MCU F05 Platform Flash-ECC Module

The flash electrically erasable programmable read-only memory (Flash EEPROM, or Flash) module is a type of nonvolatile memory which has fast read access times but slower write and erase times. Flash EEPROM performs erasing by sector, rather than word at a time as in regular EEPROM.

Topic **Page**

1 **Revision History** 2
2 **Overview**. 3
3 **Functional Block Diagram** 5
4 **Operation** 7
5 **Control Registers**. 33

1 Revision History

Date	Version	Author	Description
4/15/04	Rev 0.10	Charles Tsai	Initial Draft
4/20/04	Rev 0.11	Charles Tsai	Memory mapped ECC bank to 4Mbyte offset as a mirror image. During OTP sector read access the ECC is bypassed. Allowed any two different sectors to be excluded from ECC checking. Added diagnostic mode to verify ECC logic. ECC logic can be removed during netlist synthesis. Non-0101 value to enable ECC logic. Any flash memory access which does not decode to any bank will generate an illegal address signal.
4/22/04	Rev 0.12	Charles Tsai	Change default bank access wait state value to 1 after reset. Added Section 7, *Manufacturing Test*
4/26/04	Rev 0.13	Charles Tsai	Global low power mode with SYS_LPM=1 will put the entire flash module in sleep mode. Clarification of the manufacturing test section Clarify that both ECC logic and diagnostic mode can be enabled at the same time.
5/11/04	Rev 0.14	Charles Tsai	Change the ECC bank from 64-bit to 16-bit datawidth.
8/12/04	Rev 0.15	Charles Tsai	Remove ECCBANKID[3:0] register Reset value of EDACEN is 0101 for disable ECC. All other combinations are ECC enabled. Support byte and half word write to register file. Improve readability

2 Overview

The MCU F05 Platform Flash-ECC is generally used to provide permanent program/data storage or factory calibration data; and can be programmed and electrically erased many times to allow for faster code development. The built-in EDAC (**E**rror **D**etection **A**nd **C**orrection) circuit allows flash memory faults to be either detected or corrected.

Flash EEPROM differs from standard EERPOM in that all bits in a flash sector are erased in bulk, whereas standard EEPROM is erased a word at a time.

2.1 Features

- ❏ Supports multiple flash banks for program and data storage.
- ❏ Contains up to 8 banks (including ECC bank) of up to 16 Megabits each
- ❏ Error detection and correction capability
- ❏ Two sectors can be selected for exclusion from ECC checking
- ❏ Enables simultaneous read access on a bank while performing a write or erase operation on any one of the remaining banks.
- ❏ Supports erase and program suspend, which allows software to fetch data from the flash bank currently being erased or programmed.
- ❏ Supports automating flash erase and programming through integrated state machine
- ❏ Allows up to 32 sectors per flash bank
- ❏ Provides optional set of four 32-bit protection keys.
- ❏ Provides built-in power mode control logic.
- ❏ Support flash memory replacement by an external RAM for emulation

2.2 Definition of Terms

Terms used in this document have the following meanings:

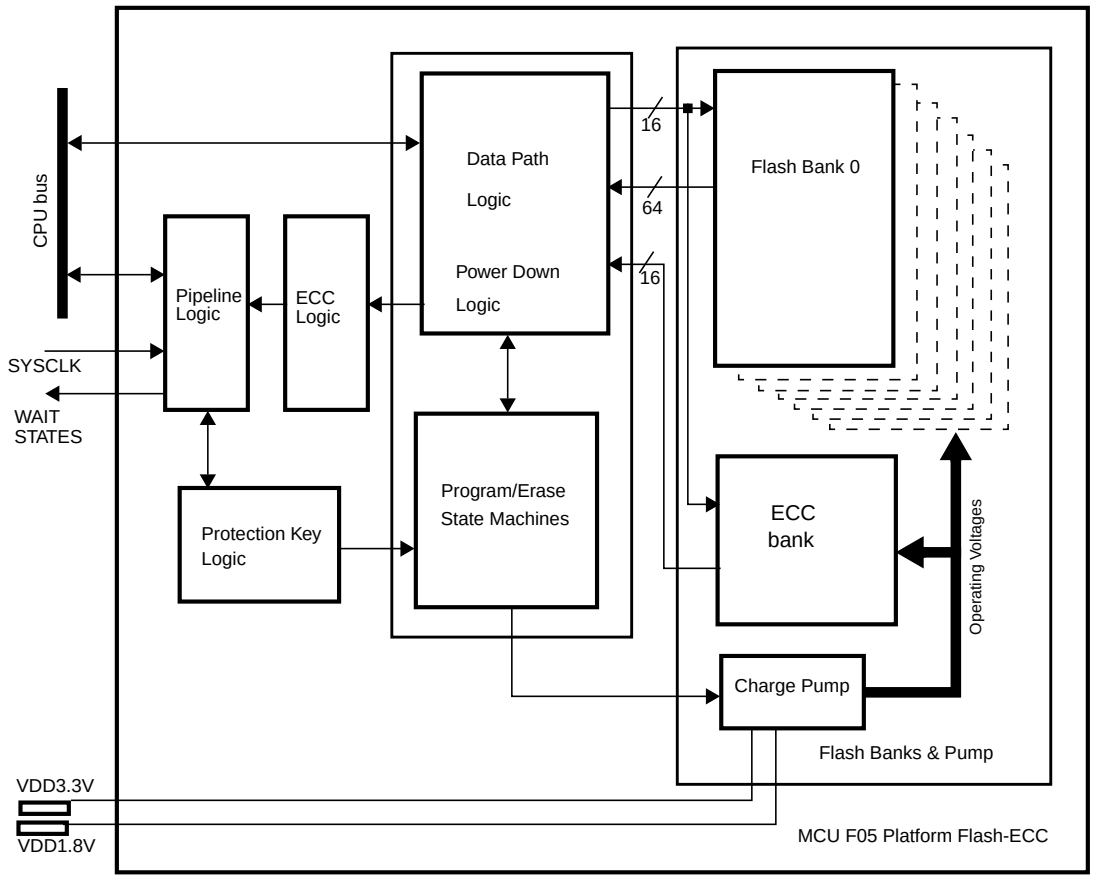
- ❏ BAGP (bank active grace period): Time (in SYSCLK cycles) from the most recent flash access of a particular bank until that bank enters fallback power mode. This reduces power consumption by the flash: however, it can also increase access time.

- ❏ Charge pump: Voltage generators and associated control (logic, oscillator, and bandgap, for example).
- ❏ CSM (command state machine): One of the three state machines present in the flash module.
- ❏ Fallback power mode: The power mode (active, standby or sleep, depending on which mode is selected) into which a bank or the charge pump falls back each time the active grace period expires.
- ❏ Flash bank: A group of flash sectors which share input/output buffers, data paths, sense amplifiers, and control logic. Flash bank can store both program instructions and data.
- ❏ ECC: Error Correction Code.
- ❏ ECC bank: Similar to flash bank but only store the ECC words.
- ❏ Flash module: Flash banks, charge pump, power and mode control logic, data path, wait logic, and write/erase state machines.
- ❏ MCU F05 Platform Flash-ECC: Flash module and CPU interface which includes pipeline logic and protection logic.
- ❏ OTP (one-time programmable): A program-only-once flash sector (cannot be erased)
- ❏ PAGP (pump active grace period): Time (in SYSCLK cycles) from when the last of the banks have entered fallback power mode until the pump enters a fallback power mode. This can reduce power consumption by the flash; however, it can also increase access time.
- ❏ Pipeline mode: The mode in which flash is read 64 bits at a time, giving faster apparent access times.
- ❏ Sector: A contiguous region of flash memory which must be erased simultaneously due to physical construction constraints.
- ❏ Standard read mode: The mode assumed when the pipeline mode is not enabled and flash is read no more than 32 bits at a time.

3 Functional Block Diagram

The MCU F05 Platform Flash-ECC consists of several major functional elements as shown in Figure 1.

Figure 1. F05 Platform Flash-ECC Functional Block Diagram



Up to eight banks (including ECC bank) of flash can be present in a single flash module. The present design supports up to 32 sectors per bank.

State machines in the flash module perform program, erase, and verify operations. The state machines permit the use of simpler program/erase/ verify software algorithms.

A single charge pump can supply all required voltages for up to eight flash banks. The MCU F05 Platform Flash-ECC requires 1.8V and 3.3V power

supplies for normal operation. Program and erase operations do not require any additional power supply.

The pipeline logic speeds up the apparent read access time of the flash module.

Programming the power-down logic correctly can produce significant power savings. The power-down registers control the power mode for each bank individually as well as for the charge pump.

The entire module is protected from unauthorized erasures or writes when the optional four 32-bit protection key words are used.

4 Operation

The following sections discuss various issues related to reset, reading, erasing, programming, power mode, protection, and wait state generation.

4.1 Reset State

The reset state exhibits the following properties:

1) Wait states are set to 1
2) Pipeline mode default is device specific
3) The OTP sector is turned off
4) All levels of protection are enabled
5) Power modes are set to *Active* (no power savings)

Your boot code must initialize the wait states and the desired pipeline mode to achieve the best possible system performance.

4.2 Flash Read Modes

In addition to *standard read mode*, the flash module also has *pipeline mode*, which affects the technique used to fetch the next memory word. Using this mode correctly increases clock speeds and CPU throughput.

4.2.1 Standard Read Mode

Standard read mode is defined as the mode in effect when pipeline mode is inactive. The number of wait states used in standard read mode is user programmable within the flash control register FMBAC2.7:0.

4.2.2 Pipeline Mode

In *pipeline mode*, two words are read in parallel from the flash core. Storing these two words in pipeline data buffers increases the bandwidth of the data coming out of the flash core, which provides effectively zero wait states on as many accesses as possible.

In pipeline mode, the flash data is always latched into the pipeline buffer first, then read from the pipeline buffer to the CPU. Pipeline mode removes the flash memory access time from the critical timing path, which allows the clock frequency to be higher.

Pipeline mode is enabled by setting the ENPIPE bit within the flash control register FMREGOPT.0. The pipeline mode default is device specific.

4.3 Memory Error Detection

Platform F05 Flash module can contain an embedded SECDED (**S**ingle **E**rror **C**orrection and **D**ouble **E**rror **D**etection) circuit. SECDED when enabled provides capability to screen out memory faults and correct the fault. SECDED requires a total of eight ECC (**E**rror **C**orrection **C**ode) check bits for each 64 bit of data to be stored in the flash memory. Data read out of flash memory pass through SECDED. During read operation, SECDED generates eight ECC bits based on the 64 bit read data value. These eight calculated ECC bits are then XORed with the pre-determined check bits associated with the read data. The 8-bit output is the syndrome. The syndrome is decoded to determine one of three conditions:

- ❏ No error occurred
- ❏ A correctable error occurred
- ❏ A non-correctable error occurred

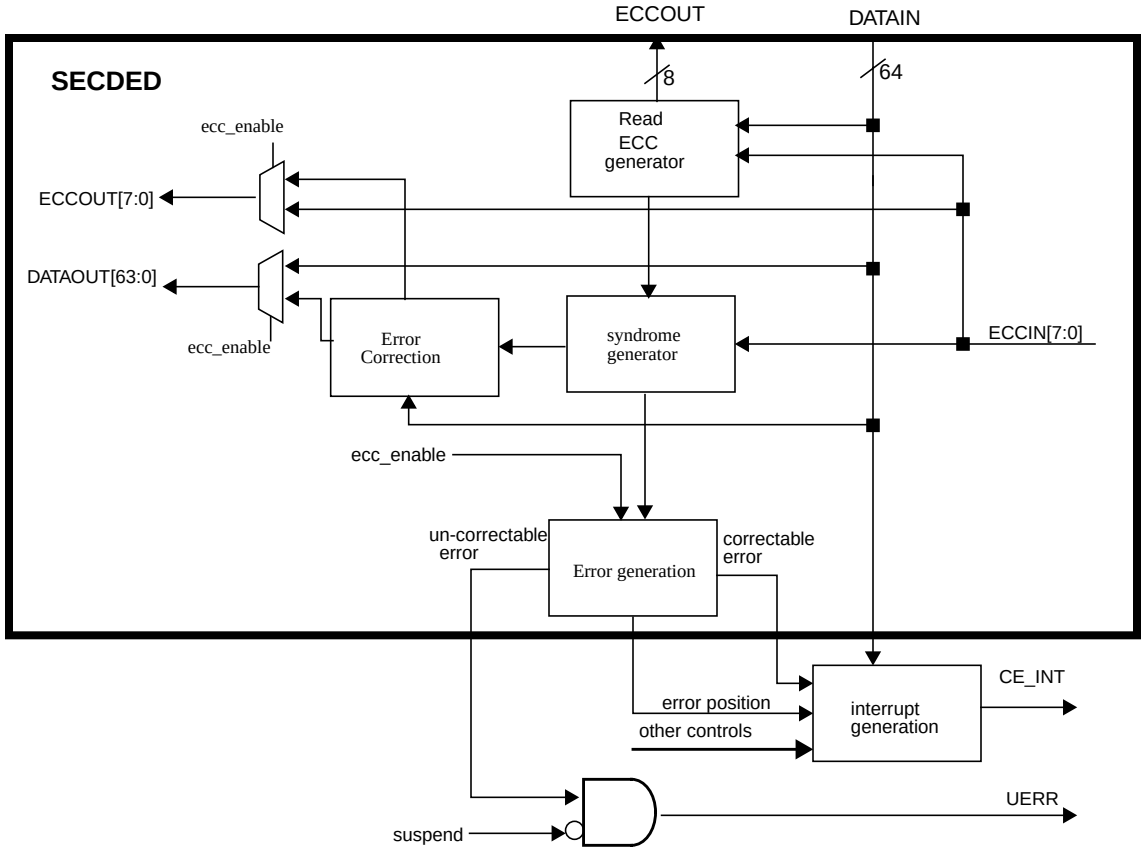
ECC is encoded based on modified Hamming code.

