

```

// {
// busy wait
}
// Done.
BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);

```

## 17.6 BCH Memory Map/Register Definition

### BCH Hardware Register Format Summary

**BCH memory map**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
11_4000	Hardware BCH ECC Accelerator Control Register (BCH_CTRL)	32	R/W	C000_0000h	<a href="#">17.6.1/772</a>
11_4004	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_SET)	32	R/W	C000_0000h	<a href="#">17.6.1/772</a>
11_4008	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_CLR)	32	R/W	C000_0000h	<a href="#">17.6.1/772</a>
11_400C	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_TOG)	32	R/W	C000_0000h	<a href="#">17.6.1/772</a>
11_4010	Hardware ECC Accelerator Status Register 0 (BCH_STATUS0)	32	R	0000_0010h	<a href="#">17.6.2/774</a>
11_4014	Hardware ECC Accelerator Status Register 0 (BCH_STATUS0_SET)	32	R	0000_0010h	<a href="#">17.6.2/774</a>
11_4018	Hardware ECC Accelerator Status Register 0 (BCH_STATUS0_CLR)	32	R	0000_0010h	<a href="#">17.6.2/774</a>
11_401C	Hardware ECC Accelerator Status Register 0 (BCH_STATUS0_TOG)	32	R	0000_0010h	<a href="#">17.6.2/774</a>
11_4020	Hardware ECC Accelerator Mode Register (BCH_MODE)	32	R/W	0000_0000h	<a href="#">17.6.3/776</a>
11_4024	Hardware ECC Accelerator Mode Register (BCH_MODE_SET)	32	R/W	0000_0000h	<a href="#">17.6.3/776</a>
11_4028	Hardware ECC Accelerator Mode Register (BCH_MODE_CLR)	32	R/W	0000_0000h	<a href="#">17.6.3/776</a>
11_402C	Hardware ECC Accelerator Mode Register (BCH_MODE_TOG)	32	R/W	0000_0000h	<a href="#">17.6.3/776</a>
11_4030	Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR)	32	R/W	0000_0000h	<a href="#">17.6.4/776</a>
11_4034	Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR_SET)	32	R/W	0000_0000h	<a href="#">17.6.4/776</a>
11_4038	Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR_CLR)	32	R/W	0000_0000h	<a href="#">17.6.4/776</a>
11_403C	Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR_TOG)	32	R/W	0000_0000h	<a href="#">17.6.4/776</a>

*Table continues on the next page...*

**BCH memory map (continued)**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/page
11_4040	Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR)	32	R/W	0000_0000h	<a href="#">17.6.5/777</a>
11_4044	Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR_SET)	32	R/W	0000_0000h	<a href="#">17.6.5/777</a>
11_4048	Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR_CLR)	32	R/W	0000_0000h	<a href="#">17.6.5/777</a>
11_404C	Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR_TOG)	32	R/W	0000_0000h	<a href="#">17.6.5/777</a>
11_4050	Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR)	32	R/W	0000_0000h	<a href="#">17.6.6/777</a>
11_4054	Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR_SET)	32	R/W	0000_0000h	<a href="#">17.6.6/777</a>
11_4058	Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR_CLR)	32	R/W	0000_0000h	<a href="#">17.6.6/777</a>
11_405C	Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR_TOG)	32	R/W	0000_0000h	<a href="#">17.6.6/777</a>
11_4070	Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT)	32	R/W	E4E4_E4E4h	<a href="#">17.6.7/778</a>
11_4074	Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT_SET)	32	R/W	E4E4_E4E4h	<a href="#">17.6.7/778</a>
11_4078	Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT_CLR)	32	R/W	E4E4_E4E4h	<a href="#">17.6.7/778</a>
11_407C	Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT_TOG)	32	R/W	E4E4_E4E4h	<a href="#">17.6.7/778</a>
11_4080	Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0)	32	R/W	070A_4080h	<a href="#">17.6.8/779</a>
11_4084	Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0_SET)	32	R/W	070A_4080h	<a href="#">17.6.8/779</a>
11_4088	Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0_CLR)	32	R/W	070A_4080h	<a href="#">17.6.8/779</a>
11_408C	Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0_TOG)	32	R/W	070A_4080h	<a href="#">17.6.8/779</a>
11_4090	Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1)	32	R/W	10DA_4080h	<a href="#">17.6.9/781</a>
11_4094	Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1_SET)	32	R/W	10DA_4080h	<a href="#">17.6.9/781</a>
11_4098	Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1_CLR)	32	R/W	10DA_4080h	<a href="#">17.6.9/781</a>
11_409C	Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1_TOG)	32	R/W	10DA_4080h	<a href="#">17.6.9/781</a>
11_40A0	Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0)	32	R/W	070A_4080h	<a href="#">17.6.10/782</a>
11_40A4	Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0_SET)	32	R/W	070A_4080h	<a href="#">17.6.10/782</a>

Table continues on the next page...

**BCH memory map (continued)**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/page
11_40A8	Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0_CLR)	32	R/W	070A_4080h	<a href="#">17.6.10/782</a>
11_40AC	Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0_TOG)	32	R/W	070A_4080h	<a href="#">17.6.10/782</a>
11_40B0	Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1)	32	R/W	10DA_4080h	<a href="#">17.6.11/784</a>
11_40B4	Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1_SET)	32	R/W	10DA_4080h	<a href="#">17.6.11/784</a>
11_40B8	Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1_CLR)	32	R/W	10DA_4080h	<a href="#">17.6.11/784</a>
11_40BC	Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1_TOG)	32	R/W	10DA_4080h	<a href="#">17.6.11/784</a>
11_40C0	Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0)	32	R/W	070A_4080h	<a href="#">17.6.12/785</a>
11_40C4	Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0_SET)	32	R/W	070A_4080h	<a href="#">17.6.12/785</a>
11_40C8	Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0_CLR)	32	R/W	070A_4080h	<a href="#">17.6.12/785</a>
11_40CC	Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0_TOG)	32	R/W	070A_4080h	<a href="#">17.6.12/785</a>
11_40D0	Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1)	32	R/W	10DA_4080h	<a href="#">17.6.13/787</a>
11_40D4	Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1_SET)	32	R/W	10DA_4080h	<a href="#">17.6.13/787</a>
11_40D8	Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1_CLR)	32	R/W	10DA_4080h	<a href="#">17.6.13/787</a>
11_40DC	Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1_TOG)	32	R/W	10DA_4080h	<a href="#">17.6.13/787</a>
11_40E0	Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0)	32	R/W	070A_4080h	<a href="#">17.6.14/788</a>
11_40E4	Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0_SET)	32	R/W	070A_4080h	<a href="#">17.6.14/788</a>
11_40E8	Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0_CLR)	32	R/W	070A_4080h	<a href="#">17.6.14/788</a>
11_40EC	Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0_TOG)	32	R/W	070A_4080h	<a href="#">17.6.14/788</a>
11_40F0	Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1)	32	R/W	10DA_4080h	<a href="#">17.6.15/790</a>
11_40F4	Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1_SET)	32	R/W	10DA_4080h	<a href="#">17.6.15/790</a>
11_40F8	Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1_CLR)	32	R/W	10DA_4080h	<a href="#">17.6.15/790</a>
11_40FC	Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1_TOG)	32	R/W	10DA_4080h	<a href="#">17.6.15/790</a>

Table continues on the next page...

