eSi-TSMC Flash

1 Contents

2 Overview

The eSi-TSMC Flash core provides an interface a TSMC embedded flash. It supports the following features:

- Supports access to one data memory and one information memory.
- Optionally supports redundancy.
- Optional ECC support with SECDEC (single-bit error correction, double-bit error detection).
- Programmable read/write timings, to support different clock frequencies.
- Write protection.
- Discharge on brown out.
- Triple AMBA 3 AHB-lite slave interfaces:
    - `cfg` for configuration and programming.
    - `mem` for memory read accesses.
    - `info` for information page read accesses.

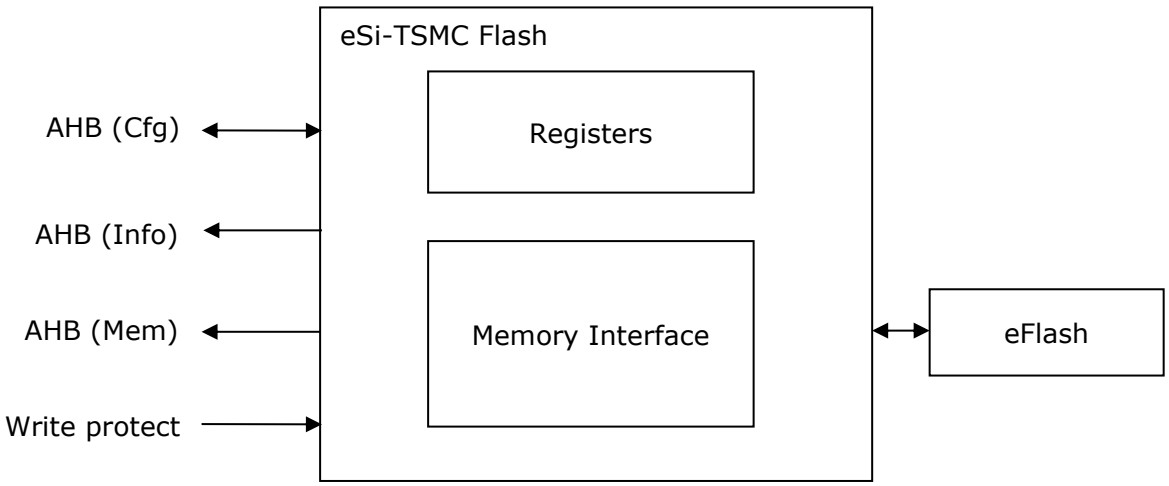

Figure 1: eSi-TSMC Flash

3 Hardware Interface

Module Name	cpu_ahb_tsmc_flash
HDL	Verilog
Technology	Generic
Source Files	cpu_ahb_tsmc_flash.v

Port	Type	Description
xadr_pins	Integer	Width of X address bus to flash
yadr_pins	Integer	Width of Y address bus to flash
data_pins	Integer	Number of pins in data bus used for data
address_pins	Integer	Width of internal address bus wide enough to access all addresses in flash data block: (xadr_pins+yadr_pins+$\log_2$(data_pins/8))
ecc_pins	Integer	Number of bit in data bus used for ECC
recall_on_first_read	Integer	Determines whether a recall operation will automatically be performed on the first memory read
bias_initial	Integer	Determines whether the "bias initial" test mode sequence is performed before erase reference cell
info_pages	Integer	How many information pages are implemented
read_wait_states	5:0	Reset value for timing0.R
five_us_reset	15:0	Reset value for timing0.F
erase_cycles_reset	23:0	Reset value for timing1.E
program_cycles_reset	15:0	Reset value for timing2.P
hundred_ns_reset	7:0	Reset value for timing2.H
ten_ns_reset	3:0	Reset value for timing2.T
resume_after_brownout	Boolean	Whether flash operation can continue after brown out
write_protect_privileged	Boolean	Whether control.WP must be clear to allow privileged writes