Figure 1–20. illustrates a SECDED block diagram.

Note: Since ECC is calculated for an entire 64-bit data, a non-64 bit read such as a byte read or a half-word read will still force the entire 64-bit data to be read and calculated.

Note: Depending on the device requirement the SECDED circuit can be included or removed during netlist synthesis

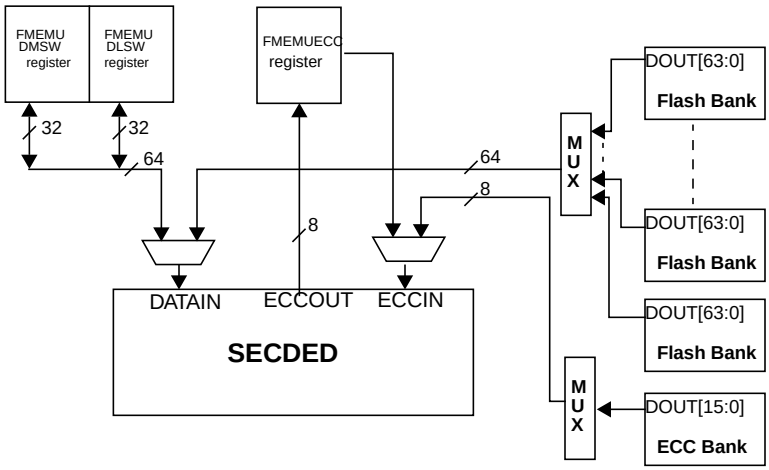
Figure 2. SECDED Block Diagram



4.3.1 ECC Flash Bank Overview

ECC words are stored in a separate flash bank. With a given flash configuration the last bank can be designated as an ECC bank. For example, if there are four flash banks then bank 4 can be designated as an ECC bank. ECC bank has 16-bit datawidth while all other flash banks have 64-bit datawidth. Therefore, reading ECC bank will return two ECC words. The correct ECC word is multiplexed out by flash wrapper during ECC syndrome generation.

Figure 3. SECDED DATAIN and ECCOUT signal mapping



Several attributes are associated with ECC bank depending if ECC feature is enabled.

When ECC feature is enabled (EDACEN=! 0101),

- ❏ flash wrapper reads both the addressed flash bank and the ECC bank
- ❏ CPU can directly read from ECC bank as if reading any other non-ECC bank but SECDED is bypassed.
- ❏ ECC logic is bypassed if reading from OTP sector.

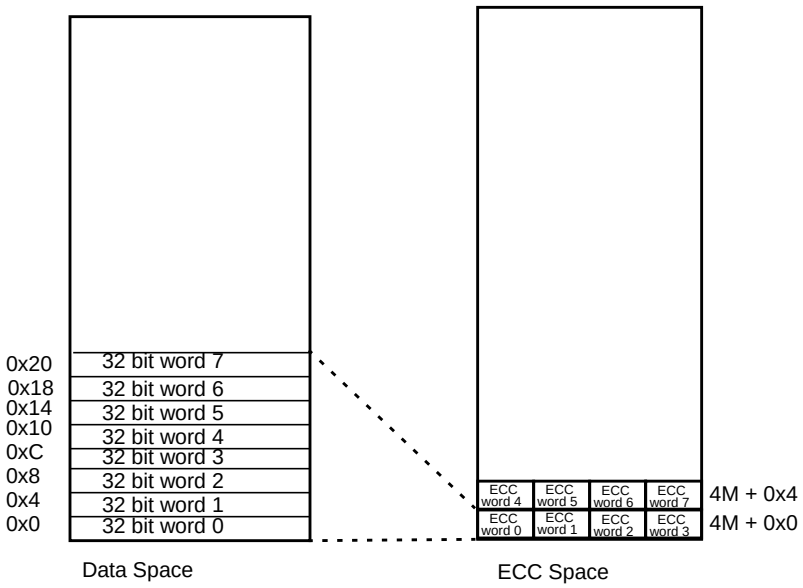
When ECC feature is disabled (EDACEN=0101),

- ❏ flash wrapper only reads the addressed flash bank without reading the designated ECC bank.
- ❏ the designated ECC bank can be used as regular data bank to store 16-bit data during non-pipeline mode.

4.3.2 ECC Memory Map

ECC data are memory mapped in the CPU's address space. It is mapped to 4M bytes offset from the flash memory base address. Each 64-bit data has its corresponding ECC word mapped to an address in the ECC space. Refer to Section Figure 4. for illustration.

Figure 4. ECC Word Memory Mapping in CPU Address Space



Notes: 1) ECC word 0 is memory mapped at byte address 4M+0. ECC word 1 is memory mapped at byte address 4M+1 and so on so forth.

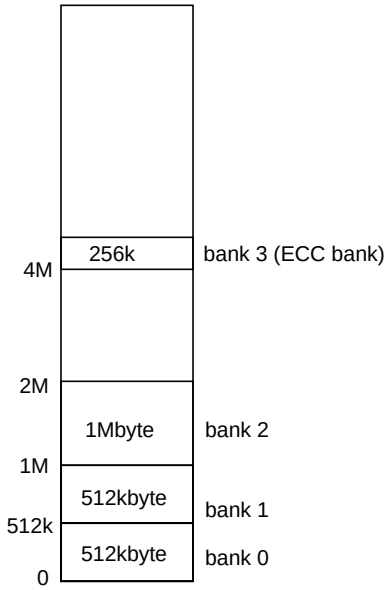
2) ECC word 0 checks the 64-bit data word consisting of 32-bit word 0 and 32-bit word 1

3)

This ECC space starting address at 4Mbytes is a fixed location which allow software development to maintain compatibility. Mapping ECC bank to 4Mbyte starting address can be disabled via FMEDACCTRL1.11 control bit. When disabled, reading from the memory space at 4Mbytes will force the flash wrapper to generate an illegal address signal.

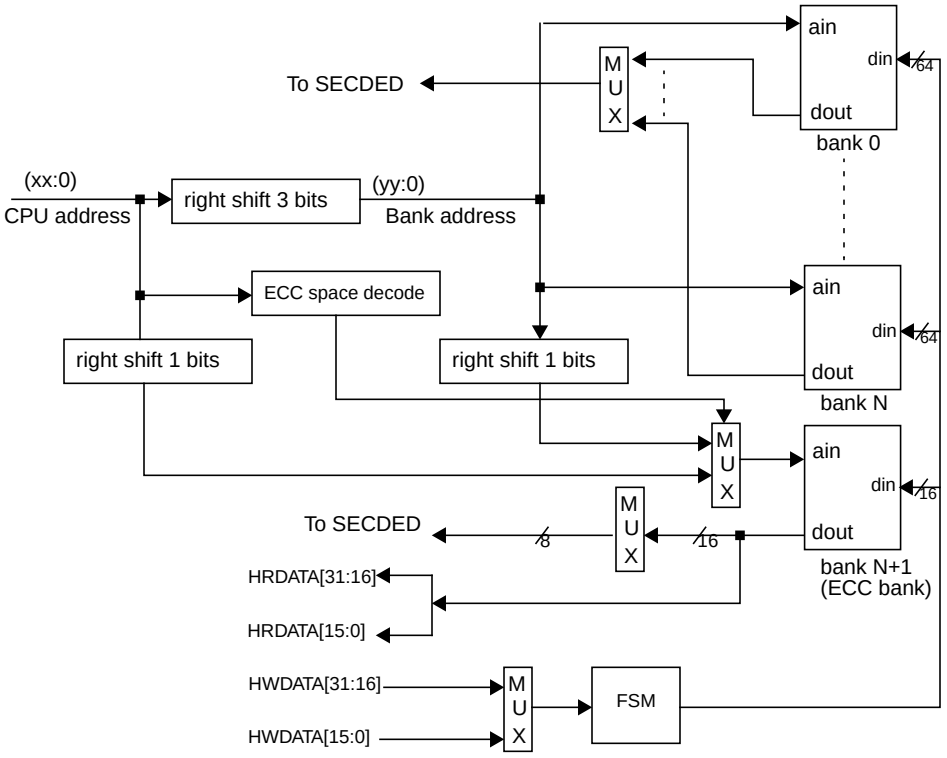
Only 16-bit read and 8-bit read are supported when reading from ECC space. During both 16-bit read and 8-bit read, the flash wrapper will duplicate ECC word x and ECC word x+1 on both the upper and lower 16 bit of the CPU read data bus. Flash wrapper can not support 32-bit read becaue the ECC memory bank is physically only 16-bit.

Figure 5. ECC Bank Memory Map Example



Note: The base address of each bank must be a multiple of its bank size.

Figure 6. Bank Address Generation



Note: To prevent empty spaces from forming in the ECC bank the memory space occupied by the data banks should be contiguous.

4.3.3 Illegal Address Generation

Any un-implemented flash space not decoded to a flash bank will cause an illegal address signal to be generated. The illegal address signal forces the CPU to take either data or instruction abort.

4.3.4 Disabling ECC

Globally the ECC logic is enabled and disabled by an 4-bit EDACEN register. When the four bits are 0101, the ECC logic is disabled. Any non-0101 combination value stored in the EDACEN register will enable the ECC logic. It is advisable to enable ECC logic with 0xA value stored in the EDACEN to prevent any soft error from disabling ECC logic.

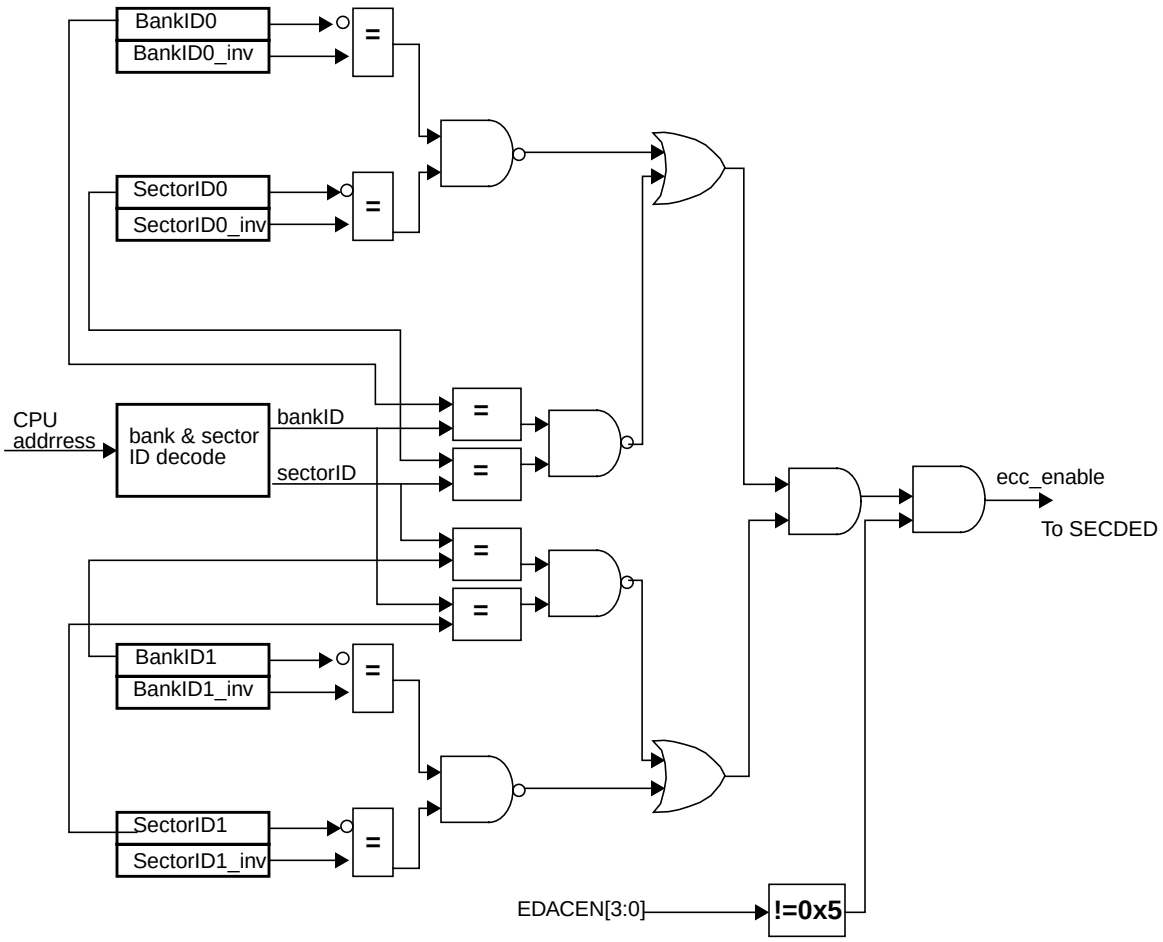
At any given time two different sectors (excluding the ECC bank) can be disabled from ECC checking. This is realized by specifying the bank/sector to be excluded and its inverse value in the sector disable register.