**BCH memory map (continued)**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/page
11_4100	Hardware BCH ECC Debug Register0 (BCH_DEBUG0)	32	R/W	0000_0000h	<a href="#">17.6.16/791</a>
11_4104	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_SET)	32	R/W	0000_0000h	<a href="#">17.6.16/791</a>
11_4108	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_CLR)	32	R/W	0000_0000h	<a href="#">17.6.16/791</a>
11_410C	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_TOG)	32	R/W	0000_0000h	<a href="#">17.6.16/791</a>
11_4110	KES Debug Read Register (BCH_DBGKESREAD)	32	R	0000_0000h	<a href="#">17.6.17/793</a>
11_4114	KES Debug Read Register (BCH_DBGKESREAD_SET)	32	R	0000_0000h	<a href="#">17.6.17/793</a>
11_4118	KES Debug Read Register (BCH_DBGKESREAD_CLR)	32	R	0000_0000h	<a href="#">17.6.17/793</a>
11_411C	KES Debug Read Register (BCH_DBGKESREAD_TOG)	32	R	0000_0000h	<a href="#">17.6.17/793</a>
11_4120	Chien Search Debug Read Register (BCH_DBGCSFEREAD)	32	R	0000_0000h	<a href="#">17.6.18/793</a>
11_4124	Chien Search Debug Read Register (BCH_DBGCSFEREAD_SET)	32	R	0000_0000h	<a href="#">17.6.18/793</a>
11_4128	Chien Search Debug Read Register (BCH_DBGCSFEREAD_CLR)	32	R	0000_0000h	<a href="#">17.6.18/793</a>
11_412C	Chien Search Debug Read Register (BCH_DBGCSFEREAD_TOG)	32	R	0000_0000h	<a href="#">17.6.18/793</a>
11_4130	Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD)	32	R	0000_0000h	<a href="#">17.6.19/794</a>
11_4134	Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD_SET)	32	R	0000_0000h	<a href="#">17.6.19/794</a>
11_4138	Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD_CLR)	32	R	0000_0000h	<a href="#">17.6.19/794</a>
11_413C	Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD_TOG)	32	R	0000_0000h	<a href="#">17.6.19/794</a>
11_4140	Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD)	32	R	0000_0000h	<a href="#">17.6.20/794</a>
11_4144	Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD_SET)	32	R	0000_0000h	<a href="#">17.6.20/794</a>
11_4148	Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD_CLR)	32	R	0000_0000h	<a href="#">17.6.20/794</a>
11_414C	Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD_TOG)	32	R	0000_0000h	<a href="#">17.6.20/794</a>
11_4150	Block Name Register (BCH_BLOCKNAME)	32	R	2048_4342h	<a href="#">17.6.21/795</a>
11_4154	Block Name Register (BCH_BLOCKNAME_SET)	32	R	2048_4342h	<a href="#">17.6.21/795</a>

*Table continues on the next page...*

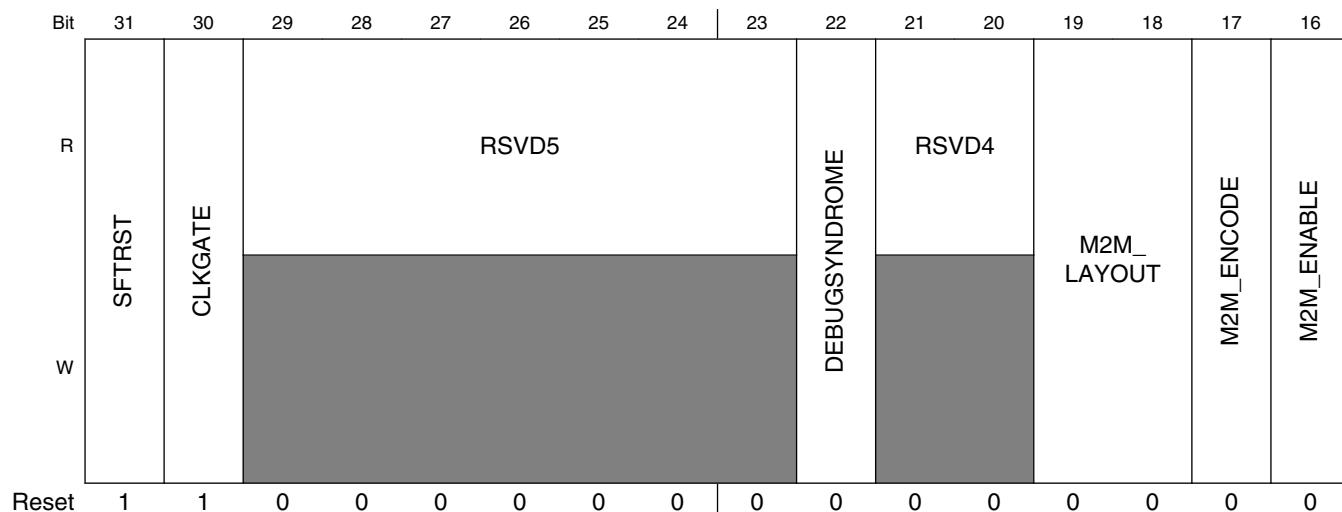
**BCH memory map (continued)**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/page
11_4158	Block Name Register (BCH_BLOCKNAME_CLR)	32	R	2048_4342h	<a href="#">17.6.21/795</a>
11_415C	Block Name Register (BCH_BLOCKNAME_TOG)	32	R	2048_4342h	<a href="#">17.6.21/795</a>
11_4160	BCH Version Register (BCH_VERSION)	32	R	0100_0000h	<a href="#">17.6.22/795</a>
11_4164	BCH Version Register (BCH_VERSION_SET)	32	R	0100_0000h	<a href="#">17.6.22/795</a>
11_4168	BCH Version Register (BCH_VERSION_CLR)	32	R	0100_0000h	<a href="#">17.6.22/795</a>
11_416C	BCH Version Register (BCH_VERSION_TOG)	32	R	0100_0000h	<a href="#">17.6.22/795</a>

### 17.6.1 Hardware BCH ECC Accelerator Control Register (BCH\_CTRL*n*)

The BCH CTRL provides overall control of the hardware ECC accelerator

Address: 11\_4000h base + 0h offset + (4d × i), where i=0d to 3d



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
						RSVD3										
							DEBUG_STALL_IRQ_EN		RSVD2				RSVD1			
														BM_ERROR_IRQ		
														DEBUG_STALL_IRQ		
														RSVD0		
															COMPLETE_IRQ	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**BCH\_CTRLn field descriptions**