Table 1: Parameters

Port	Direction	Width	Description
write_protect_all	Input	1	Prevents all write and erase operations
write_protect	Input	32	Data bock write protection flags
info_write_protect	Input	info_pages	Information block write protection flags
disable_redundancy	Input	1	Disables use of redundancy information
brown_out	Input	1	Power brown out indicated. When asserted, the controller will perform the discharge sequence if occurs during erase or programming
discharged	Output	1	Indicates discharge sequence is complete after brown out
mem_hclk	Input	1	Memory interface, AHB clock
mem_hresetn	Input	1	Memory interface, AHB reset, active-low
mem_haddr	Input	BITS	Memory interface, AHB address
mem_hburst	Input	3	Memory interface, AHB burst type
mem_hmastlock	Input	1	Memory interface, AHB locked transfer
mem_hprot	Input	4	Memory interface, AHB protection
mem_hsize	Input	3	Memory interface, AHB size
mem_htrans	Input	2	Memory interface, AHB transfer type
mem_hwdata	Input	BITS	Memory interface, AHB write data
mem_hwrite	Input	1	Memory interface, AHB write
mem_hready	Input	1	Memory interface, AHB ready

mem_hsel	Input	1	Memory interface, AHB select
mem_hready	Output	1	Memory interface, AHB ready
mem_hrdata	Output	BITS	Memory interface, AHB read data
mem_hresp	Output	1	Memory interface, AHB response
info_hclk	Input	1	Info page interface, AHB clock
info_hresetn	Input	1	Info page interface, AHB reset, active-low
info_haddr	Input	BITS	Info page interface, AHB address
info_hburst	Input	3	Info page interface, AHB burst type
info_hmastlock	Input	1	Info page interface, AHB locked transfer
info_hprot	Input	4	Info page interface, AHB protection
info_hsize	Input	3	Info page interface, AHB size
info_htrans	Input	2	Info page interface, AHB transfer type
info_hwdata	Input	BITS	Info page interface, AHB write data
info_hwrite	Input	1	Info page interface, AHB write
info_hready	Input	1	Info page interface, AHB ready
info_hsel	Input	1	Info page interface, AHB select
info_hready	Output	1	Info page interface, AHB ready
info_hrdata	Output	BITS	Info page interface, AHB read data
info_hresp	Output	1	Info page interface, AHB response
cfg_hclk	Input	1	Configuration interface, AHB clock
cfg_hresetn	Input	1	Configuration interface, AHB reset, active-low
cfg_haddr	Input	BITS	Configuration interface, AHB address
cfg_hburst	Input	3	Configuration interface, AHB burst type
cfg_hmastlock	Input	1	Configuration interface, AHB locked transfer
cfg_hprot	Input	4	Configuration interface, AHB protection
cfg_hsize	Input	3	Configuration interface, AHB size
cfg_htrans	Input	2	Configuration interface, AHB transfer type
cfg_hwdata	Input	BITS	Configuration interface, AHB write data
cfg_hwrite	Input	1	Configuration interface, AHB write
cfg_hready	Input	1	Configuration interface, AHB ready
cfg_hsel	Input	1	Configuration interface, AHB select
cfg_hready	Output	1	Configuration interface, AHB ready
cfg_hrdata	Output	BITS	Configuration interface, AHB read data
cfg_hresp	Output	1	Configuration interface, AHB response
data_in	Input	ecc_pins +data_pins	Data input
data_out	Output	ecc_pins +data_pins	Data output
data_out_enable	Output	1	Data output enable
output_enable_n	Output	1	Flash data output enable, active-low
xadr	Output	xadr_pins	Row address
yadr	Output	yadr_pins	Column address
xe	Output	1	X address enable
ye	Output	1	Y address enable
se	Output	1	Sense amplifier enable
ifren	Output	1	Information block enable
erase	Output	1	Erase cycle
mas1	Output	1	Mass erase cycle
prog	Output	1	Program cycle
nvstr	Output	1	Non-volatile store cycle
recall	Output	1	Recall trimming code
slm	Output	1	Sleep mode enable
tmr	Output	1	Test mode reset
chip_select_n	Output	1	Chip-select, active-low

ifren1	Output	1	Information block 1 enable
reden	Output	2	Redundancy page select
interrupt_n	Output	1	Interrupt, active-low

Table 2: I/O Ports

For complete details of the AHB signals, please refer to the AMBA 3 AHB-Lite Protocol v1.0 Specification available at:

http://www.arm.com/products/system-ip/amba/amba-open-specifications.php

For further details of the flash interface signals, refer to the TSMC flash macro datasheet.