Table 1. Sector Disable Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BankID1_inverse			SectorID1_inverse					BankID1			SectorID1				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BankID0_inverse			SectorID0_inverse					BankID0			SectorID0				

The sector disable register allows two different sectors to be specified for exclusion from ECC checking. Each sector to be excluded consists of two portions: the bank/sector ID value and its inverse value. Only when the programmed bank/sector ID value and its calculated inverted value matches the programmed inverse value will the sector selected be excluded from ECC checking. After reset the sector register is reset to all zeros which means that no sector is excluded from ECC checking.

Figure 7. ECC_ENABLE Generation



4.3.5 Programming ECC Bank

ECC bank is programmed and erased the same way as other flash data banks. During programming, the bank ID must be specified along with the intended address/data are given to the flash wrapper. During erasure both the bank ID and the intended address are given to the flash wrapper. Platform F05 Flash-ECC wrapper contains an embedded FSM which carries out the operations. For details refer to Section 4.4, *Flash Commands*.

4.3.6 ECC Generation Algorithm

Platform F05 Flash-ECC wrapper uses modified Hamming code theory to generate ECC check bits. The ECC code word allow the gross-error condition of all zeros and all ones from memory to be detected.

Table 2. ECC encoding

Participating Data bits																																				
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27
								x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									x	x	x	x	x
x	x	x	x	x	x	x	x																	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x									x	x	x	x	x	x	x	x									x	x	x	x	x
x	x							x	x	x	x	x	x			x	x							x	x	x	x	x	x			x	x			
		x	x	x				x	x	x				x	x			x	x	x				x	x	x				x	x			x	x	x
x		x			x	x		x			x	x			x	x		x			x	x		x			x	x			x	x		x		
			x		x		x		x		x		x	x	x				x		x		x		x		x		x	x	x				x	
x		x	x		x			x	x		x				x	x		x	x		x			x	x		x				x		x			x

Participating Data bits																											Parity[2]	Checkbits[1]
26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
x	x	x																	x	x	x	x	x	x	x	x	Odd	**ECC[7]**
x	x	x																	x	x	x	x	x	x	x	x	Odd	**ECC[6]**
x	x	x									x	x	x	x	x	x	x	x									Odd	**ECC[5]**
			x	x	x	x	x	x			x	x							x	x	x	x	x	x			Odd	**ECC[4]**
			x	x	x				x	x			x	x	x				x	x	x				x	x	Odd	**ECC[3]**
x	x		x			x	x			x	x		x			x	x		x			x	x			x	Odd	**ECC[2]**
x		x		x		x		x	x	x				x		x		x		x		x		x	x	x	Even	**ECC[1]**
	x	x			x		x	x	x			x			x		x	x			x		x	x	x		Even	**ECC[0]**

Notes: 1) Each ECC[x] bit represents the parity bit for the corresponding data bits marked with x in the same row.

2) The ECC check bit is generated as either an XOR(Even) or an XNOR(Odd) of the data bits marked with x in the same row.

Table 3. Syndrome Decode to Bit in Error

Data Error Bit																																				
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0
0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1
1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0
0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0
1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	1	0	1	0	0	1

Data Error Bit																											ECC Error Bit									
26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	07	06	05	04	03	02	01	00		
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	**Bit[7]**	**Syndrome**[1,2,3]
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	**Bit[6]**	
1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**Bit[5]**	
0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	**Bit[4]**	
0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	0	**Bit[3]**	
1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	**Bit[2]**	
1	0	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0	0	0	1	0	**Bit1]**	
0	1	1	0	0	1	0	1	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	**Bit[0]**	

Notes:

1) Syndrome is a 8 bit value which decodes to the bit in error. The bit in error can be a bit among the 64 data bits or a bit among the 8 ECC check bits.
2) For example, if data bit 30 is in error then the syndrome would indicate as 11110001. If ECC check bit 5 is in error then syndrome would indicate as 00100000
3) Syndrome value of 00000000 indicates there is no error. Any other syndrome combinations not shown in the table are either double bit error or un-correctable multibit error

4.3.7 No error occurred

This is the normal condition. No further action is taken. Note that the error correction logic is always in the datapath there will be a negative impact on the flash speed regardless of error occurring or not.

4.3.8 Single error correction

SECDED is capable to correct an erroneous bit if it determines the total number of error bits in the data code word is one. The syndrome is decoded and generates a signal to invert the failing bit. There are three types of

interrupts for which SECDED can generate when a single-bit error is detected and corrected. They are:

- ❏ Interrupt on zero fail
- ❏ Interrupt on one fail
- ❏ Interrupt on error profiling

Note: Regardless if the interrupt is enabled, a correctable error is always corrected if EDACEN is enabled.

4.3.9 Interrupt on Zero Fail

The most common failure mechanism of flash memory is a data retention loss where electrons drain off of the floating gate and results in a zero changing into a one. The SECDED circuit can detect this type of failure. The failing bit can be reprogrammed without the need of erasing flash first, thus restoring the electrons to the floating gate. This effectively fixes the flash and reduces the possibility that later on, a non-correctable failure will occur. The address and failing bit of each correctable error is latched and an interrupt can be generated to signal the program that a failure occurred.

When “Error On Zero Fail Only Enable” bit is set, the “Error On Zero Fail Only” flag can interrupt the processor when an expected 0 value turning into a 1 value. The address and error position are frozen from being updated until the correctable error status flag is cleared by the host system.

Note: Instruction to clear the status flag should be placed at the end of the interrupt subroutine. This is to avoid the address and position registers latching new values during interrupt subroutine.

4.3.10 Interrupt on One Fail

In the less likely event of data gain when a failing bit occurs resulting in a one turning into a zero the flash sector must be erased first before reprogramming.

When “Error On One Fail Only Enable” bit is set, the “Error On One Fail Only” flag can interrupt the processor only when an expected 1 value turning into a 0 value. The address and error position are frozen from being updated until the correctable status flag is cleared by the host system

To generate interrupt for both zero fail and one fail simply enable both “Error on Zero Fail Only” and “Error on One Fail Only” enable bits.

If both enable bits are disabled then no interrupt is generated but SECDED still detects and corrects any correctable error.

4.3.11 Profiling Mode

When profiling interrupt is enabled, each time a correctable error (either error on zero fail or error on one fail) is detected by the SECDED logic the correctable error occurrence counter (SEC_OCCUR) is incremented. A profiling interrupt is generated when the correctable error occurrence counter is equal to the threshold value. The threshold value is stored in a 16 bit threshold (SEC_THRESHOLD) register. The error occurrence counter is frozen when SUSPEND is high.

An error can be either a hard error or soft error. If the number of correctable error reaches a large threshold value in a short period of time then it is a high possibility that the error is a hard error. Although a hard single-bit error is correctable, it does increase the risk that additional soft error on the same word can cause a non-correctable failure.

4.3.12 Interrupt Request Generation

When the SECDED logic block detects a correctable error, the associated error address and error position are immediately stored into registers. During pipeline mode when data are pre-fetched in advanced of an CPU fetch the interrupt is not generated until the word is requested by CPU. When the addressed word is requested by CPU, the associated error interrupt status flag (Error on Zero Fail or Error on One Fail) is set and the interrupt request is generated if the word was detected to have error.

During pipeline mode the error address and position registers may be overridden with new values if the earlier word detected with error is not requested by CPU and another error is detected for the subsequent bank reads. This means that the address and position are overridden as long as the error status flag is not set. Once an error status flag bit is set the error address and position are frozen until the associated error status flag is cleared by CPU.

If profiling is enabled then any correctable error detected will increment the error counter. The counter is incremented immediately when an error is detected regardless if the faulted word is later accessed or not by CPU.

4.3.13 Non-correctable error detection

When there are more than one error bit detected, the SECDED generates a un-correctable error signal on UERR output pin to system module. During an non-correctable error the data is unchanged. The address location of which the un-correctable error is detected is saved in an error address register. This

register is frozen from being updated until it is read by the CPU. The UERR will set a un-correctable error interrupt flag inside system module. Abort is not generated for a non-correctable error due to timing impact on HOLDREQ generation. Inside system module a programmable error counter will make sure that the un-correctable error interrupt is handled within a proper time by the CPU. If the error counter times out then a system reset is issued to the CPU.

When an un-correctable error is detected, the error address is immediately saved to the un-correctable error register. The error signal is not generated to System module until the word is requested by CPU. Once the error signal is generated the un-correctable error address register is frozen until it is read by CPU. Similar to correctable error address register the un-correctable error address register can be overridden with new value if the previous detected un-correctable data word is not requested by CPU and another un-correctable error is detected by SECDED logic.

4.3.14 EEPROM emulation

Three features assist using the flash memory for EEPROM emulation. The first feature is that writing to a 64-bit data register will generate the proper 8-bit check bits in the check bit register. The data register and check bit register are memory mapped and can be accessed through flash wrapper's VBUS interface. Since the VBUS interface is 32-bit and therefore takes two 32-bit writes to input a 64-bit data. Any write to any one of these two registers will trigger a check bit generation.

During the cycle when data register is written to obtain the corresponding check bit value the CPU is waited for one clock cycle if ECC_ENABLE signal is active since instruction fetch can take place at the same cycle.

Second, by setting the "Zero Condition Valid" flag, reading a zeroed 64-bits of memory with an associated zeroed 8-bit check bits will not create an error. This allows the programmer to zero out used portions of the flash when the table is to be moved to a new block.

Third, by setting the "One Condition Valid" flag, the check bits were chosen to allow for the valid state of all data bits one and all check bits one. This allows reading erased memory locations without creating an error.

4.3.15 System Emulation

During emulation when SUSPEND signal is high the data read from memory is still passed to SECDED for correction if ECC_ENABLE is active. If a correctable error is detected then it is corrected but error interrupt is not generated and error occurrence counter is not incremented if in profiling

mode. If an un-correctable error is detected then the raw data is returned without generating un-correctable error signal.

4.3.16 Diagnostic Mode

Flash wrapper can be put in diagnostic mode to verify SECDED logic. There are two diagnostic modes supported by the wrapper. The diagnostic mode is entered via DIAGMODE[1:0] control bits. Diagnostic mode can be enabled while ECC logic is also enabled for normal bank read. Flash wrapper will arbitrate for the usage of ECC logic if conflict occurs between normal bank read checking and diagnostic checking.

Table 4. DIAGMODE Encoding

DIAGMODE[1:0]		Description
0	0	Default after reset. Diagnostic disabled.
0	1	Diagnostic enabled. Data correction mode.
1	0	Diagnostic enabled. Syndrome reporting mode.
1	1	Reserved

When diagnostic mode is disabled with DIAGMODE=00, the user can still write to the 64-bit EEPROM emulation data register. The corresponding check-bit is calculated and captured into ECC check-bit register. Write to ECC check-bit register has no effect.

When in diagnostic data correction mode with DIAGMODE=01, the 64-bit EEPROM emulation data register and the 8-bit ECC check-bit register are used to enter diagnostic data to exercise the SECDED logic. User can apply a diagnostic 64-bit data word with an error in any bit location and compare against a known checkword stored in the ECC check-bit register. When the correctable error is detected, the corrected data is saved back to the 64-bit data register. User can also apply a known 64-bit data word to the 64-bit data register and apply a corresponding 8-bit check word with an error in any bit location, The corrected check word is saved back to the ECC check-bit register. The error position register is also updated to indicate the bit position in error. For single bit correctable error the diagnostic correctable error status bit DCERR is set when a correctable error is detected. For un-correctable error the diagnostic un-correctable error status bit DUERR is set when an un-correctable error is detected. Status bits should be cleared by the user before applying a new diagnostic data.

When in diagnostic syndrome reporting mode with DIAGMODE=10, the resulting syndrome calculated by SECDED is captured into ECC check-bit register. The syndrome can be read by the user and compare with a known

syndrome value. Diagnostic data is not corrected and the error position register is not updated.

Note: Diagnostic data results in error detected during diagnostic mode does not set any interrupt status flag and does not generate any interrupt request or error signal.

During either diagnostic mode a correct sequence of writes to the data register and ECC check-bit register is required. The data register consists of both a 32-bit MSW (most significant word) and a 32-bit LSW (least significant word). The diagnostic is triggered by a write to the ECC check-bit register. When ECC check bit register is written, the 64-bit data word and the 8-bit check word are passed to the SECDED logic. Writing to either the MSW or LSW data word does not trigger the diagnostic event.

4.4 Flash Commands

The MCU F05 Platform Flash-ECC should be programmed, erased and verified only by using the F05 Flash API Library Functions. These functions are written, compiled and validated by Texas Instruments. The information provided in this document is intended to help explain how these functions work. It is not intended to provide sufficient detail for implementing programming or erasing functions.

The flash module contains a Command State Machine (CSM) to perform program, erase, and validate operations. The CSM supports the commands shown in Table 5. At the end of a system reset, the CSM is in Idle mode. The completion of any command also returns the CSM to Idle mode.