Field	Description
31 SFTRST	Set this bit to zero to enable normal BCH operation. Set this bit to one (default) to disable clocking with the BCH and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the BCH block to its default state. This bit resets all state machines except for the AHB master state machine  0x0 <b>RUN</b> — Allow BCH to operate normally. 0x1 <b>RESET</b> — Hold BCH in reset.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.  0x0 <b>RUN</b> — Allow BCH to operate normally. 0x1 <b>NO_CLKS</b> — Do not clock BCH gates in order to minimize power consumption.
29–23 RSVD5	Reserved, always set this bit to zero.
22 DEBUGSYNDROME	(For debug purposes only). Enable write of computed syndromes to memory on BCH decode operations. Computed syndromes will be written to the auxiliary buffer after the status block. Syndromes will be written as padded 16-bit values.
21–20 RSVD4	Reserved, always set these bits to zero.
19–18 M2M_LAYOUT	Selects the flash page format for memory-to-memory operations.
17 M2M_ENCODE	Selects encode (parity generation) or decode (correction) mode for memory-to-memory operations.
16 M2M_ENABLE	NOTE! WRITING THIS BIT INITIATES A MEMORY-TO-MEMORY OPERATION. The BCH module must be inactive (not processing data from the GPMI) when this bit is set. The M2M_ENCODE and M2M_LAYOUT bits as well as the ENCODEPTR, DATAPTR, and METAPTR registers are used for memory-to-memory operations and must be correctly programmed before writing this bit.

*Table continues on the next page...*

**BCH\_CTRLn field descriptions (continued)**

Field	Description
15–11 RSVD3	Reserved, always set these bits to zero.
10 DEBUG_STALL_ IRQ_EN	1 = interrupt on debug stall mode is enabled. The irq is raised on every block
9 RSVD2	Reserved, always set these bits to zero.
8 COMPLETE_IRQ_ EN	1 = interrupt on completion of correction is enabled.
7–4 RSVD1	Reserved, always set these bits to zero.
3 BM_ERROR_IRQ	AHB Bus interface Error Interrupt Status. Write a one to the SCT clear address to clear the interrupt status bit.
2 DEBUG_STALL_IRQ	DEBUG STALL Interrupt Status. Write a one to the SCT clear address to clear the interrupt status bit.
1 RSVD0	Reserved, always set these bits to zero.
0 COMPLETE_IRQ	This bit indicates the state of the external interrupt line. Write a one to the SCT clear address to clear the interrupt status bit. NOTE: subsequent ECC completions will be held off as long as this bit is set. Be sure to read the data from BCH_STATUS0,1 before clearing this interrupt bit.

## 17.6.2 Hardware ECC Accelerator Status Register 0 (BCH\_STATUS0n)

The BCH STAT register provides visibility into the run-time status of the BCH and status information when processing is complete. It provides overall status of the hardware ECC accelerator.

Address: 11\_4000h base + 10h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R									HANDLE					COMPLETED_CE		
W																

Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									RSVD1		ALLONES	CORRECTED	UNCORRECTABLE		RSVD0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

**BCH\_STATUS0n field descriptions**

Field	Description
31–20 HANDLE	Software supplies a 12 bit handle for this transfer as part of the GPMI DMA PIO operation that started the transaction. That handle passes down the pipeline and ends up here at the time the BCH interrupt is signaled.
19–16 COMPLETED_CE	This is the chip enable number corresponding to the NAND device from which this data came.
15–8 STATUS_BLK0	Count of symbols in error during processing of first block of flash (metadata block). The number of errors reported will be in the range of 0 to the ECC correction level for block 0.  0x00 <b>ZERO</b> — No errors found on block. 0x01 <b>ERROR1</b> — One error found on block. 0x02 <b>ERROR2</b> — Two errors found on block. 0x03 <b>ERROR3</b> — Three errors found on block. 0x04 <b>ERROR4</b> — Four errors found on block. 0xFE <b>UNCORRECTABLE</b> — Block exhibited uncorrectable errors. 0xFF <b>ERASED</b> — Page is erased.
7–5 RSVD1	Reserved, always set these bits to zero.
4 ALLONES	1 = All data bits of this transaction are ONE.
3 CORRECTED	1 = At least one correctable error encountered during last processing cycle.
2 UNCORRECTABLE	1 = Uncorrectable error encountered during last processing cycle.
RSVD0	Reserved, always set these bits to zero.

### 17.6.3 Hardware ECC Accelerator Mode Register (BCH\_MODEn)

The BCH MODE register provides additional mode controls.

Contains additional global mode controls for the BCH engine.

Address: 11\_4000h base + 20h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

#### BCH\_MODEn field descriptions

Field	Description
31–8 RSVD	Reserved, always set these bits to zero.
ERASE_THRESHOLD	This value indicates the maximum number of zero bits on a flash page for it to be considered erased. For SLC NAND devices, this value should be programmed to 0 (meaning that the entire page should consist of bytes of 0xFF). For MLC NAND devices, bit errors may occur on reads (even on blank pages), so this threshold can be used to tune the erased page checking algorithm.

### 17.6.4 Hardware BCH ECC Loopback Encode Buffer Register (BCH\_ENCODEPTRn)

When performing memory to memory operations, indicates the address of the encode buffer. This register should be programmed before writing a 1 to the M2M\_ENABLE bit in the CTRL register.

For memory to memory operations, this register is used as the pointer to the encoded data, which is an output when encoding and an input while decoding.

Address: 11\_4000h base + 30h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

#### BCH\_ENCODEPTRn field descriptions

Field	Description
ADDR	Address pointer to encode buffer. This is the source for decode operations and the destination for encode operations. This value must be aligned on a 4 byte boundary.

## 17.6.5 Hardware BCH ECC Loopback Data Buffer Register (BCH DATAPTR $n$ )

When performing memory to memory operations, indicates the address of the data buffer.

For memory to memory operations, this register is used as the pointer to the data to encode or the destination buffer for decode operations.

Address: 11 4000h base + 40h offset + (4d x i), where i=0d to 3d

## BCH DATAPTR $n$ field descriptions

Field	Description
ADDR	Address pointer to data buffer. This is the source for encode operations and the destination for decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 byte boundary.

## 17.6.6 Hardware BCH ECC Loopback Metadata Buffer Register (BCH\_METAPTR $n$ )

When performing memory to memory operations, indicates the address of the metadata buffer.

For memory to memory operations, this register is used as the pointer to the metadata to encode or the extracted metadata for decode operations.

Address: 11 4000h base + 50h offset + (4d × i), where i=0d to 3d

## BCH METAPTR $n$ field descriptions

Field	Description
ADDR	Address pointer to metadata buffer. This is the source for encode metadata read operations and the destination for metadata decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 byte boundary.

## 17.6.7 Hardware ECC Accelerator Layout Select Register (BCH\_LAYOUTSELECT $n$ )

The BCH LAYOUTSELECT register provides a mapping of chip selects to layout registers.