4 Software Interface

4.1 Register Map

Register	Address offset	Access	Description
control	0x00	R/W	Control register
timing0	0x04	R/W	Timing register 0
timing1	0x08	R/W	Timing register 1
timing2	0x0c	R/W	Timing register 2
unlock1	0x18	W	Unlock 1
unlock2	0x1c	W	Unlock 2
address	0x20	W	Erase / Program address
pb_data	0x24	W	Program buffer data
pb_index	0x28	W	Program buffer index
status	0x2c	R/W	Status register
redun_0	0x30	R/W	Redundant address 0
redun_1	0x34	R/W	Redundant address 1
sec_count	0x38	R/W	Single error corrected count
ded_count	0x3c	R/W	Double error detected count

Table 3: Register Map

4.1.1 Control Register

The control register contains flags that control the operation of the flash memory. After reset, register write protection is enabled.

	12	11	10	9	8	7	6	5	4	3	2	1	0
	EDIE	ECIE	WCIE	EI	AP	R	ERC	P	EP	E	-	WP	SLM

Figure 2: Format of the `control` register

Register	Values	Description
SLM	0 – Active mode 1 – Sleep mode	Sleep mode. Directly drives the SLM signal to the flash
WP	0 – Write protect disabled 1 – Write protect enabled	Register write protect
E	0 – No op 1 – Erase	Erase. Write-only
EP	0 – No op 1 – Erase page	Erase page. Write-only
P	0 – No op 1 – Program	Program flash. Write-only
ERC	0 – No op 1 – Erase ref. cell	Erase reference cell. Write-only
R	0 – No op 1 – Recall	Recall trim code. Write-only
AP	0 – Disable auto-program 1 – Enable auto-program	Enables automatic programming of the flash when the program buffer becomes full
EI	0 – Error not indicated 1 – Error indicated	Indicates an error on `mem_hresp` if an uncorrectable error is detected by ECC logic
WCIE	0 – Interrupt disabled 1 – Interrupt enabled	Write complete interrupt enable
ECIE	0 – Interrupt disabled 1 – Interrupt enabled	Error corrected interrupt enable

EDIE	0 – Interrupt disabled 1 – Interrupt enabled	Error detected interrupt enable

Table 4: Fields of the `control` register

4.1.2 Timing Register 0

The timing control register 0 contains timing information for read and other operations.

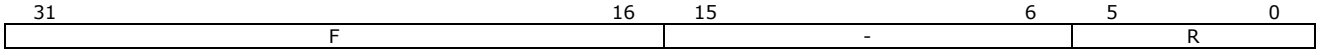

Figure 3: Format of the `timing0` register

Register	Values	Description
R	0 – 63	Read wait states. This should be a minimum of 2 and meet Tacc
F	0 – 65,536	Number of clock cycles to meet following minimum timings: Tnvs: 90nm 5us, 55nm 5us, 40nm 8us Tnvh: 90nm 5us, 55nm 5us, 40nm 5us Trcv (x2): 90nm 10us, 55nm 10us, 40nm 10us Tpgs (x2): 90nm 10us, 55nm 10us, 40nm 2us Tnvh1 (x32): 90nm 100us, 55nm 100us, 40nm 100us So: - For 90nm, set to number of clock cycles in 5us - For 55nm, set to number of clock cycles in 5us - For 40nm, set to number of clock cycles in 8us

Table 5: Fields of the `timing0` register

4.1.3 Timing Register 1

The timing register 1 contains timing information for erase operations.