Table 5. Flash Command Summary

Command Operand	Description	Write Cycles Required
0x0010	Program	2
0x0020	Erase Sector	2
0x0040	Clear Status Register	1
0x0080	Suspend (program or erase)	1
0x0200	Resume Programming	1
0x0400	Resume Erasing	1
0x0800	Program OTP	2
0x1000	Validate Sector	2

Command writes (as well as OTP reads) can be performed only when pipeline mode is off. This is necessary to prevent data stored in the pipeline buffers from becoming inconsistent with the data stored in memory.

The program, erase, and validate sector commands require two write cycles, whereas the remaining commands require only one write cycle.

After a program, erase, or validate sector command has been written to the flash module, the module waits indefinitely for the second write cycle to latch the address and/or data; therefore, the valid data write cycle must have valid data, whether it immediately follows the command or not.

The control port can be used to read or write registers, or bank reads can be performed between the command and the data for the program or erase command.

The BUSY status bit becomes active upon receipt of any command except for the clear status command.

After an erase or program command has been issued, the operation cannot be aborted except by resetting the flash module. The operation can, however, be allowed to complete or be suspended. A suspended operation, likewise, can only be aborted by resetting the flash module.

4.4.1 CPU Operations Required for Executing Commands

Before the MCU F05 Platform Flash-ECC can execute a command, the host CPU must do the following:

1) Select the desired bank to be programmed by writing BANK[2:0] of FMMAC2
2) Select one or more bits in FMBSEA or FMBSEB to disable Level 1 protection for the particular sector to be erased/written
3) Write the correct four 32-bit FMPKEY words (if applicable)
4) Clear the READOTP bit in the FMREGOPT register
5) Write an operand, as shown in Table 5, to any location in flash memory which initiates the operation

For operations that apply to specific address locations (program, erase, and validate) the host must also:

6) Write the desired data half-word to the appropriate address in the selected flash bank

After these operations have been completed in sequence, the command is performed. These flash commands are described in the following sections in more detail.

4.4.2 Program Sector Command

If not already set up, the host must follow the procedure outlined in Section 4.4.1 above. Writing the program operand, as shown in Table 5, to any location in flash memory initiates the program operation. This must be followed by writing the desired data half-word to the appropriate address in the selected flash bank.

Note: Program Data to the Flash Module in Half-Words

All data programmed into the MCU F05 Platform Flash-ECC must be done one half-word (16 bits) at a time.

The flash module checks the sector protection status and, if allowed, proceeds to program and verify the data. Once the command is received, the busy flag is held active.

While the busy flag is active, the flash module responds only to the suspend command. If an attempt is made to read a bank being programmed, invalid data is returned. Reads may be performed correctly on other banks. If the program operation is suspended and the suspended bank is read, then only data other than the address that was being programmed should be assumed valid.

If the program operation is suspended, the busy flag becomes inactive within 2.0µs of receiving the suspend command.

During the program operation, 3.3V V_{DD} must remain within the appropriate range; otherwise, the error flag 3VSTAT is set in the status register FMMSTAT.3.

When the program command is complete, the user should parse the Status register to verify that the command was successful. Upon completion, the flash module resets the CSM state to Idle mode; therefore, the program command must be reissued before another half-word is written to flash memory.

4.4.3 Erase Sector Command

If it is not already set up, the host must follow the procedure outlined in Section 4.4.1. Writing the erase operand, as shown in Table 5, to any location in flash memory, initiates the erase operation. This is followed by writing any data to the selected sector address (an address anywhere within the sector to be erased), in the flash memory.

The flash module checks the sector enable status and, if allowed, proceeds to erase and verify the data. Once the command is received, the busy flag is held active.

While the busy flag is active, the flash module responds only to the suspend command. If an attempt is made to read a bank being erased, invalid data is returned. Reads can be performed on other banks. If the erase operation is suspended and the suspended bank is read, then the entire sector's data is assumed invalid. Read data from other sectors and banks is invalid due to the partial erasure of this sector.

If the erase operation is suspended, the busy flag becomes inactive within 2.0µs of receiving the suspend command.

Upon completion of the erase command, the user should parse the Status register to verify that the command was successful.

During the erase operation, 3.3V V_{DD} must remain within the appropriate range; otherwise, the error flag 3VSTAT is set in the status register FMMSTAT.3.

When the erase operation is complete, the flash module resets the CSM state to Idle mode; therefore, the erase command must be reissued before another sector can be erased.

4.4.4 Suspend/Resume Commands

The suspend command allows erase or program operations, whichever is active, to be suspended so other operations can be performed on the affected bank. It is also possible to suspend a program operation so that a sector in a different bank can be erased, or suspend an erase operation so that information may be programmed in a different bank. The suspend command is acted upon only if the BUSY bit of FMMSTAT is high and the flash module is performing an erase or program operation. Issuing a suspend command when BUSY is low has no effect on the flash module.

Note: Flash Module Ignores All But Suspend Command When BUSY Bit is High

If a command other than suspend is issued while the BUSY bit is high, it is ignored.

It is possible to suspend an operation at certain points in the operation of the state machine so that the operation actually finishes and the suspend flags are never set. Issuing a resume command when suspend is not active has no effect on the flash module.

Only one operation can be active at a time. Also, a program operation cannot be initiated while a previous program operation is suspended, and an erase operation cannot be initiated while a previous erase operation is suspended. The flash module ignores any command that violates these conditions.

4.4.5 Clear Status Command

The Status register allows the user to determine whether an erase/program operation is successfully completed, in progress, suspended, or failed.

If not already set up, the host must follow the procedure outlined in Section 4.4.1. Writing the clear status operand, as shown in Table 5, to any location in flash memory, executes the clear status operation.

If the BUSY bit is low, this operation clears the following status bits: SLOCK (sector locked), CSTAT (command status), 3VSTAT (3.3V V_{DD} status), INVDAT (invalid program data) in the FMMSTAT register.

4.4.6 Program OTP Command

The flash module may have one or more OTP sectors, up to one OTP sector per flash bank. A minimum of one is required for production and test purposes. Each OTP sector contains 2K bytes. The bits can be programmed via the program OTP command. Once the command is issued, the bank identifier, address within the OTP sector, and the data must be specified.

The program OTP flow is the same as the program half-word flow. It can be suspended and resumed.

4.4.7 Validate Sector Command

A validate sector command has been provided to enable the host to determine if a sector contains depleted bits. A depleted bit may occur only in the unlikely event that an erase operation has been started, but not completed. The following conditions could leave depleted bits in a sector:

- ❏ An erase is in progress when power is removed
- ❏ An erase suspend is active when power is removed
- ❏ A flash module reset interrupts an erase operation
- ❏ A flash module reset occurs while an erase suspend is active

After a flash module reset, including power up, the host should perform a validation check on sectors which may have been corrupted to check that no sectors contain depleted bits. It is not required to check sectors which are not programmed in the field, as they cannot have depleted bits. It is, however, possible for depleted bits in one sector to cause bits in another sector to read incorrectly; therefore, it is vital that depleted bits be corrected.

Note: There Is No Suspend for the Validate Sector Command

The *validate sector* command cannot be suspended.

4.5 Data Security

Data security against either accidental or deliberate access by unauthorized agents is built into the flash module in two levels of data security: Level 1 security allows each sector to be individually protected from any access other than read; level 2 security protects the entire module from non-read access using four optional 32-bit protection keys.

4.5.1 Sector Enable

Sector enable (registers BSEA and BSEB) is a means of preventing data from being modified within a sector. Since the flash module may store permanent and/or semi-permanent program code and/or data this capability is provided. A sector is protected if data within that sector is prevented from being modified (the sector-enable bit for that sector is cleared). Currently the MCU F05 Platform Flash-ECC supports a maximum of 32 sectors per bank.

Flash memory can be programmed or erased only if the specified sector is enabled. If the sector is protected, then the state machine of the flash module halts and sets the status bit SLOCK in FMMSTAT.0.

At power up, all of the sector-enable bits are initialized so that the flash memory location cannot be modified. Sector enable is a feature used only by the flash module when erasing or programming flash memory.

When an erase or program operation resumes from a suspended state, the sector-enable bit is checked again as though it were a new erase or program command.

4.5.2 Four-Word Protection Keys

If this option is present, the CPU reads the four stored protection keys out of the flash bank one at a time and into a register in the flash module. After the CPU loads each key from the bank to the control logic, the CPU must load an identical user key into the FMPKEY control register. The CPU must load and match all four keys before any program or erase command can be sent to the flash module.

To enable the module for programming, the CPU must load each stored key value from the bank to the control logic by performing a normal read access to one of the four protection key addresses in the Flash module. The CPU must then load the matching user key value into the FMPKEY register. This process is repeated until all four keys are loaded and matched. The control logic monitors which keys have been matched, so the CPU can not gain write access until it loads and matches all four keys at least once without any intervening mismatch.

If the CPU writes a mismatching key at any time (that is, if the user key does not match the key that was most recently loaded from the bank to the control logic), then all key match states are cleared and the CPU must *reload and rematch all four keys again* to gain write access to the module. This feature can be used to disable write access after programming is completed.

After the CPU has successfully loaded and matched all four keys, flash write access is enabled and the PROTL2DIS flag (FMBBUSY.15) is set until either

a device reset occurs or until the CPU writes a mismatching key to the FMPKEY register.

To store the key values, the CPU programs the key data into the bank by performing normal program and erase operations on these four protection key addresses. The key values are stored in the bank as ordinary data, so the CPU must provide the correct keys before it can perform any program or erasure of the key values.

When a new device is delivered to the customer, the keys will be all ones, so keys of all ones should be used to enable flash writes for the first time. Once the keys are changed in the Flash bank, the CPU must deliberately write a mismatching key value to FMPKEY in order to disable further programming until the new key values have been loaded and matched. In other words, the flash module remains enabled for the remainder of this programming session even though the keys have been modified in flash.

The difference between flash protection key read accesses and other reads is that the key data does not propagate to the CPU data bus *until the correct keys are written* and the user has taken all required steps to gain programming access. This is intended to prevent unauthorized discovery of the stored keys by reading them out via the CPU: only a user who already knows the keys can gain access to them.

The availability of this protection feature and the location of the four protection keys depend on the specific device being used (as specified in the specific TMS470R1x device data sheet). If this feature is not available, then the four protection key addresses in the module are available for normal memory access.

4.6 Automatic Power-down of Flash Banks

The flash module provides a mechanism to automatically power down flash banks after they have not been accessed for some user programmable time. Special timers automatically sequence the power up and power down of each bank independently of each other. The charge pump module has its own independent power up/down timers.

During global low power mode when SYS_LPM=1 the flash module which includes the wrapper logic, banks, and pump module are put into sleep mode. If there is an operation such as a write then the flash module waits until the operation to finish before it enters sleep mode. Flash wrapper uses nLP_RDY handshake signal to indicate it is ready.

When SYS_LPM signal goes low again the banks and pump module will go through the normal power up sequence from sleep to active power state. CPU

access to the flash module during the power up sequence will be held in wait state.

4.6.1 Active Grace Period

Active grace period (AGP) is a feature which the host can use to optimize the flash module power consumption/access time trade-off. Faster access times are associated with higher power modes of operation. At one extreme, the power control logic could attempt to reduce power consumption by putting the banks and charge pump into a low-power mode immediately at the end of every flash access; however, if accesses are a few cycles apart, this can actually increase power consumption over if the flash had remained powered, because the banks and charge pump consume more power during flash startup and access.

Active grace periods (supported for each bank independently in addition to the charge pump module) allow the banks and/or charge pump to be maintained in active mode for a specified period following an access. This is done in anticipation of another read within the AGP time, to allow the subsequent read to have a faster access and spend less time dissipating power than if the bank went into one of the low power modes immediately. On the other hand, if the next access does not occur within the AGP time, the power control logic can automatically put the bank and/or charge pump into a low-power mode to reduce power consumption during long periods of inactivity.

AGP is realized by a set of host-programmable counters which keep the flash bank or charge pump in active mode until the counter expires, at which time the bank or charge pump reverts to its fallback power mode. See register descriptions for FMBAC1, FMBAC2, FMMAC1, FMMAC2, and FMPAGP.

The bank and/or charge pump are kept active in anticipation of another read. Assuming AGP is being utilized, the bank AGP counter is set, keeping the bank active, while an access is in progress. The counter begins counting when no more accesses are in progress. If the AGP timers have not timed out and another access occurs then there is a substantial performance gain compared to the access when the bank is not active (either the bank is in standby or Sleep mode). If the AGP counter times out, the bank or charge pump is put into a host programmable fallback power mode. The host can program the fallback power mode to be standby or sleep mode to reduce power consumption, or program it to be active mode to keep the bank active regardless of counter settings (default).

The ECC bank power control logic is identical to other banks in which different fallback power mode can be selected. It is possible to have flash data banks in different fallback power mode than the ECC bank. When an access is made to a flash bank, the flash wrapper power control logic will wait for both the

addressed bank and the ECC bank to become active before returning the data to the CPU if EDACEN is enabled.

The charge pump AGP counter remains in its initialized state when any one of the banks is active, including the AGP counter of the bank. The charge pump AGP counter begins counting when all banks have become inactive.

The bank and charge pump active grace period counters count at SYSCLK frequency.

4.6.2 Power Mode Control

The primary contributors to flash module power consumption are the flash banks, and the charge pump. This section deals with how the flash wrapper handles the control of the different power modes of the flash banks and charge pump.