When the BCH engine receives a request to process a data block from the GPMI interface, it will use this register to map the incoming chip select to one of the four possible flash layout registers

Address: 11\_4000h base + 70h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CS15_ SELECT	CS14_ SELECT	CS13_ SELECT	CS12_ SELECT			CS11_ SELECT		CS10_ SELECT		CS9_ SELECT		CS8_ SELECT			
W																
Reset	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CS7_ SELECT	CS6_ SELECT	CS5_ SELECT	CS4_ SELECT			CS3_ SELECT		CS2_ SELECT		CS1_ SELECT		CS0_ SELECT			
W																
Reset	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0

### BCH\_LAYOUTSELECT $n$ field descriptions

Field	Description
31–30 CS15_SELECT	Selects which layout is used for chip select 15.
29–28 CS14_SELECT	Selects which layout is used for chip select 14.
27–26 CS13_SELECT	Selects which layout is used for chip select 13.
25–24 CS12_SELECT	Selects which layout is used for chip select 12.
23–22 CS11_SELECT	Selects which layout is used for chip select 11.
21–20 CS10_SELECT	Selects which layout is used for chip select 10.
19–18 CS9_SELECT	Selects which layout is used for chip select 9.
17–16 CS8_SELECT	Selects which layout is used for chip select 8.
15–14 CS7_SELECT	Selects which layout is used for chip select 7.
13–12 CS6_SELECT	Selects which layout is used for chip select 6.
11–10 CS5_SELECT	Selects which layout is used for chip select 5.

Table continues on the next page...

## BCH LAYOUTSELECT $n$ field descriptions (continued)

Field	Description
9–8 CS4_SELECT	Selects which layout is used for chip select 4.
7–6 CS3_SELECT	Selects which layout is used for chip select 3.
5–4 CS2_SELECT	Selects which layout is used for chip select 2.
3–2 CS1_SELECT	Selects which layout is used for chip select 1.
CS0_SELECT	Selects which layout is used for chip select 0.

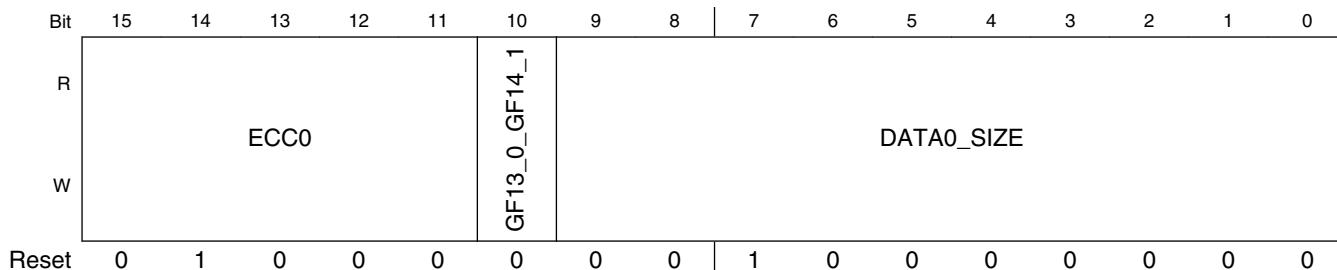
## 17.6.8 Hardware BCH ECC Flash 0 Layout 0 Register (BCH\_FLASH0LAYOUT0n)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT1 register to control the format for the devices selecting layout 0 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading/writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks. See the BCH programming reference manual for more information on setting up the flash layout registers.

Address: 11\_4000h base + 80h offset + (4d x i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NBLOCKS								META_SIZE							
W	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0

**BCH\_FLASH0LAYOUT0n field descriptions**

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata -- if set to zero, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a zero, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15–11 ECC0	<p>Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field.</p> <p>0x0 <b>NONE</b> — No ECC to be performed      0x1 <b>ECC2</b> — ECC 2 to be performed      0x2 <b>ECC4</b> — ECC 4 to be performed      0x3 <b>ECC6</b> — ECC 6 to be performed      0x4 <b>ECC8</b> — ECC 8 to be performed      0x5 <b>ECC10</b> — ECC 10 to be performed      0x6 <b>ECC12</b> — ECC 12 to be performed      0x7 <b>ECC14</b> — ECC 14 to be performed      0x8 <b>ECC16</b> — ECC 16 to be performed      0x9 <b>ECC18</b> — ECC 18 to be performed      0xA <b>ECC20</b> — ECC 20 to be performed      0xB <b>ECC22</b> — ECC 22 to be performed      0xC <b>ECC24</b> — ECC 24 to be performed      0xD <b>ECC26</b> — ECC 26 to be performed      0xE <b>ECC28</b> — ECC 28 to be performed      0xF <b>ECC30</b> — ECC 30 to be performed      0x10 <b>ECC32</b> — ECC 32 to be performed      0x11 <b>ECC34</b> — ECC 34 to be performed      0x12 <b>ECC36</b> — ECC 36 to be performed      0x13 <b>ECC38</b> — ECC 38 to be performed      0x14 <b>ECC40</b> — ECC 40 to be performed</p>
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to zero, the first block will only contain metadata.

## 17.6.9 Hardware BCH ECC Flash 0 Layout 1 Register (BCH\_FLASH0LAYOUT1n)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT0 register to control the format for the device selecting layout 0 in the LAYOUTSELECT register.

Address: 11\_4000h base + 90h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	1	0	0	0	0	1	1	0	1	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W																
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

### BCH\_FLASH0LAYOUT1n field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). <ul style="list-style-type: none"> <li>0x0 <b>NONE</b> — No ECC to be performed</li> <li>0x1 <b>ECC2</b> — ECC 2 to be performed</li> <li>0x2 <b>ECC4</b> — ECC 4 to be performed</li> <li>0x3 <b>ECC6</b> — ECC 6 to be performed</li> <li>0x4 <b>ECC8</b> — ECC 8 to be performed</li> <li>0x5 <b>ECC10</b> — ECC 10 to be performed</li> <li>0x6 <b>ECC12</b> — ECC 12 to be performed</li> <li>0x7 <b>ECC14</b> — ECC 14 to be performed</li> <li>0x8 <b>ECC16</b> — ECC 16 to be performed</li> <li>0x9 <b>ECC18</b> — ECC 18 to be performed</li> <li>0xA <b>ECC20</b> — ECC 20 to be performed</li> <li>0xB <b>ECC22</b> — ECC 22 to be performed</li> <li>0xC <b>ECC24</b> — ECC 24 to be performed</li> </ul>

Table continues on the next page...