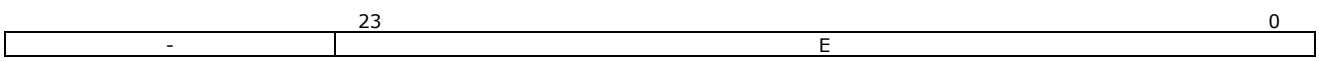

Figure 4: Format of the `timing1` register

Register	Values	Description
E	0 – 16,777,216	Erase: Number of clock cycles to meet Tme / Terase time: 90nm: 20ms / 40ms 55nm: 80ms / 160ms 40nm: 10ms / 20ms Erase reference cell: Number of clock cycles to meet Tre time:

		90nm: 40ms 55nm: 100ms 40nm: 10-20ms

Table 6: Fields of the `timing1` register

4.1.4 Timing Register 2

The timing register 2 contains timing information for programming and other operations.

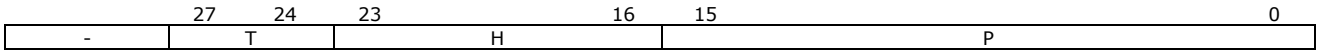

Figure 5: Format of the `timing2` register

Register	Values	Description
P	0 – 65,535	Program: Number of clock cycles to meet Tprog time: 90nm: 20us to 40us 55nm: 8us to 16us 40nm: 8us to 16us Erase reference cell: Number of clock cycles to meet Ttmh minimum time: 90nm: 100ns 55nm: 100ns 40nm: 30us
H	0 – 255	Number of clock cycles in 100ns
T	0 – 15	Number of clock cycles in 10ns

Table 7: Fields of the `timing2` register

4.1.5 Unlock 1

The unlock 1 register is a register that must be written in a specific sequence in order to unlock access to the write protection register.

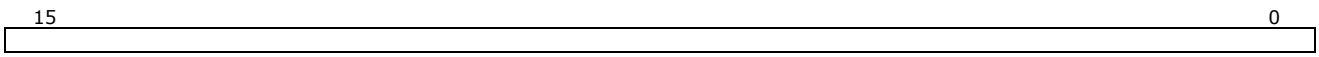

Figure 6: Format of the `unlock1` register

4.1.6 Unlock 2

The unlock 2 register is a register that must be written in a specific sequence in order to unlock access to the write protection register.

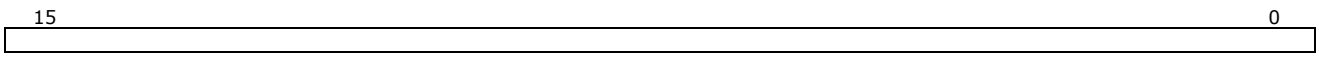

Figure 7: Format of the `unlock2` register

4.1.7 Address Register

The address register specifies the address for program and erase page operations. The `address` register will be automatically incremented by the length specified by `pb_index` after a program operation.

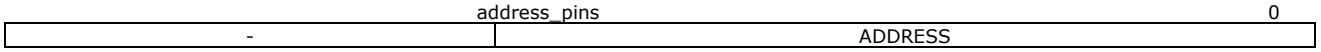

Figure 8: Format of the `address` register

4.1.8 Program Buffer Data Register

The program buffer data register provides access to a 32-byte program buffer. When this register is written, the bytes specified by the `pb_index` register in the program buffer will be set to the written data. The `pb_index` register will be automatically incremented after an access to the `pb_data` register. The number of bytes written to the program buffer will depend on the size of the AHB write. After a program operation, the contents of the program buffer are undefined.

Figure 9: Format of the `pb_data` register

4.1.9 Program Buffer Index Register

The program buffer index register specifies which bytes within a 32-byte program buffer will be written via the program buffer data register. It is automatically incremented after an access to the program buffer data register. It is also used to specify the number of bytes to program during a program operation. A value of 0 in this register results in 32 bytes being programmed.

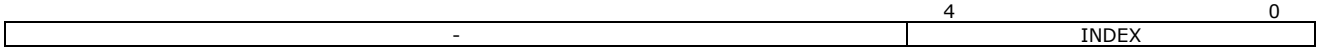

Figure 10: Format of the `pb_index` register

4.1.10 Status Register

The status register contains flags that indicate the status of erase and programming operations. To clear a bit in the status register, write a 1 to it. Writing a zero leaves a bit alone.