A couple of the components of the module power reduction have been discussed. These are the bank fallback power bits in the FMBAC1 register, the charge pump fallback power bits in the FMMAC2 register, and the BAGP and PAGP operation. The fallback power control bits contain the bank and charge pump modes, which become active upon time-out of the AGP counters described in Section 4.6.2. Any access to a flash bank causes the bank and charge pump to go into active mode, regardless of their current state. Also, any erase, program, or validate sector command causes the charge pump to become active.

If the charge pump is in sleep mode when the flash access begins, the power mode control logic automatically sequences the charge pump to standby mode, then to active mode. Also, if any bank is active or in standby mode, the charge pump is active, independent of the charge pump fallback power mode.

The host can override the power control functions of the flash module by setting all of the AGP counters to zero. In this case, the power mode control logic still sequences the pump through standby mode automatically if needed, and it activates the pump automatically if any bank is put into any power mode other than sleep mode.

4.7 Wait States

The number of wait states can vary depending on the type of read access performed. The different types of read accesses are defined below:

- *Initial Access* - First address read after initialization of the flash. Typically after reset.

- *Sequential Access* - Instruction accesses the very next flash address from the previous instruction's flash address.
- *Random Access* - Instruction accesses a non-sequential flash address from the previous instruction's address.

Wait states are added to a read access when either the flash bank or charge pump are not active, or when the flash bank can not have data valid at the frequency demanded by the host. The flash module generates an internal ready signal for each bank based on the values in the wait state registers which depend on the status of the bank, the charge pump, and the data rate.

The bank has three wait state generators and the pump has two wait state counters. Both the bank and the charge pump have standby and sleep counters which are initialized when a transition is made from non-active mode to active mode. The bank also has a wait state counter to allow for a faster clock than the flash data rate. All wait state generators are clocked by the flash system clock.

The outputs of the charge pump counters and bank counters are combined. The resulting signal generates the wait states and ready signal of each bank when an access is in progress.

4.8 VDD3V Out of Range Check

The VDD3V out of range status bit informs the host if there was a low supply during an erase or program operation. A comparator in the charge pump module sets the 3VSTAT bit if the 3.3V supply is less than 2.4V for more than three SYSCLK cycles; otherwise it is cleared.

5 Control Registers

This section covers the MCU F05 Platform Flash-ECC control register memory map and a basic description of each register and its bits.

5.1 Memory Map

The MCU F05 Platform Flash-ECC uses several registers to control the various modes of operation, numbers of wait states, and bank selection. All user flash registers are shown in Table 6 and Table 7. The addresses shown are relative offsets from the base address, which depends upon the device being used. See the specific TMS470R1x device data sheet for the register base address.

Table 6. 32-bit Flash Memory Registers

Offset Address†	Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x0100	FMRE-GOPT	Reserved															
		Reserved											SPOTP	RD MRGN1	RD MRGN0	READ OTP	ENPIPE
0x0108	FMBBUSY	PROTL 2DIS	Reserved							BUSY[7:0]							
0x010C	FMPKEY	PKEY[31:16]															
		PKEY[15:0]															
0x0110	FMEDACC TRL1	Reserved															
		Reserved				ECC-MAP	EOFEN	EZF EN	EP EN	Reserved		EO CV	EZ CV	EDACEN[3:0]			
0x0114	FMEDACC TRL2	Reserved															
		SEC_THRESHOLD[15:0]															
0x0118	FMEDAC-CNT	Reserved															
		SEC_OCCR[15:0]															
0x011C	FMCER-RADDR	Reserved								CERRADDR[23:16]							
		CERRADDR[15:0]															
0x0120	FMCER-RPOSI-TION	Reserved															
		Reserved							ERR TYP E	0	0	SERRPOSITION[5:0]					
0x0124	FMEDAC-STATUS	Reserved															
		Reserved													EO FS	EZ FS	EPS
0x0128	FMUER-RADDR	Reserved								UERRADDR[23:16]							
		UERRADDR[15:0]															
0x012C	FMEMU DMSW	EMUDW[63:48]															
		EMUDW[47:32]															
0x0130	FMEMU DLSW	EMUDW[31:16]															
		EMUDW[15:0]															
0x0134	FMEMU ECC	Reserved															
		Reserved								EMUECC[7:0]							
0x0138	FMSEC-DIS	bankID1_inverse[2:0]			SectorID1_inverse[4:0]					bankID1[2:0]			SectorID1[4:0]				
		bankID0_inverse[2:0]			SectorID0_inverse[4:0]					bankID0[2:0]			SectorID0[4:0]				
0x013C	FMDIAGC-TRL	Reserved															
		Reserved						DU-ERR	DCE RR	Reserved						DIAGMODE	

Table 7. 16-bit Flash Memory Registers

Offset Address	Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	FMBAC1	BAGP								BSTDBY						BNKPWR	
0x0004	FMBAC2	OTP PROT DIS	BSLEEP							WAIT				WAIT			
0x0008	FMBSEA	BSEA															
0x000C	FMBSEB	BSEB															
0x0010	FMBRDY	Reserved										BANK RDY	Reserved				
0x0014	FMPRDY	Reserved						PUMP RDY	Reserved								
0x0018	FMMAC1	PROTL1 DIS	PSLEEP														
0x001C	FMMAC2	PSTDBY											PMPPWR		BANK		
0x0020	FMPAGP	PAGP															
0x0024	FMMSTAT	Reserved							BUSY	ERS	PGM	INV DAT	CSTAT	3V STAT	ESUSP	PSUSP	SLOCK

5.2 Register Access

The flash module control registers can only be read and/or written by the CPU while in privilege mode.

Byte, half-word and word writes are supported to all registers. Remember that half-word access and byte access are affected by big or little endian configuration.

Note: All reserved fields will read as zeros.

5.3 Option Control Register (FMREGOPT)

FMREGOPT is a word-access only register. It supports OTP sector access, margin testing, and pipeline mode. There is only one FMREGOPT register for the entire MCU F05 Platform Flash-ECC.

Bit	31	5	4	3	2	1	0
0x0100	Reserved		Reserved	RD MRGN1	RD MRGN0	READ OTP	EN PIPE
	0		RWP-0	RWP-0	RWP-0	RWP-0	RWP-d

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset, -d: Device specific value

Bits 31:5 **Reserved**

Read values are zeros. Writes have no effect.

Bit 4 **Reserved**

This bit is reserved for TI's use. **Do not set this bit** as it will cause the user OTP sector to be disabled even when otherwise enabled by READOTP.

Bit 3 **RDMRGN1.** Read Margin 1.

When set, enables Read Margin 1 mode (RDMRGN1 is overridden by RDMRGN0).

Bit 2 **RDMRGN0.** Read Margin 0.

When set, enables Read Margin 0 mode (RDMRGN0 mode overrides RDMRGN1 mode).

Bit 1 **READOTP.** Read OTP Sector.

When set, this bit enables reading from the OTP sector. The starting address of the OTP sector is located at the relative flash module address 0x0000 in the first bank of the flash module. Clear this bit should after the OTP sector is read to ensure that flash programming functions normally.

Bit 0 **ENPIPE.** Enable Pipe Mode.

Pipeline mode is active when ENPIPE is set and the MCU F05 Platform Flash-ECC is not in halt mode. Pipeline mode is overridden in halt mode. The default value of ENPIPE is device specific. See the device-specific datasheet for the reset state of ENPIPE.

Note: Command Writes and OTP Reads Performed Only in Pipeline Mode

Command writes and OTP reads can be performed only if pipeline mode is off. This is necessary to prevent data stored in the pipeline buffers from becoming inconsistent with the data stored in memory.

5.4 Bank Busy Register (FMBBUSY)

FMBBUSY is a word-access read-only register. It supports checking the busy status of all banks in parallel. The MCU F05 Platform Flash-ECC has only one FMBBUSY register.

Bit	31 - 16	15	14 - 8	7	6	5	4	3	2	1	0
0x0108	Reserved	PROTL2DIS	Reserved	BUSY[7:0]							
	0	R-0	0	R-0							

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 31:16 **Reserved**

Read values are zeros. Writes have no effect.

Bit 15 **PROTL2DIS.** Protection Key Level 2 Disabled Flag.

This bit is read-only and when set, it indicates that all four protection keys have been written correctly. If any of the 128 protection key bits are incorrect then PROTL2DIS is zero. Writes to this bit have no effect.

Bits 14:8 **Reserved**

Read values are zeros. Writes have no effect.

Bit 7-0 **BUSY[7:0].** Bank Busy.

This read-only location allows the CPU to determine if any flash bank is busy performing an OTP read, a command operation, or in the process of being reset. It displays the states of the BUSY signals from each bank simultaneously. Each bit corresponds to one flash bank. These bits are read-only. Writes have no effect.

5.5 Protection Key Register (FMPKEY)

FMPKEY is a word-access only register. It controls access protection for the entire MCU F05 Platform Flash-ECC. The MCU F05 Platform Flash-ECC has only one FMPKEY register.

Bit	31	0
0x010C	PKEY[31:0]	
	RWP-0	

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 31-0 **PKEY[31:0].** Protection Key.

These bits receive and store the four 32-bit protection keys from the CPU for the four 32-bit protection key feature. See Section 4.5.2, *Four-Word Protection Keys*.

5.6 FMEDACCTRL1 (Error Detection and Correction Control Register1)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x0110	Reserved															
	-0															
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved				ECC<MAP	EOFEN	EZFEN	EPEN	reserved		EOCV	EZCV	EDACEN			
	-0				RWP-0	RWP-0	RWP-0	RWP-0	-0		RWP-0	RWP-0	RWP-0x5			

U = Undefined; *-n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 31:12 **Reserved**

Bits 11 **ECCMAP: ECC Bank Memory Map**

0 = Disable. Read from the 4Mbyte offset will result in illegal address

1 = Enable. ECC bank is memory mapped to 4Mbytes offset from flash memory base address.

Bits 10 **EOFEN: Error on One Fail Only Enable.**

When this bit is set, only correctable errors where a one reads as a zero will generate an interrupt. Sector erase is required to restore a bit from zero to one.

0 = Disable.
1 = Enable.

Bits 9 **EZFEN: Error on Zero Fail Only Enable.**

When this bit is set, only correctable errors where a zero bit reads as a one will generate an interrupt. This bit is used in systems where the user wants to restore failing zeros by reprogramming the failing bit. Failing ones can not be restored.

0 = Disable.
1 = Enable.

Bits 8 **EPEN: Error Profiling Enable.**

0 = Disable.
1 = Enable. When enabled the correctable error interrupt is generated when number of occurrences of correctable bit error detected and corrected has reached the threshold value defined in SEC_TRESHOLD register

Bits 7:6 **Reserved**

Bits 5 **EOCV: One Condition Valid.**

This bit is used to allow the condition of all 64 data bits and the corresponding 8 check bits to be one and not to create error. This bit is used to allow reading of an erased location without generating error. Error counter for profiling does not increment if EOCV is set and all ones is detected.

0 = Disable.
1 = Enable.

Bits 4 **EZCV: Zero Condition Valid.**

This bit is used to allow the condition of all 64 data bits and the corresponding 8 check bits to be zero and not to create error. This bit is used to support zeroing of used tables in EEPROM emulation. Error counter for profiling does not increment if EZCV is set and all zeros is detected.

0 = Disable.
1 = Enable.

Bits 3:0 **EDACEN: Error Detection and Correction Enable.**

0101 = Disable. Error Detection and Correction is disabled
all others = Enable. Error Detection and Correction is enabled

Note: It is recommended writing “1010” to enable EDACEN to guard against soft error from flipping EDACEN to a disable state.

5.7 FMEDACCTRL2 (Error Detection and Correction Control Register2)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x0114	Reserved															
	-0															
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	SEC_THESHOLD															
	RWP-0															

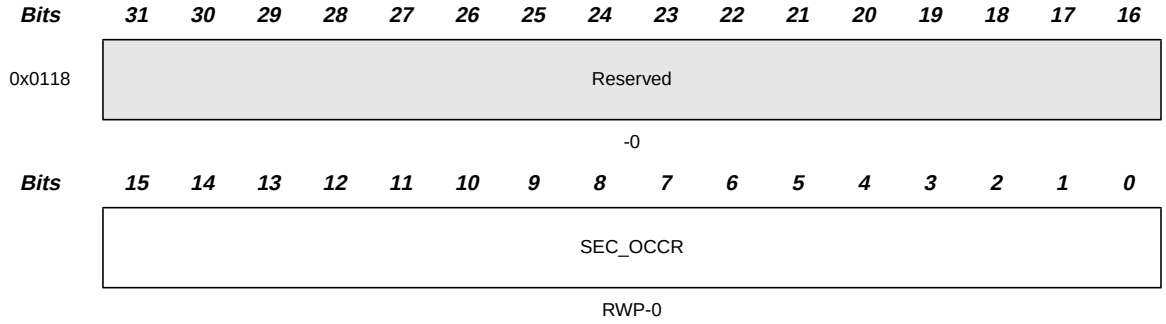
U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 31:16 **Reserved**

Bits 15:0 **SEC_THRESHOLD: Single Error Correction Threshold**

This register contains the threshold value for the SEC (single error correction) occurrences before CE_INT interrupt request is generated.