## **BCH\_FLASH0LAYOUT1*n* field descriptions (continued)**

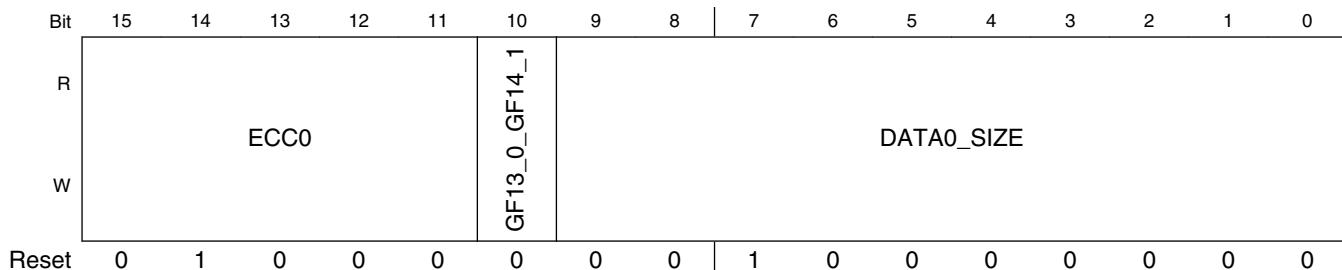
Field	Description
	0xD <b>ECC26</b> — ECC 26 to be performed 0xE <b>ECC28</b> — ECC 28 to be performed 0xF <b>ECC30</b> — ECC 30 to be performed 0x10 <b>ECC32</b> — ECC 32 to be performed 0x11 <b>ECC34</b> — ECC 34 to be performed 0x12 <b>ECC36</b> — ECC 36 to be performed 0x13 <b>ECC38</b> — ECC 38 to be performed 0x14 <b>ECC40</b> — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page.
	The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

## 17.6.10 Hardware BCH ECC Flash 1 Layout 0 Register (BCH\_FLASH1LAYOUT0n)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT1 register to control the format for the devices selecting layout 1 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading/writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks. See the BCH programming reference manual for more information on setting up the flash layout registers.

Address: 11 4000h base + A0h offset + (4d × i), where i=0d to 3d

**BCH\_FLASH1\_LAYOUT0n field descriptions**

Field	Description																																										
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.																																										
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata -- if set to zero, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a zero, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.																																										
15–11 ECC0	<p>Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field.</p> <table> <tr><td>0x0</td><td><b>NONE</b> — No ECC to be performed</td></tr> <tr><td>0x1</td><td><b>ECC2</b> — ECC 2 to be performed</td></tr> <tr><td>0x2</td><td><b>ECC4</b> — ECC 4 to be performed</td></tr> <tr><td>0x3</td><td><b>ECC6</b> — ECC 6 to be performed</td></tr> <tr><td>0x4</td><td><b>ECC8</b> — ECC 8 to be performed</td></tr> <tr><td>0x5</td><td><b>ECC10</b> — ECC 10 to be performed</td></tr> <tr><td>0x6</td><td><b>ECC12</b> — ECC 12 to be performed</td></tr> <tr><td>0x7</td><td><b>ECC14</b> — ECC 14 to be performed</td></tr> <tr><td>0x8</td><td><b>ECC16</b> — ECC 16 to be performed</td></tr> <tr><td>0x9</td><td><b>ECC18</b> — ECC 18 to be performed</td></tr> <tr><td>0xA</td><td><b>ECC20</b> — ECC 20 to be performed</td></tr> <tr><td>0xB</td><td><b>ECC22</b> — ECC 22 to be performed</td></tr> <tr><td>0xC</td><td><b>ECC24</b> — ECC 24 to be performed</td></tr> <tr><td>0xD</td><td><b>ECC26</b> — ECC 26 to be performed</td></tr> <tr><td>0xE</td><td><b>ECC28</b> — ECC 28 to be performed</td></tr> <tr><td>0xF</td><td><b>ECC30</b> — ECC 30 to be performed</td></tr> <tr><td>0x10</td><td><b>ECC32</b> — ECC 32 to be performed</td></tr> <tr><td>0x11</td><td><b>ECC34</b> — ECC 34 to be performed</td></tr> <tr><td>0x12</td><td><b>ECC36</b> — ECC 36 to be performed</td></tr> <tr><td>0x13</td><td><b>ECC38</b> — ECC 38 to be performed</td></tr> <tr><td>0x14</td><td><b>ECC40</b> — ECC 40 to be performed</td></tr> </table>	0x0	<b>NONE</b> — No ECC to be performed	0x1	<b>ECC2</b> — ECC 2 to be performed	0x2	<b>ECC4</b> — ECC 4 to be performed	0x3	<b>ECC6</b> — ECC 6 to be performed	0x4	<b>ECC8</b> — ECC 8 to be performed	0x5	<b>ECC10</b> — ECC 10 to be performed	0x6	<b>ECC12</b> — ECC 12 to be performed	0x7	<b>ECC14</b> — ECC 14 to be performed	0x8	<b>ECC16</b> — ECC 16 to be performed	0x9	<b>ECC18</b> — ECC 18 to be performed	0xA	<b>ECC20</b> — ECC 20 to be performed	0xB	<b>ECC22</b> — ECC 22 to be performed	0xC	<b>ECC24</b> — ECC 24 to be performed	0xD	<b>ECC26</b> — ECC 26 to be performed	0xE	<b>ECC28</b> — ECC 28 to be performed	0xF	<b>ECC30</b> — ECC 30 to be performed	0x10	<b>ECC32</b> — ECC 32 to be performed	0x11	<b>ECC34</b> — ECC 34 to be performed	0x12	<b>ECC36</b> — ECC 36 to be performed	0x13	<b>ECC38</b> — ECC 38 to be performed	0x14	<b>ECC40</b> — ECC 40 to be performed
0x0	<b>NONE</b> — No ECC to be performed																																										
0x1	<b>ECC2</b> — ECC 2 to be performed																																										
0x2	<b>ECC4</b> — ECC 4 to be performed																																										
0x3	<b>ECC6</b> — ECC 6 to be performed																																										
0x4	<b>ECC8</b> — ECC 8 to be performed																																										
0x5	<b>ECC10</b> — ECC 10 to be performed																																										
0x6	<b>ECC12</b> — ECC 12 to be performed																																										
0x7	<b>ECC14</b> — ECC 14 to be performed																																										
0x8	<b>ECC16</b> — ECC 16 to be performed																																										
0x9	<b>ECC18</b> — ECC 18 to be performed																																										
0xA	<b>ECC20</b> — ECC 20 to be performed																																										
0xB	<b>ECC22</b> — ECC 22 to be performed																																										
0xC	<b>ECC24</b> — ECC 24 to be performed																																										
0xD	<b>ECC26</b> — ECC 26 to be performed																																										
0xE	<b>ECC28</b> — ECC 28 to be performed																																										
0xF	<b>ECC30</b> — ECC 30 to be performed																																										
0x10	<b>ECC32</b> — ECC 32 to be performed																																										
0x11	<b>ECC34</b> — ECC 34 to be performed																																										
0x12	<b>ECC36</b> — ECC 36 to be performed																																										
0x13	<b>ECC38</b> — ECC 38 to be performed																																										
0x14	<b>ECC40</b> — ECC 40 to be performed																																										
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14																																										
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to zero, the first block will only contain metadata.																																										

### 17.6.11 Hardware BCH ECC Flash 1 Layout 1 Register (BCH\_FLASH1LAYOUT1n)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT0 register to control the format for the device selecting layout 1 in the LAYOUTSELECT register.