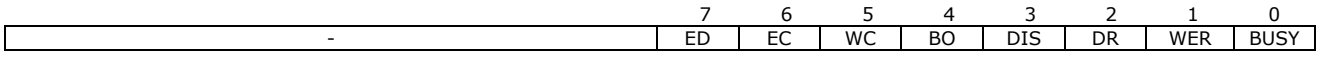

Figure 11: Format of the `status` register

Register	Values	Description
BUSY	0 – Not busy 1 – Busy	Indicates whether the flash controller is busy performing an erase or program operation
WER	0 – No error 1 – Write Protect Error	Indicates an attempt was made to erase / write a write protected page or a brown out occurred during write. Sticky
DR	0 – Redundancy enabled 1 – Redundancy disabled	Indicates the state of the `disable_redundancy` input. Read only
DIS	0 – Normal	Indicates whether the discharge sequence has

	1 – Discharged	been performed in response to `brown_out` having been asserted. Sticky
`BO`	0 – Power normal 1 – Brown out detected	State of `brown_out` input. Read only
`WC`	0 – Write not complete 1 – Write complete	Indicates if a write operation (program / erase) has completed. Sticky
`EC`	0 – No error corrected 1 – Error corrected	Indicates if an error has been corrected by the ECC logic. Sticky
`ED`	0 – No error detected 1 – Error detected	Indicates if an uncorrectable error has been detected by the ECC logic. Sticky

Table 8: Fields of the `status` register

4.1.11 Redundant Address Register

The redundant address registers contains information indicating whether the redundant pages are being used. These registers are initialised automatically after reset based on information read from info 1 block. These registers are only for diagnostic purposes.

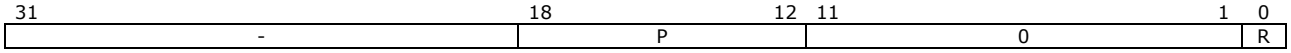

Figure 12: Format of the `redun_N` registers

Register	Values	Description
`R`	0 – Not used 1 – Used	Indicates whether the redundant page is being used
`P`		Redundant page address

Table 9: Fields of the `redun_N` register

4.1.12 Single Error Corrected Count Register

The single error corrected count register is incremented each time the ECC logic corrects a single error. This register is only implemented if `ecc_pins` > 0.

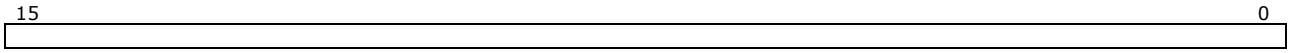

Figure 13: Format of the `sec_count` registers

4.1.13 Double Error Detected Count Register

The double error detected count register is incremented each time the ECC logic detects an uncorrectable double bit error. This register is only implemented if `ecc_pins` > 0.

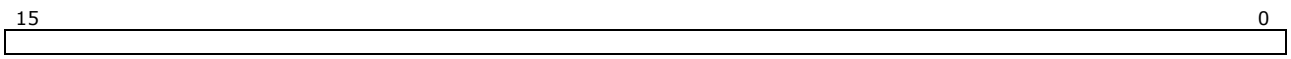

Figure 14: Format of the `ded_count` registers

4.2 Register Write Protection

When the control.WP flag is set, register writes to registers other than the unlock registers are ignored, unless the privileged protection flag is set on the AHB transfer (HPROT[1] equals 1) and write_protect_privileged is FALSE. When the control.WP flag is clear, register writes will take place regardless of the value of HPROT[1].

In order to write to the control.WP flag, access to this register must be unlocked. To do this:

- Write 0x7123 to unlock1.
- Write 0x812a to unlock2.
- Write 0xbee1 to unlock1.

Writing any other register part way through the sequence will result in the unlocked state not being reached. When in the unlocked state, writing any register will return to the locked state.