5.8 FMEDACCNT (Error Detection and Correction Counter Register)



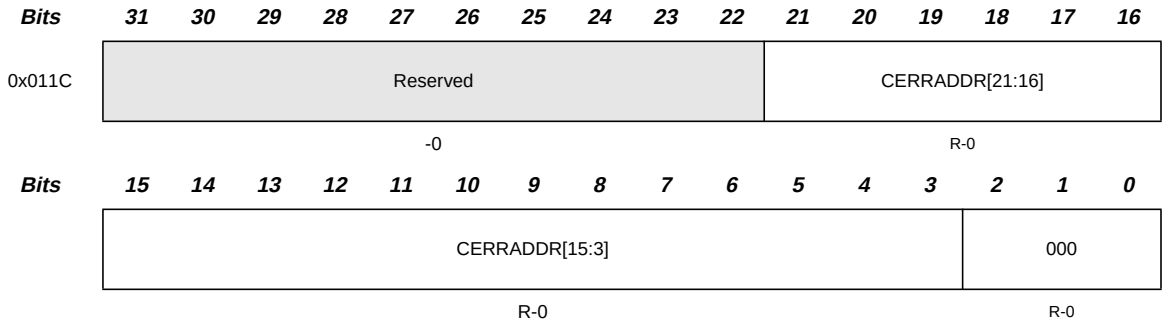
U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 31:16 Reserved

Bits 15:0 SEC_OCCR: Single Error Occurrence Counter

This 16 bit counter contains the number of correctable error occurrences. Write to this register with any value will reset the counter to all zeros. The counter resets to 0 when it is greater or equal to the threshold value and continues to increment if it detects error again. This counter is frozen during suspend.

5.9 FMCERRADDR (Correctable Error Address)



U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 31:22 Reserved.

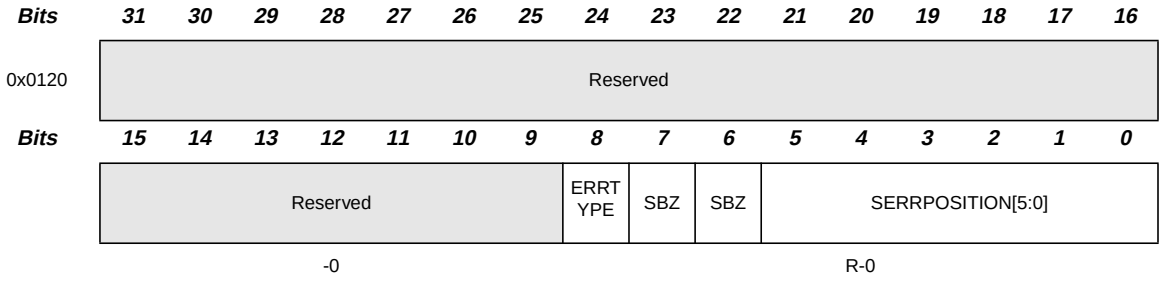
Bits 21:0 CERRADDR: Error Address

CERRADDR records the CPU logical address of which a correctable error is detected by either SECDED logic block. This register is frozen from changing during suspend mode.

Since ECC is checked on 64 bit data, the address captured is aligned to a 64-bit boundary with bit[2:0] tied to 0.

Note: The error address is captured when either EOFEN or EZFEN enable bit is set. During error profiling mode when only EPEN is set the error address is not captured if an correctable error is detected.

5.10 FMCERRPOSITION (Correctable Error Position Register)



U = Undefined; R=Read, WP=Write in privilege mode,-*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 15:9 **Reserved**

Bits 8 **ERRTYPE: Error Type.** This bit indicates whether the single error detected is a data bit error or check bit error.

0 = The error is a data bit error
1 = The error is a check bit error

Bits 7:6 **These two bits are always read zero.**

Bits 5:0 **SERRPOSITION[5:0]: Single Error Position**

SERRPOSITION records the binary encoded error position of which a single error is detected. Error position is captured into SERRPOSITION when a single bit error is detected. Following examples illustrate the error position being captured in FSERRPOSITION.

FSERRPOSITION[8:0] = 0_0000_0000 indicates data bit[0] is in error.

FSERRPOSITION[8:0] = 0_0000_0111 indicates data bit[7] is in error.

FSERRPOSITION[8:0] = 0_0000_1111 indicates data bit[15] is in error.

FSERRPOSITION[8:0] = 0_0001_1111 indicates data bit[31] is in error.

FSERRPOSITION[8:0] = 0_0011_1111 indicates data bit[63] is in error.

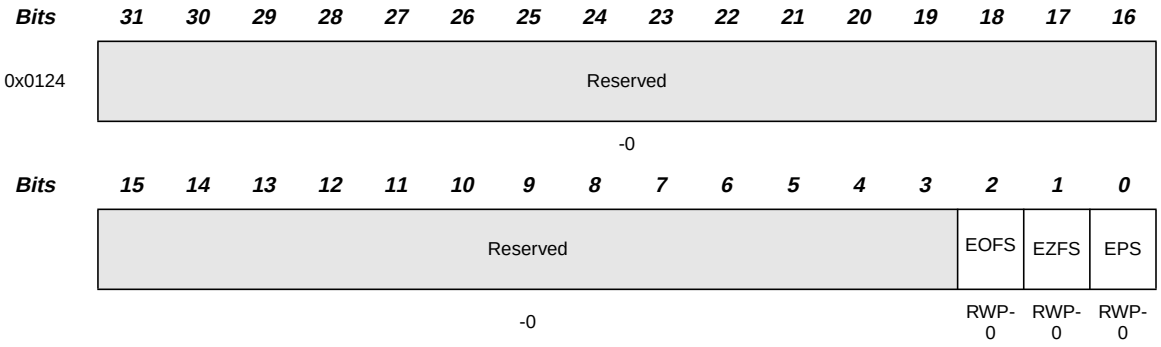
FSERRPOSITION[8:0] = 1_0000_0000 indicates ECC bit[0] is in error.

FSERRPOSITION[8:0] = 1_0000_0111 indicates ECC bit[7] is in error.

This register is frozen from changing during suspend mode.

When diagnostic data correction mode is enabled the error position register will also record the error position bit if a correctable error is detected.

5.11 FMEDACSTATUS (Error Status Register)



U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 31:3 Reserved

Bit 2 EOFS: Error On One Fail Status Flag

If EOFEN (Error on One Fail Only Enable) bit is set then EOFS flag is set when a correctable error is detected and the bit at fault is read as 0 and corrected by either SECDED to be 1. This bit is cleared by writing a '1' to it.

0 = Error on One Fail is not detected.
1 = Error on One Fail is detected.

Bit 1 EZFS: Error On Zero Fail Status Flag

If EZFEN (Error on Zero Fail Only Enable) bit is set then EZFS flag is set when a correctable error is detected and the bit at fault is read as 1 and corrected by either SECDED to be 0. This bit is cleared by writing a '1' to it.

0 = Error on Zero Fail is not detected.
1 = Error on Zero Fail is detected.

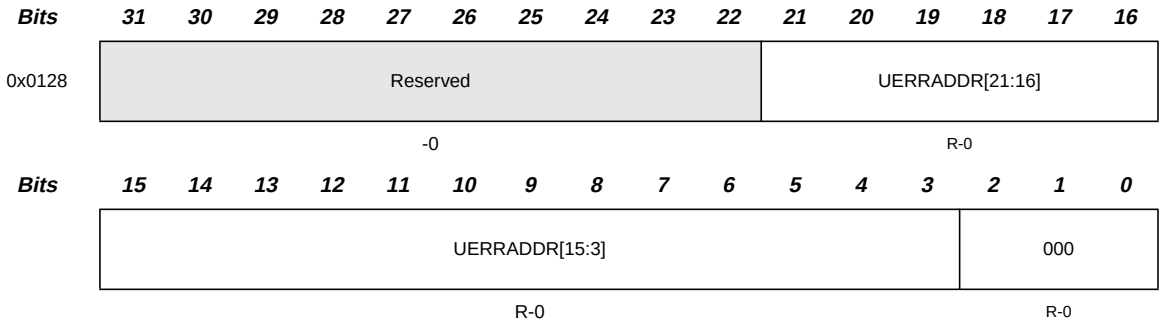
Bit 0 **EPS: Error Profiling Status Flag**

EPS flag is set when the number of occurrences of correctable error detected and corrected by either SECDEDx has reached the programmed threshold value stored in SEC_TRESHOLD register when EPEN (Error Profiling Enable) is set. This bit is cleared by writing a '1' to it.

0 = Profiling error is not detected.
1 = Profiling error is detected.

Bit 7:0 **Reserved.**

5.12 FMUERRADDR (Un-correctable Error Address)



U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 31:27 **Reserved.**

Bits 26:0 **UERRADDR: Un-correctable Error Address**

UERRADDR records the CPU logical address of which an un-correctable error is detected during SECDED scheme. When parity scheme is selected, the UERRADDR captures the error address when a parity mismatch is detected. This error address is frozen from begin updated until it is read by the CPU. During emulation mode when SUSPEND is high this address is frozen even when read.

Since ECC is checked on 64 bit data, the address captured is aligned to a 64-bit boundary with bit[2:0] tied to 0.

5.13 FMEMUDMSW (EEPROM Emulation Data Most Significant Word Register)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x012C	EMUDW[63:48]															
	RWP-0															
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EMUDW[47:32]															
	RWP-0															

U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bit 31:0 **EMUDW[63:32].** EEPROM Emulation Most Significant Data Word.

This register contains the upper 32 bits of the 64-bit EEPROM emulation data word. If diagnostic mode is disabled and this register is written the corresponding ECC word is updated in FMEMUECC register. If diagnostic data correction mode is enabled then EMUDW[63:0] can be used to apply a diagnostic data to the SECDED logic and EMUDW[63:0] is updated with the corrected diagnostic data if a correctable error is detected.

5.14 FMEMUDLSW (EEPROM Emulation Data Lest Significant Word Register)

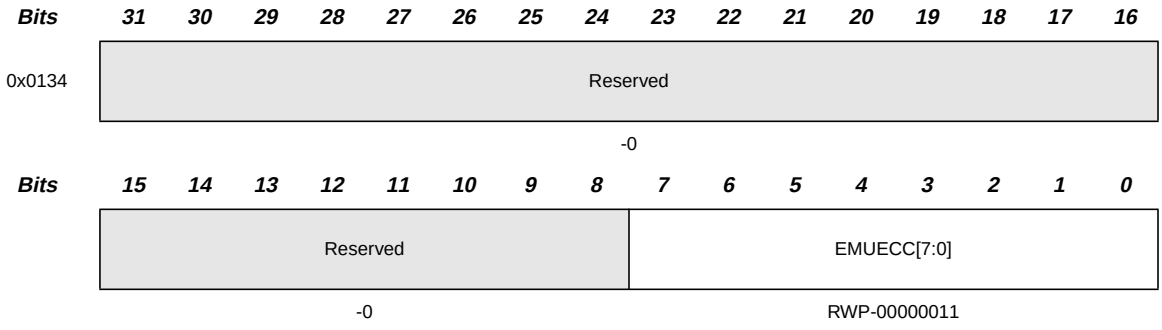
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x0130	EMUDW[31:16]															
	RWP-0															
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EMUDW[15:0]															
	RWP-0															

U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bit 31:0 **EMUDW[31:0].** EEPROM Emulation Least Significant Data Word.

This register contains the lower 32 bits of the 64-bit EEPROM emulation data word. When this register is written, the corresponding ECC word is updated in FEMUECC register.

5.15 FMEMUECC (EEPROM Emulation ECC Register)



U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bits 31:8 Reserved.

Bits 7:0 EMUECC:

If diagnostic mode is disabled and either FEMUDMSW or FEMUDLSW register is written the flash wrapper will return the corresponding 8-bit check bits stored in FMEMUECC register. If diagnostic data correction mode is enabled then EMUECC[7:0] can be used to apply a diagnostic ECC data to the SECDED logic and EMUECC[7:0] is updated with the corrected diagnostic ECC data if a correctable error is detected. If diagnostic syndrome mode is enabled then this register will be updated with the calculated syndrome value.

5.16 FMSECDIS (Sector Disable Register)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x0138	bankID1_inverse[2:0]			SectorID1_inverse[4:0]					bankID1[2:0]			SectorID1[4:0]				
	RWP-0x00											RWP-0x00				
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	bankID0_inverse[2:0]			SectorID0_inverse[4:0]					bankID0[2:0]			SectorID0[4:0]				
	RWP-0x00											RWP-0x00				

U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bit 31:29 **BankID1_inverse[2:0].** Inverted Bank Number of which the sector is to be disable for ECC checking

Bit 28:24 **SectorID1_inverse[4:0].** Inverted Sector Number of which the sector is to be disable for ECC checking

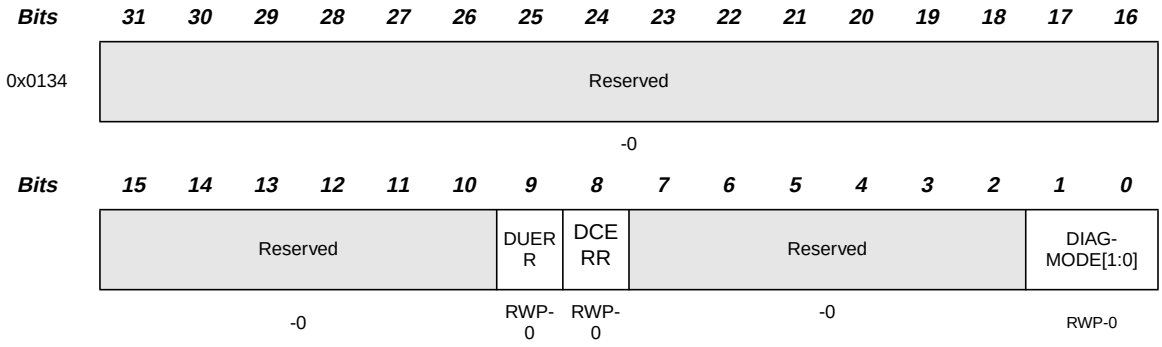
Bit 23:21 **BankID1[2:0].** Bank Number of which the sector is to be disable for ECC checking

Bit 20:16 **SectorID1[4:0].** Inverted Sector Number of which the sector is to be disable for ECC checking

Value entered in BankID1_inverse[2:0] and SectorID1_inverse[4:0] must be the complement value of BankID1[2:0] and SectorID1[4:0] respectively in order for the sector selected in SectorID1[4:0] to be excluded from ECC checking.