Address: 11\_4000h base + B0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R									PAGE_SIZE								
W																	
Reset	0	0	0	1	0	0	0	0		1	1	0	1	1	0	1	0
Bit	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R									GF13_0_GF14_1						DATAN_SIZE		
W																	
Reset	0	1	0	0	0	0	0	0		1	0	0	0	0	0	0	0

#### BCH\_FLASH1LAYOUT1n field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). <ul style="list-style-type: none"> <li>0x0 <b>NONE</b> — No ECC to be performed</li> <li>0x1 <b>ECC2</b> — ECC 2 to be performed</li> <li>0x2 <b>ECC4</b> — ECC 4 to be performed</li> <li>0x3 <b>ECC6</b> — ECC 6 to be performed</li> <li>0x4 <b>ECC8</b> — ECC 8 to be performed</li> <li>0x5 <b>ECC10</b> — ECC 10 to be performed</li> <li>0x6 <b>ECC12</b> — ECC 12 to be performed</li> <li>0x7 <b>ECC14</b> — ECC 14 to be performed</li> <li>0x8 <b>ECC16</b> — ECC 16 to be performed</li> <li>0x9 <b>ECC18</b> — ECC 18 to be performed</li> <li>0xA <b>ECC20</b> — ECC 20 to be performed</li> <li>0xB <b>ECC22</b> — ECC 22 to be performed</li> <li>0xC <b>ECC24</b> — ECC 24 to be performed</li> </ul>

Table continues on the next page...

**BCH\_FLASH1LAYOUT1*n* field descriptions (continued)**

Field	Description
	0xD <b>ECC26</b> — ECC 26 to be performed 0xE <b>ECC28</b> — ECC 28 to be performed 0xF <b>ECC30</b> — ECC 30 to be performed 0x10 <b>ECC32</b> — ECC 32 to be performed 0x11 <b>ECC34</b> — ECC 34 to be performed 0x12 <b>ECC36</b> — ECC 36 to be performed 0x13 <b>ECC38</b> — ECC 38 to be performed 0x14 <b>ECC40</b> — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

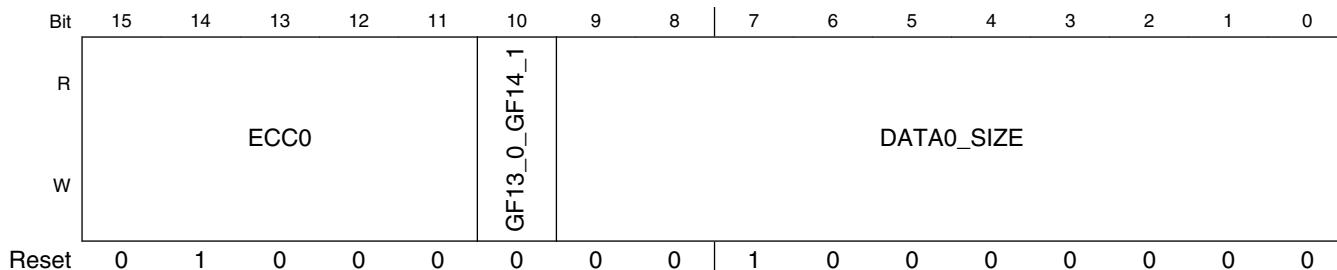
### 17.6.12 Hardware BCH ECC Flash 2 Layout 0 Register (BCH\_FLASH2LAYOUT0*n*)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT1 register to control the format for the devices selecting layout 2 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading/writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks. See the BCH programming reference manual for more information on setting up the flash layout registers.

Address: 11\_4000h base + C0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W									NBLOCKS				META_SIZE			
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0

**BCH\_FLASH2LAYOUT0n field descriptions**

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata -- if set to zero, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a zero, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15–11 ECC0	<p>Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field.</p> <p>0x0 <b>NONE</b> — No ECC to be performed      0x1 <b>ECC2</b> — ECC 2 to be performed      0x2 <b>ECC4</b> — ECC 4 to be performed      0x3 <b>ECC6</b> — ECC 6 to be performed      0x4 <b>ECC8</b> — ECC 8 to be performed      0x5 <b>ECC10</b> — ECC 10 to be performed      0x6 <b>ECC12</b> — ECC 12 to be performed      0x7 <b>ECC14</b> — ECC 14 to be performed      0x8 <b>ECC16</b> — ECC 16 to be performed      0x9 <b>ECC18</b> — ECC 18 to be performed      0xA <b>ECC20</b> — ECC 20 to be performed      0xB <b>ECC22</b> — ECC 22 to be performed      0xC <b>ECC24</b> — ECC 24 to be performed      0xD <b>ECC26</b> — ECC 26 to be performed      0xE <b>ECC28</b> — ECC 28 to be performed      0xF <b>ECC30</b> — ECC 30 to be performed      0x10 <b>ECC32</b> — ECC 32 to be performed      0x11 <b>ECC34</b> — ECC 34 to be performed      0x12 <b>ECC36</b> — ECC 36 to be performed      0x13 <b>ECC38</b> — ECC 38 to be performed      0x14 <b>ECC40</b> — ECC 40 to be performed</p>
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to zero, the first block will only contain metadata.

### 17.6.13 Hardware BCH ECC Flash 2 Layout 1 Register (BCH\_FLASH2LAYOUT1n)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT0 register to control the format for the device selecting layout 2 in the LAYOUTSELECT register.

Address: 11\_4000h base + D0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R									PAGE_SIZE								
W																	
Reset	0	0	0	1	0	0	0	0		1	1	0	1	1	0	1	0
Bit	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R									GF13_0_GF14_1						DATAN_SIZE		
W																	
Reset	0	1	0	0	0	0	0	0		1	0	0	0	0	0	0	0

#### BCH\_FLASH2LAYOUT1n field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). <ul style="list-style-type: none"> <li>0x0 <b>NONE</b> — No ECC to be performed</li> <li>0x1 <b>ECC2</b> — ECC 2 to be performed</li> <li>0x2 <b>ECC4</b> — ECC 4 to be performed</li> <li>0x3 <b>ECC6</b> — ECC 6 to be performed</li> <li>0x4 <b>ECC8</b> — ECC 8 to be performed</li> <li>0x5 <b>ECC10</b> — ECC 10 to be performed</li> <li>0x6 <b>ECC12</b> — ECC 12 to be performed</li> <li>0x7 <b>ECC14</b> — ECC 14 to be performed</li> <li>0x8 <b>ECC16</b> — ECC 16 to be performed</li> <li>0x9 <b>ECC18</b> — ECC 18 to be performed</li> <li>0xA <b>ECC20</b> — ECC 20 to be performed</li> <li>0xB <b>ECC22</b> — ECC 22 to be performed</li> <li>0xC <b>ECC24</b> — ECC 24 to be performed</li> </ul>

Table continues on the next page...

**BCH\_FLASH2LAYOUT1*n* field descriptions (continued)**

Field	Description
	0xD <b>ECC26</b> — ECC 26 to be performed 0xE <b>ECC28</b> — ECC 28 to be performed 0xF <b>ECC30</b> — ECC 30 to be performed 0x10 <b>ECC32</b> — ECC 32 to be performed 0x11 <b>ECC34</b> — ECC 34 to be performed 0x12 <b>ECC36</b> — ECC 36 to be performed 0x13 <b>ECC38</b> — ECC 38 to be performed 0x14 <b>ECC40</b> — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

### 17.6.14 Hardware BCH ECC Flash 3 Layout 0 Register (BCH\_FLASH3LAYOUT0*n*)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT1 register to control the format for the devices selecting layout 3 in the LAYOUTSELECT register.

