FF	4,096	D-Flash Memory
0x0_5400 – 0x0_57FF	1,024	Reserved
0x0_5800 – 0x0_5AFF	768	Memory Controller Scratch RAM (RAMON(281) = 1)
0x0_5B00 – 0x0_5FFF	1,280	Reserved
0x0_6000 – 0x0_67FF	2,048	Reserved
0x0_6800 – 0x0_7FFF	6,144	Reserved

Notes
281.MMCCTL1 register bit

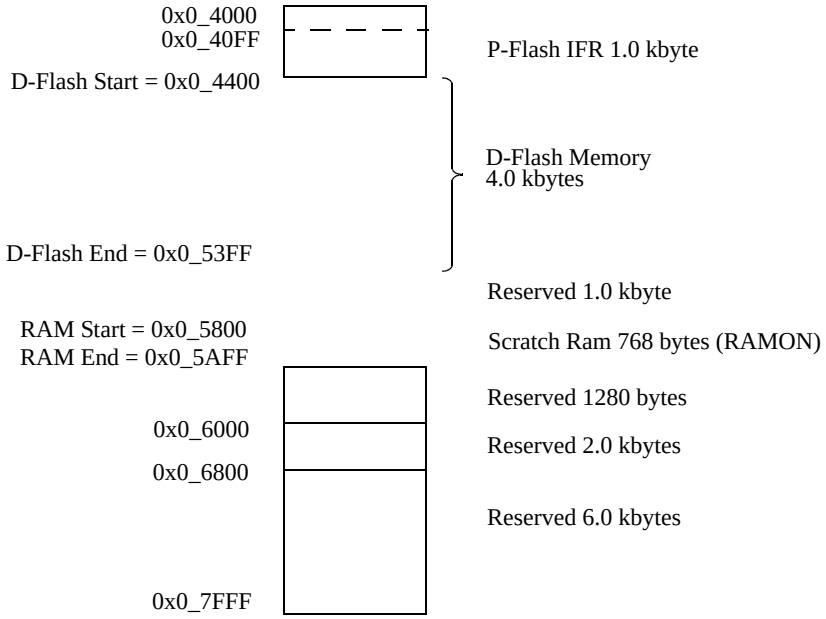


Figure 111. D-Flash and Memory Controller Resource Memory Map

5.24.3.2 Register Descriptions

The Flash module contains a set of 20 control and status registers located between 0x0100 and 0x0113. A summary of the Flash module registers is given in Table 466 with detailed descriptions in the following subsections.

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and adversely affect Memory Controller behavior.

Table 493. D-Flash Protection Address Range

DPS[3:0]	Global Address Range	Protected Size
0000	0x0_4400 – 0x0_44FF	256 bytes
0001	0x0_4400 – 0x0_45FF	512 bytes
0010	0x0_4400 – 0x0_46FF	768 bytes
0011	0x0_4400 – 0x0_47FF	1024 bytes
0100	0x0_4400 – 0x0_48FF	1280 bytes
0101	0x0_4400 – 0x0_49FF	1536 bytes
0110	0x0_4400 – 0x0_4AFF	1792 bytes
0111	0x0_4400 – 0x0_4BFF	2048 bytes
1000	0x0_4400 – 0x0_4CFF	2304 bytes
1001	0x0_4400 – 0x0_4DFF	2560 bytes
1010	0x0_4400 – 0x0_4EFF	2816 bytes
1011	0x0_4400 – 0x0_4FFF	3072 bytes
1100	0x0_4400 – 0x0_50FF	3328 bytes
1101	0x0_4400 – 0x0_51FF	3584 bytes
1110	0x0_4400 – 0x0_52FF	3840 bytes
1111	0x0_4400 – 0x0_53FF	4096 bytes

5.24.3.2.11 Flash Common Command Object Register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

Table 494. Flash Common Command Object High Register (FCCOBHI)

Address: 0x010A

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 495. Flash Common Command Object Low Register (FCCOBLO)

Address: 0x010B

	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

5.24.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 496. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.