Bit 15:0 **See above for details.**

5.17 FMDIAGCTRL (Diagnostic Control Register)



U = Undefined; -*n* = Value after reset, R=Read, WP=Write in Privilege Mode

Bit 31:10 **Reserved.**

Bit 9 **DUERR. Diagnostic Un-correctable Error Status Flag**

When this bit is set, it indicates that the ECC logic has detected an un-correctable error during diagnostic mode.

Bit 8 **DCERR. Diagnostic Correctable Error Status Flag**

When this bit is set, it indicates that the ECC logic has detected an correctable error during diagnostic mode.

Bit 7:2 **Reserved.**

Bit 1:0 **DIAGMODE[1:0]. Diagnostic Mode**

00 = Diagnostic disabled.
01 = Diagnostic enabled. Diagnostic data correction mode.
10 = Diagnostic enabled. Diagnostic syndrome reporting mode
11 = reserved

5.18 Bank Access Control Register 1 (FMBAC1)

FMBAC1 is a half-word register. It controls bank standby mode wait state generation, bank fallback power mode, and bank active grace period (AGP) delay. Each bank in the flash module has one FMBAC1 register. The bank is selected via BANK[2:0] of the FMMAC2 register. As only one bank at a time

can be selected by FMMAC2, only the register of the selected bank appears at this address.

Bit	15	8	7	2	1	0
0x0000	BAGP[7:0]		BSTDBY[5:0]		BNKPWR[1:0]	
	RWP-00000000		RWP-111111		RWP-11	

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 15-8 **BAGP[7:0].** Bank Active Grace Period.

These bits contain the starting count value for the BAGP down counter. Any access to a given bank causes its BAGP counter to reload the BAGP value for that bank. After the last access to this flash bank, the down counter delays from 0 to 255 SYSCLK cycles before putting the bank into one of the fallback power modes as determined by BNKPWR[1:0] in this register. See Section 4.6.2, *Power Mode Control* on page 31 for bank activation logic.

Bits 7-2 **BSTBY[5:0]. Bank Standby.**

These bits contain the starting count value for the bank standby down counter. While the bank is in standby mode, the power mode management logic holds the bank standby counter at this value. When the bank exits standby power mode, the down counter delays (counts down to zero) from 0 to 63 SYSCLK cycles before putting the bank into bank active mode.

Bits 1-0 **BNKPWR[1:0].** Bank Power Mode.

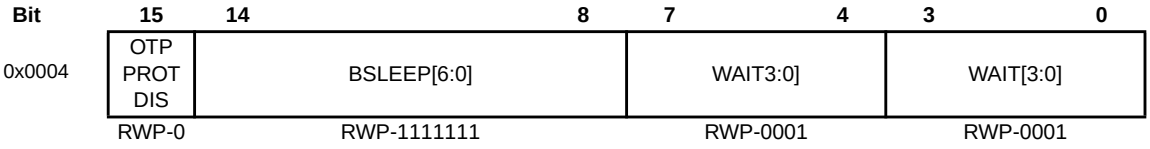
These bits describe the fall back power mode which the flash bank enters after the bank active grace period counter has timed out. The default (11 binary) is to stay active.

00 = Sleep (Sense amplifiers and sense reference disabled)
01 = Standby (Sense amplifiers disabled, but sense reference enabled)
10 = Reserved
11 = Active (Both sense amplifiers and sense reference enabled)

5.19 Bank Access Control Register 2 (FMBAC2)

FMBAC2 is a half-word register. It controls wait state generation, bank sleep delay, and OTP sector protection. There is one FMBAC2 register for each bank in the flash module. The bank is selected via BANK[2:0] of the FMMAC2

register. As only one bank at a time can be selected by FMMAC2, only the register of the selected bank appears at this address.



Bit	15	14 … 8	7 … 4	3 … 0
0x0004	OTP PROT DIS	BSLEEP[6:0]	WAIT3:0]	WAIT[3:0]
	RWP-0	RWP-1111111	RWP-0001	RWP-0001

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bit 15 **OTPPROTDIS.** OTP Sector Protection Disable.

When this bit is set, it enables programming of the OTP sector. This bit can be set only when PROTL1DIS = 1.

Bits 14-8 **BSLEEP[6:0].** Bank Sleep.

These bits contain the starting count value for the bank sleep down counter. While the bank is in sleep mode, the power mode management logic holds the bank sleep counter at this value. When the bank exits sleep power mode, the down counter delays from 0 to 127 SYSCLK cycles before putting the bank into active mode.

Bits 7-4 **WAIT[7:4].** Wait State Counter.

These bits contain the starting count value for the wait state down counter. The down counter delays from 0 to 15 SYSCLK cycles before indicating that data is available. For normal operation, these bits are set to 000 for single cycle standard read mode, or to 001 for pipeline mode. Wait bits 7:4 must match wait bits 3:0.

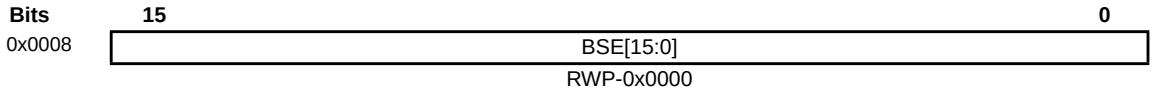
Bits 3-0 **WAIT[3:0].** Wait State Counter.

For normal operation, these bits are set to 000 for single cycle standard read mode, or to 001 for pipeline mode. Wait bits 3:0 must match wait bits 7:4.

5.20 Bank Sector Enable Registers (FMBSEA and FMBSEB)

FMBSEA and FMBSEB are half-word registers. Together they provide one enable bit per sector for up to 32 sectors per bank. Each bank in the flash module has one FMBSEA and one FMBSEB register. The bank is selected via the BANK[2:0] bits of the FMMAC2 register. As only one bank at a time

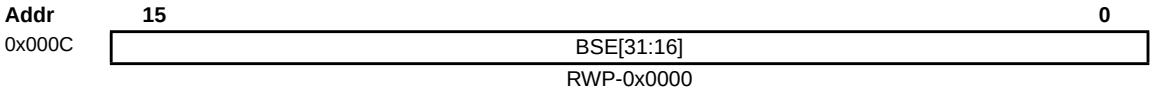
can be selected by FMMAC2, only the register for the bank selected appears at this address.



RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 15-0 **BSE[15:0].** Bank Sector Enable.

When set, a bit enables the corresponding numbered sector for program or erase access. These bits can be set only when PROTL1DIS = 1.



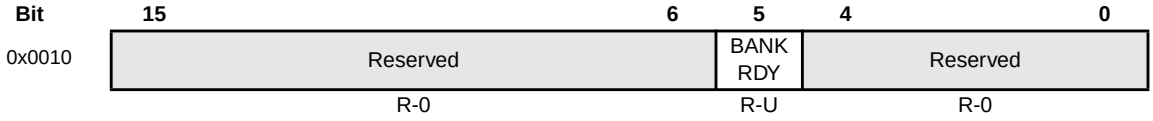
RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 15-0 **BSE[31:16].** Bank Sector Enable.

When set, a bit enables the corresponding numbered sector for program or erase access. These bits can be set only when PROTL1DIS = 1.

5.21 Bank Ready Register (FMBRDY)

FMBRDY is a half-word register. It allows the user to determine if the associated bank is ready for read access. FMBRDY is a local register; therefore, there is one for each flash bank present.



R: Read Only, -U: Undefined after reset

Bits 15-6 **Reserved**

Read values are zeros. Writes have no effect.

Bit 5 **BANKRDY.** Bank Ready.

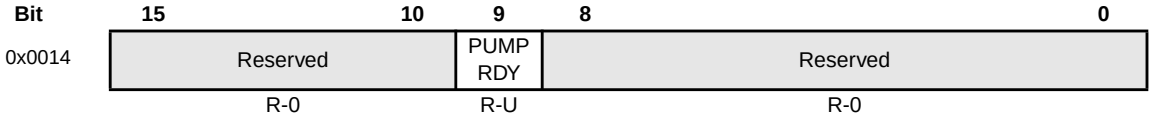
This is a read-only bit which allows software to determine if a bank is ready for access before the access is attempted.

Bits 4-0 **Reserved**

Read values are zeros. Writes have no effect.

5.22 Pump Ready Register (FMPRDY)

FMPRDY is a half-word register. It allows software to determine if the charge pump is ready for flash read access. FMPRDY is a global register; therefore, the MCU F05 Platform Flash-ECC has only one FMPRDY register.



R: Read Only, -U: Undefined after reset

Bits 15-10 **Reserved**

Read values are zeros. Writes have no effect.

Bit 9 **PUMPRDY.** Pump Ready.

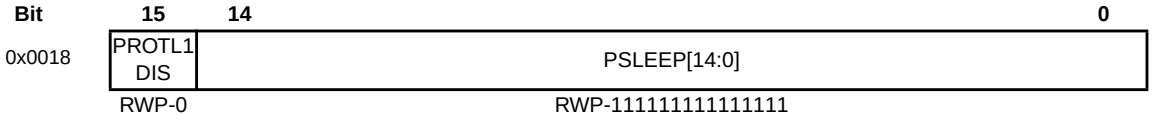
This is a read-only bit which allows software to determine if the charge pump is ready for flash access before the access is attempted.

Bits 8-0 **Reserved**

Read values are zeros. Writes have no effect.

5.23 Module Access Control register 1 (FMMAC1)

FMMAC1 is a half-word register. It supports charge pump sleep wait state generation and Level 1 protection. FMMAC1 is a global register; therefore, the flash module has only one, regardless of the number of banks present.



RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bit 15 **PROTL1DIS.** Level 1 Protection Disable.

Setting this bit enables writing to the OTPROTDIS bits as well as to the Sector Enable registers, FMBSEA and FMBSEB, for all banks. Clearing this bit disables these accesses.

Bits 14-0 **PSLEEP[14:0].** Pump Sleep.

These bits contain the starting count value for the charge pump sleep down counter. While the charge pump is in sleep mode, the power mode management logic holds the charge pump sleep counter at this value. When the charge pump exits sleep power mode, the down counter delays from 0 to 32767 SYSCLK cycles before putting the charge pump into standby power mode (the flash module *can not* exit charge pump sleep mode directly to active mode).

5.24 Module Access Control Register 2 (FMMAC2)

FMMAC2 is a half-word register. It supports control port operations, charge pump fallback power mode, and charge pump standby wait state generation. FMMAC2 is a global register; therefore, the flash module has only one, regardless of the number of banks present.

Bit	15	5	4	3	2	0
0x001C	PSTDBY[10:0]		PMPPWR[1:0]		BANK[2:0]	
	RWP-11111111111		RWP-11		RWP-000	

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 15-5 **PSTDBY[10:0].** Pump Standby.

These bits contain the starting count value for the charge pump standby down counter. While the charge pump is in standby mode, the power mode management logic holds the charge pump standby counter at this value. When the charge pump exits standby power mode, the down counter delays from 0 to 2047 SYSCLK cycles before putting the charge pump into active mode.

Bits 4-3 **PMPPWR[1:0].** Flash Pump Fallback Power Mode.

These bits select what power mode the charge pump enters after the pump active grace period (PAGP) counter has timed out.

00 = Sleep (all pump circuits disabled)
01 = Standby (only bandgap reference active)
10 = Reserved
11 = Active (all pump circuits active)

Bits 2-0 **BANK[2:0].** Bank Enable.

These bits select which bank is enabled for operations such as local register access, OTP sector access, and program/erase commands. BANK selects only one bank at a time from up to eight banks depending on the specific device being used. For example, a value of 000 binary selects Bank 0; 101 binary selects Bank 5.

5.25 Pump Active Grace Period Register (FMPAGP)

FMPAGP is a half-word register. It supports the pump active grace period delay value. FMPAGP is a global register; therefore, the flash module has only one, regardless of the number of banks present.

Bit	15	0
0x0020	PAGP[15:0]	
	RWP-0000000000000000	

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 15-0 **PAGP[15:0]. Pump Active Grace Period.**

This register contains the starting count value for the PAGP mode down counter. Any access to flash memory causes the counter to reload with the PAGP value. After the last access to flash memory, the down counter delays from 0 to 65535 SYSCLK cycles before entering one of the charge pump fallback power modes as determined by PMPPWR[1:0] in the FMMAC2 register. See Section 4.6.2, *Power Mode Control* on page 31 for charge pump activation logic.