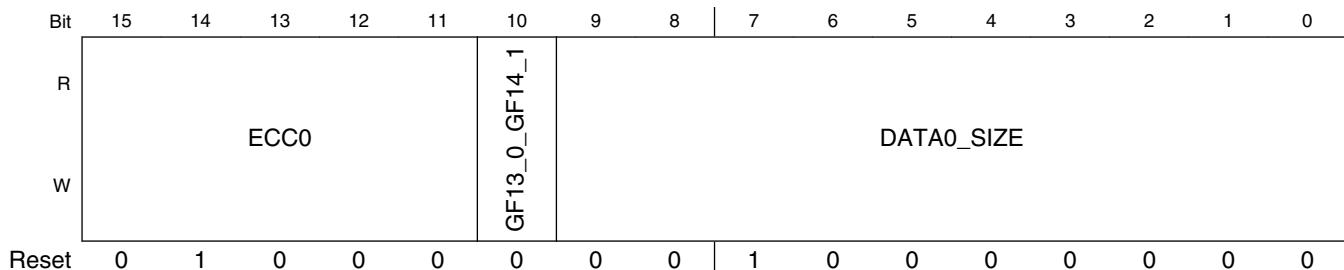
Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading/writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks. See the BCH programming reference manual for more information on setting up the flash layout registers.

Address: 11\_4000h base + E0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0

NBLOCKS

META\_SIZE

**BCH\_FLASH3\_LAYOUT0n field descriptions**

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata -- if set to zero, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a zero, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15–11 ECC0	<p>Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field.</p> <p>0x0 <b>NONE</b> — No ECC to be performed      0x1 <b>ECC2</b> — ECC 2 to be performed      0x2 <b>ECC4</b> — ECC 4 to be performed      0x3 <b>ECC6</b> — ECC 6 to be performed      0x4 <b>ECC8</b> — ECC 8 to be performed      0x5 <b>ECC10</b> — ECC 10 to be performed      0x6 <b>ECC12</b> — ECC 12 to be performed      0x7 <b>ECC14</b> — ECC 14 to be performed      0x8 <b>ECC16</b> — ECC 16 to be performed      0x9 <b>ECC18</b> — ECC 18 to be performed      0xA <b>ECC20</b> — ECC 20 to be performed      0xB <b>ECC22</b> — ECC 22 to be performed      0xC <b>ECC24</b> — ECC 24 to be performed      0xD <b>ECC26</b> — ECC 26 to be performed      0xE <b>ECC28</b> — ECC 28 to be performed      0xF <b>ECC30</b> — ECC 30 to be performed      0x10 <b>ECC32</b> — ECC 32 to be performed      0x11 <b>ECC34</b> — ECC 34 to be performed      0x12 <b>ECC36</b> — ECC 36 to be performed      0x13 <b>ECC38</b> — ECC 38 to be performed      0x14 <b>ECC40</b> — ECC 40 to be performed</p>
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to zero, the first block will only contain metadata.

### 17.6.15 Hardware BCH ECC Flash 3 Layout 1 Register (BCH\_FLASH3LAYOUT1n)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT0 register to control the format for the device selecting layout 3 in the LAYOUTSELECT register.

Address: 11\_4000h base + F0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R									PAGE_SIZE								
W																	
Reset	0	0	0	1	0	0	0	0		1	1	0	1	1	0	1	0
Bit	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R									GF13_0_GF14_1						DATAN_SIZE		
W																	
Reset	0	1	0	0	0	0	0	0		1	0	0	0	0	0	0	0

#### BCH\_FLASH3LAYOUT1n field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accommodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). <ul style="list-style-type: none"> <li>0x0 <b>NONE</b> — No ECC to be performed</li> <li>0x1 <b>ECC2</b> — ECC 2 to be performed</li> <li>0x2 <b>ECC4</b> — ECC 4 to be performed</li> <li>0x3 <b>ECC6</b> — ECC 6 to be performed</li> <li>0x4 <b>ECC8</b> — ECC 8 to be performed</li> <li>0x5 <b>ECC10</b> — ECC 10 to be performed</li> <li>0x6 <b>ECC12</b> — ECC 12 to be performed</li> <li>0x7 <b>ECC14</b> — ECC 14 to be performed</li> <li>0x8 <b>ECC16</b> — ECC 16 to be performed</li> <li>0x9 <b>ECC18</b> — ECC 18 to be performed</li> <li>0xA <b>ECC20</b> — ECC 20 to be performed</li> <li>0xB <b>ECC22</b> — ECC 22 to be performed</li> <li>0xC <b>ECC24</b> — ECC 24 to be performed</li> </ul>

Table continues on the next page...

**BCH\_FLASH3LAYOUT1*n* field descriptions (continued)**

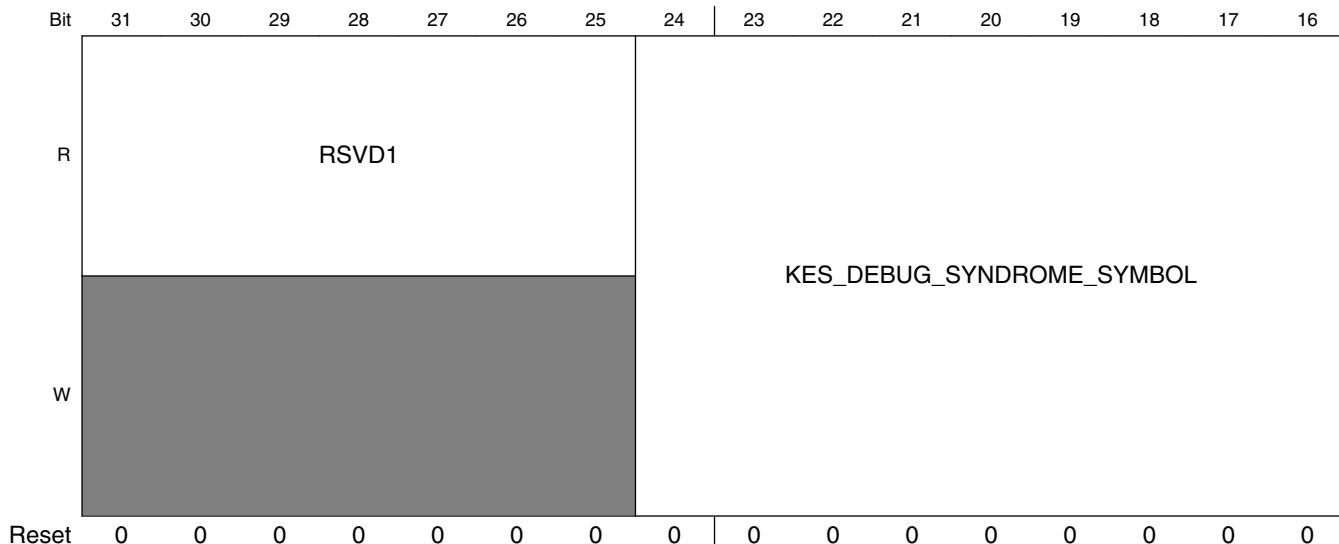
Field	Description
	0xD <b>ECC26</b> — ECC 26 to be performed 0xE <b>ECC28</b> — ECC 28 to be performed 0xF <b>ECC30</b> — ECC 30 to be performed 0x10 <b>ECC32</b> — ECC 32 to be performed 0x11 <b>ECC34</b> — ECC 34 to be performed 0x12 <b>ECC36</b> — ECC 36 to be performed 0x13 <b>ECC38</b> — ECC 38 to be performed 0x14 <b>ECC40</b> — ECC 40 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

**17.6.16 Hardware BCH ECC Debug Register0 (BCH\_DEBUG0*n*)**

The hardware BCH accelerator internal state machines and signals can be seen in the ECC debug register.