5.26 Module Status Register (FMMSTAT)

FMMSTST is a half-word-access read only register. This register indicates whether an erase or program operation has been completed, suspended, failed, or is in progress. FMSTAT is a global register; therefore, the flash module has only one, regardless of the number of banks present.

Bit	15	10	9	8	7	6	5	4	3	2	1	0
0x0024	Reserved			BUSY	ERS	PGM	INVDAT	CSTAT	3VSTAT	ESUSP	PSUSP	SLOCK
	0		R-0	R-1->0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

RW: Read/Write in all modes, U: Undefined, *-0:* Value after reset

Bits 15-9 **Reserved**

Read values are zeros. Writes have no effect.

Bit 8 **BUSY.** Busy.

When set, this bit indicates that a program, erase, or suspend operation is being processed, or that the module is in the process of being reset. Initially after reset, BUSY is set, then it is cleared after the flash module is ready for access.

Bit 7 **ERS.** Erase Active.

When set, this bit indicates that the flash module is actively performing an erase operation. This bit is set when erasing starts and is cleared when erasing is complete. It is also cleared when the erase is suspended and set when the erase resumes.

Bit 6 **PGM.** Program Active.

When set, this bit indicates that the flash module is currently performing a program operation. This bit is set when programming starts and is cleared when programming is complete. It is also cleared when programming is suspended and set when programming is resumes.

Bit 5 **INVDAT.** Invalid Data.

When set, this bit indicates that the user attempted to program a “1” where a “0” was already present. This bit is cleared by the Clear Status command.

Bit 4 **CSTAT.** Command Status.

When set, this bit informs the host that the program, erase, or validate sector command failed. This bit is cleared by the Clear Status command.

Bit 3 **3VSTAT.** VDD3V Status.

When set, this bit indicates if the 3.3V power supply dipped below the lower limit allowable during a program or erase operation. This bit is cleared by the Clear Status command.

Bit 2 **ESUSP.** Erase Suspend.

When set, this bit indicates that the flash module has received and processed an erase suspend command. This bit remains set until the erase resume command has been issued.

Bit 1 **PSUSP.** Program Suspend.

When set, this bit indicates that the flash module has received and processed a program suspend command. This bit remains set until the program resume command has been issued.

Bit 0 **SLOCK.** Sector Lock Status.

When set, this bit indicates that the operation was halted because the target sector was locked for erasing and programming either by the sector protect bit or by write protection key logic. This bit is cleared by the Clear Status command.

6 Application Information

6.1 Powering Down Flash for Halt Mode

To completely power down all of the flash banks and the flash pumps, the code must be executing from RAM or some memory other than flash when the CPU is halted. The host must first set the fallback power bits to sleep mode. Then, before the host enables halt mode, it must execute from RAM long enough to let the active grace period for all banks and the charge pump expire.

Note: When Putting All Banks Into Sleep or Standby Mode, Ensure That There Are No Accesses to Non-Existing Memory

If all banks are in sleep or standby mode and an access to a non-existing bank is performed, the CPU will hang. This can be avoided by setting the memory decoder to enable only the amount of flash that is physically implemented on the chip.

6.2 Setting a Different Number of Wait States for Each Bank

Typically, the number of wait states is set to the same value for all banks - one when in pipeline mode and zero when not in pipeline mode. If for any reason the number of wait states will differ between banks, the bank with the higher number of wait states must be set before the bank with the lower number.

6.3 Platform Flash-F05 Flash Wrapper Registers

The table shows the offset address of each register in Platform Flash-F05 vs. R1x Flash-F05 wrapper.

Table 8. Register Offset Address

Register Name	Offset in R1x	Offset in Platform
FCI_BAC1	0x0000	0x0000
FCI_BAC2	0x0004	0x0004
FCI_BSPA	0x0008	0x0008
FCI_BSPB	0x000C	0x000C
FCI_BRDY	0x0010	0x0010
FCI_PRDY	0x0014	0x0014

Register Name	Offset in R1x	Offset in Platform
FCI_MAC1	0x3C00	0x0018
FCI_MAC2	0x3C04	0x001C
FCI_PAGP	0x3C08	0x0020
FCI_MSTAT	0x3C0C	0x0024
FCI_TC	0x3C10	0x0028
FCI_OPT	0x1C00	0x0100
FCI_REGION	0x1C04	0x0104
FCI_BUSYP	0x1C08	0x0108
FCI_PROTKEY	0x1C0C	0x010C
FCI_EDACCTRL1		0x0110
FCI_EDACCTRL2		0x0114
FCI_EDACCNT		0x0118
FCI_CERRADDR		0x011C
FCI_CERRPOSITION		0x0120
FCI_EDACSTATUS		0x0124
FCI_UERRADDR		0x0128
FCI_EMUDMSW		0x012C
FCI_EMUDLSW		0x0130
FCI_SECDIS		0x0134
FCI_DIAGCTRL		0x0138
FCI_PUTE0	0x2800	0x0300
FCI_PUTE1	0x2804	0x0304
FCI_PUTE2	0x2808	0x0308
FCI_PUTE3	0x280C	0x030C
FCI_PUTE4	0x2810	0x0310
FCI_BANKTE	0x2814	0x0314

Register Name	Offset in R1x	Offset in Platform
FCI_CNTE0	0x2818	0x0318
FCI_CNTE1	0x281C	0x031C
FCI_CNTE2	0x2820	0x0320
FCI_CNTE3	0x2824	0x0324
FCI_DALATCH	0x2828	0x0328
FCI_CSMSTATE	0x2000	0x0200
FCI_WSMSTATE	0x2004	0x0204
FCI_COM	0x2008	0x0208
FCI_HIADD	0x200C	0x020C
FCI_LOADD	0x2010	0x0210
FCI_DAREG	0x2014	0x0214
FCI_PRGERSETUP	0x2018	0x0218
FCI_MASKDA	0x201C	0x021C
FCI_COMSETUP	0x2020	0x0220
FCI_PEVSETUP	0x2024	0x0224
FCI_RMODECNT	0x2028	0x0228
FCI_CVSETUP	0x202C	0x022C
FCI_SETUPCNT	0x2030	0x0230
FCI_PRGCMPHD	0x2034	0x0234
FCI_ERHD	0x2038	0x0238
FCI_SPRGPL	0x203C	0x023C
FCI_PGMERVHD	0x2040	0x0240
FCI_CPVHD	0x2048	0x0248
FCI_HDCNT	0x204C	0x024C
FCI_PGPLWI	0x2050	0x0250
FCI_ERPLWI	0x2054	0x0254

Register Name	Offset in R1x	Offset in Platform
FCI_CMPLWI	0x2058	0x0258
FCI_PGERVACC	0x205C	0x025C
FCI_CMPVACC	0x2060	0x0260
FCI_SERPL	0x2064	0x0264
FCI_PLWICNT	0x2068	0x0268
FCI_WSMMODE	0x206C	0x026C
FCI_PGLO	0x2070	0x0270
FCI_ERLO	0x2074	0x0274
FCI_HIADDSA	0x2078	0x0278
FCI_MAXPRPL	0x207C	0x027C
FCI_MAXERPL	0x2080	0x0280
FCI_MAXCMPPL	0x2084	0x0284
FCI_PLCNT	0x2088	0x0288
FCI_SEFILO	0x208C	0x028C
FCI_SELALO	0x2090	0x0290
FCI_SEHI	0x2094	0x0294
FCI_ADDCNTLO	0x2098	0x0298
FCI_ADDCNTHI	0x209C	0x029C
FCI_PGSUDA	0x20A0	0x02A0
FCI_STMATE	0x20A4	0x02A4

7 Manufacturing Test

F05 Flash-ECC module consists of both the flash wrapper and flash memory. Flash wrapper is a soft module constructed with TI ASIC gates while the flash memory is constructed with array of memory cell bits with additional supporting custom circuit. They need to be tested with different test methodology.

7.1 Flash Wrapper Testing

Flash wrapper should be tested with the traditional ATPG stuck-at fault test methodology as the primary metric to measure test coverage. In addition, at-speed test using ATPG based at-speed methodology along with functional tests generated by TDL are required to fully uncover any manufacturing defects. The at-speed coverage needs to be independent of stuck-at ATPG test coverage.

7.1.1 Stuck-At ATPG Testing

Flash wrapper needs to be designed with the goal of achieving 99% stuck-at ATPG test coverage. The designer needs to generate, simulate and fault-grade all stuck-at ATPG patterns for the wrapper logic and show that the module achieves 99% test coverage. Standard Platform DFT guidelines must be strictly followed. Below are notable guidelines to ensure compliance.

1) All flip-flops must be scannable
2) All latches made transparent during test
3) All I/O's must either be registered or bounded with scan cells.
4) No asynchronous path except resets
5) Asynchronous resets overridden during test
6) No tri-states busses

7.1.2 IDDQ Testing

Platform DFT architecture provides capability to measure IDDQ current using device level multiple global scan chains. During IDDQ testing the wrapper must be able to put both the flash banks and pump module in sleep mode to obtain minimum current.

7.1.3 At-Speed Testing

At speed testing is an important manufacturing test for flash wrapper since the module contains many critical paths which determines the final system

performance. At speed test utilizes both the ATPG based at-speed methodology and functional tests generated from TDL.

ATPG tools supports two delay fault models: transition and path delay. Both of these fault models catch slow-to-rise and slow-to-fall faults.

7.1.4 Transition Delay Testing

Transition delay testing usually requires about 3-4 times the stuck-at ATPG patterns for the same design. At the module level the designer will generate the transitional delay coverage.

7.1.5 Path Delay Testing

This methodology is similar to transition delay testing. The only difference is that the critical paths have to be selected by the designer using STA. This requires the critical timing path from STP tool to be fed to the ATPG tools. Flash module critical path analysis is part of the device STA. Device path delay ATPG will include flash wrapper path delay.

7.1.6 Functional Testing

Note that ATPG based at-speed testing can not distinguish between true and false critical paths. Functional patterns specifically engineered by the designer not only catch critical timing paths but also exercises flash wrapper functionality which can not be tested using the ATPG based methodology. Not all but selected functional test patterns must be used to test the following critical flash module features. The following should not be taken as an exhaustive module verification plan but only serve as a guide on what functional testcases should be used for manufacturing tests.

1) Running program code from flash memory at pipeline mode with different wait states.
2) Running program code from flash memory at non-pipeline mode
3) Data read from each flash bank at pipeline mode with different wait states
4) Data read from each flash bank at non-pipeline mode
5) Multiple master reads at the same time from flash wrapper to exercise the built-in arbiter.
6) The above tests running with both ECC enabled and disabled.
7) All write command operations (program, erase, validate, etc).
8) power-up and down sequence
9) Sleep and standby power fall-back power mode
10) Flash register read/write

7.1.7 ECC Testing

ECC logic is a major part of the flash wrapper. It is both timing critical and requires special testing strategy. To exercise both the critical paths and functionality some faults must be intentionally injected into the flash memory during flash programming. This is doable but it does not fully test the entire ECC logic. The flash wrapper provides an ECC diagnostic mode which allows CPU to apply diagnostic data into the ECC logic directly without fetching data from the flash bank. This diagnostic mode provides an easy mechanism to fully every aspect of the ECC logic. Following are required as part of the manufacturing tests.

1) Use diagnostic data correction mode to test single-bit and multi-bit errors.
2) Use diagnostic syndrome mode to test single-bit and multi-bit errors.
3) Injects faults during flash programming so it causes ECC to violate either single-bit or multi-bit error during pipeline read.
4) Injects faults during flash programming so it causes ECC to violate either single-bit or multi-bit error during non-pipeline read.
5) Interrupts due to single-bit zero fail or one fail or profiling error.
6) Error signal due to multi-bit error.
7) All of the above for each bank

7.2 Flash Bank/Pump Test

Flash module contains a PMT interface which allows tester direct testing of both the flash banks and pump module. The PMT interface and various PMT modes supported are identical between Platform F05 Flash and R1x F05 Flash module. Refer to R1x F05 Flash PMT document for details. Platform F05 PMT offers the following capabilities.

- ❏ Read from flash banks (including OTP sectors)
- ❏ Read wrapper registers
- ❏ Write wrapper registers
- ❏ Issue commands to state machine
- ❏ Program flash memory (including OTP sectors)
- ❏ Erase sectors
- ❏ Compact sectors
- ❏ Perform verify reads of flash (Including OTP sectors)
- ❏ Enable 64 bank/pump test modes (TCR register)

In one of the following test modes:

- PMT
  - This mode provides memory access from either pads at multiprobe or pins on a package for testing. It is possible to perform read, program, erase, compaction, and all of the verification read modes via the test port in the PMT test mode.
- PMT Direct Address
  - This mode is an extended PMT mode which allows faster testing of the flash memory for read, verify read, program, and compaction operation. It requires the special address bus.
- PMT with Address Auto Increment
  - This is also an extended PMT mode which allows faster test of the memory for read operations. Flash wrapper test logic automatically increments the address on every rising edge of TCLK.
- Register
  - Register mode allows the tester to write or read any memory mapped registers in the flash wrapper.
- Special Register Read
  - This special Register mode allows the tester to read a register real time. This feature is used to monitor registers used by FSM while it is running.
- State Machine
  - This mode allows tester to perform any flash operation which were available to user in normal functional mode in test mode as well.