The BCH\_DEBUG0 register provides access to various internal state information which might prove useful during hardware debug and validation.

Address: 11\_4000h base + 100h offset + (4d × i), where i=0d to 3d



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	KES_DEBUG_SHIFT_SYND	KES_DEBUG_PAYLOAD_FLAG	KES_DEBUG_MODE4K	KES_DEBUG_KICK	KES_STANDALONE	KES_DEBUG_STEP	KES_DEBUG_STALL	BM_KES_TEST_BYPASS	RSVD0							
W																DEBUG_REG_SELECT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**BCH\_DEBUG0n field descriptions**

Field	Description
31–25 RSVD1	Reserved, always set these bits to zero.
24–16 KES_DEBUG_ SYNDROME_ SYMBOL	The 9 bit value in this bit field will be shifted into the syndrome register array at the input of the KES engine whenever BCH_DEBUG0_KES_DEBUG_SHIFT_SYND is toggled.  0x0 <b>NORMAL</b> — Bus master address generator for synd_gen writes operates normally. 0x1 <b>TEST_MODE</b> — Bus master address generator always addresses last four bytes in Auxilliary block.
15 KES_DEBUG_ SHIFT_SYND	Toggling this bit causes the value in BCH_DEBUG0_KES_SYNDROME_SYMBOL to be shift into the syndrome register array at the input to the KES engine. After shifting in 16 symbols, one can kick off both KES and CF cycles by toggling BCH_DEBUG0_KES_DEBUG_KICK. Be sure to set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.
14 KES_DEBUG_ PAYLOAD_FLAG	When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input payload flag.  0x1 <b>DATA</b> — Payload is set for 512 byte data block. 0x1 <b>AUX</b> — Payload is set for 65 or 19 byte auxilliary block.
13 KES_DEBUG_ MODE4K	When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input mode (4K or 2K pages).  0x1 <b>4k</b> — Mode is set for 4K NAND pages. 0x1 <b>2k</b> — Mode is set for 2K NAND pages.
12 KES_DEBUG_ KICK	Toggling causes KES engine FSM to start as if kick by the Bus Master. This allows stand alone testing of the KES and Chien Search engines. Be sure to set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.
11 KES_ STANDALONE	Set to one to cause the KES engine to suppress toggling the KES_BM_DONE signal to the bus master and to suppress toggling the CF_BM_DONE signal by the CF engine.  0x0 <b>NORMAL</b> — Bus master address generator for synd_gen writes operates normally. 0x1 <b>TEST_MODE</b> — Bus master address generator always addresses last four bytes in Auxilliary block.

Table continues on the next page...

**BCH\_DEBUG0n field descriptions (continued)**

Field	Description
10 KES_DEBUG_STEP	Toggling this bit causes the KES FSM to skip passed the stall state if it is in DEBUG_STALL mode and it has completed processing a block.
9 KES_DEBUG_STALL	Set to one to cause KES FSM to stall after notifying Chien search engine to start processing its block but before notifying the bus master that the KES computation is complete. This allows a diagnostic to stall the FSM after each blocks key equations are solved. This also has the effect of stalling the CSFE search engine so its state can be examined after it finishes processing the KES stalled block.  0x0 <b>NORMAL</b> — KES FSM proceeds to next block supplied by bus master. 0x1 <b>WAIT</b> — KES FSM waits after current equations are solved and the search engine is started.
8 BM_KES_TEST_BYPASS	1 = Point all synd_gen writes to dummy area at the end of the AUXILLIARY block so that diagnostics can preload all payload, parity bytes and computed syndrome bytes for test the KES engine.  0x0 <b>NORMAL</b> — Bus master address generator for synd_gen writes operates normally. 0x1 <b>TEST_MODE</b> — Bus master address generator always addresses last four bytes in Auxilliary block.
7–6 RSVD0	Reserved, always set these bits to zero.
DEBUG_REG_SELECT	The value loaded in this bit field is used to select the internal register state view of KES engine or the Chien search engine.

**17.6.17 KES Debug Read Register (BCH\_DBGKESREADn)**

The hardware BCH ECC accelerator key equation solver internal state machines and signals can be seen in the ECC debug registers.

Address: 11\_4000h base + 110h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

**BCH\_DBGKESREADn field descriptions**

Field	Description
VALUES	This register will return the ROM BIST CRC value after a BIST test.

**17.6.18 Chien Search Debug Read Register (BCH\_DBGCSFEREADn)**

The hardware BCH ECC accelerator Chien Search internal state machines and signals can be seen in the ECC debug registers.

Address: 11\_4000h base + 120h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

#### BCH\_DBGCSFEREADn field descriptions

Field	Description
VALUES	Reserved

### 17.6.19 Syndrome Generator Debug Read Register (BCH\_DBGSYNDGENREADn)

The hardware BCH ECC accelerator syndrome generator internal state machines and signals can be seen in the ECC debug registers.

Address: 11\_4000h base + 130h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

#### BCH\_DBGSYNDGENREADn field descriptions

Field	Description
VALUES	Reserved

### 17.6.20 Bus Master and ECC Controller Debug Read Register (BCH\_DBGAHBMREADn)

The hardware BCH ECC accelerator bus master and ecc controller internal state machines and signals can be seen in the ECC debug registers.

Address: 11\_4000h base + 140h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

#### BCH\_DBGAHBMREADn field descriptions

Field	Description
VALUES	Reserved

### 17.6.21 Block Name Register (BCH\_BLOCKNAMEn)

Read only view of the block name string BCH.

Fixed pattern read only value for test purposes. Can be read as an ASCII string with the zero termination coming from the first byte of the BLOCKVERSION register.

Address: 11\_4000h base + 150h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAME																															
W																																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	

#### BCH\_BLOCKNAMEn field descriptions

Field	Description
NAME	Should be the ASCII characters BCH (0x20, H, C, B).

### 17.6.22 BCH Version Register (BCH\_VERSIONn)

This register always returns a known read value for debug purposes and indicates the version of the block and RTL version in use.

Address: 11\_4000h base + 160h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR										MINOR										STEP											
W																																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

**BCH\_VERSIONn field descriptions**

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.