4.3 Flash Operations

4.3.1 Flash Memory Organisation

A single flash memory can be controlled by the core. For programming and erase, the flash memory is divided in to two blocks: $2^{address\_pins}$ bytes of data memory and info_pages*page_size bytes of information memory. Blocks are divided into pages of 4096 or 8192 bytes. The blocks are arranged as follows:

Low Address	High Address	Description
0x0000_0000	$2^{address\_pins}-1$	Data memory
$2^{address\_pins}$	$2^{address\_pins}$+info_pages*page_size-1	Information memory

Table 10: Flash Memory Organisation

If the flash supports the information 1 page (used for redundancy), this is not erasable or programmable.

4.3.2 Erase

To erase all pages in the data memory:

- Set the address register to 0x0.
- Set control.E to 1.

To erase all pages in the information memory and data memory:

- Set the address register to $2^{address\_pins}$.
- Set control.E to 1.

During the erase operation, status.BUSY will be set to 1 and AHB read accesses via the Mem interface will fail (HRESP will be 1). When the erase operation is complete, status.BUSY will be set to 0.

If any page to be erased is write protected, no erase operation will be performed and status.WER will be set 1.

4.3.3 Erase Page

To erase a single page in either the data memory or information memory:

- Set the address register to the address of the page.
- Set control.EP to 1.

During the erase operation, status.BUSY will be set to 1 and AHB read accesses via the Mem interface will fail (HRESP will be 1). When the erase operation is complete, status.BUSY will be set to 0.

If the page specified by the address is write protected, the erase will not be performed and status.WER will be set 1.

4.3.4 Program

A 32-byte program buffer is implemented, allowing from 1 to 32 bytes to be programmed at a time. The program buffer is accessed via the pb_data and pb_index registers.

To program N bytes to the flash at address A:

- Set pb_index to 0.
- Write the N bytes of data to pd_data.
- Set the address register to the address to start programming at, A.
- Set control.P to 1.

During the program operation, status.BUSY will be set to 1 and AHB read accesses via the Mem interface will be stalled (HRREADYOUT will be 0). When the program operation is complete, status.BUSY will be set to 0.

Programming should not cross a 32-byte address boundary. That is, the following must be true: A[4:0] + N <= 32.

When ECC is implemented, the minimum number of bytes that can be programmed is determined by the flash word width (data_pins/8)

If the page specified by the address is write protected, the program operation will not be performed and status.WER will be set to 1.

4.3.5 Erase Reference Cell

The TSMC embedded flash datasheet indicates that the erase reference cell operation should be performed once for each sample. The erase reference cell operation can be performed by setting control.ERC to 1. During the erase reference cell operation, status.BUSY will be set to 1. When complete, status.BUSY will be set to 0. The timing1.E register should specify the number of clock cycles to meet Tre and timing2.P should specify the number of clock cycles to meet Ttmh.

4.3.6 Recall

The recall operation recalls the trim code. The recall operation is performed automatically on the first read from the flash, if the operation has not been started manually and the recall_on_first_read parameter is 1. It can be started manually by setting control.R to 1. Newer flash macros do not require the recall operation to be performed.

5 Revision History

Hardware Revision	Software Release	Description
1	2.5.3	Initial release
2	2.5.4	Support two eFlash instances Add read and write protection
3	2.6.0	Revert to single eFlash instance support Remove read protection
4	2.6.4	Replace CFI interface with register based interface
5	3.3.3	Added `write_protect` inputs Added `status.WER` Made `status` register writable
6	3.3.7	Add `recall_on_first_read` parameter Add `info` interface Add `redun_0` and `redun_1` registers
7	3.3.15	Added `disable_redundancy` input Added `control.AP` `pb_data` register now supports word writes as well as byte writes `address` register is auto-incremented on program
8	3.3.18	Added `info_pages` parameter Added `info_write_protect` inputs Added `brown_out` input Added `status.DIS`
9	3.3.22	Added `read_wait_states` parameter
10	4.0.8	Added `resume_after_brown_out` parameter Add `status.BO`
11	4.1.9	Add ECC support Added `ecc_pins` parameter Added `control.WCIE` Added `control.ECIE` Added `control.EDIE` Added `status.WC` Added `status.EC` Added `status.ED` Added `interrupt_n` output Added `sec_count` register Added `ded_count` register
12	4.1.13	Added `write_protect_privileged` parameter
13	4.1.14	Added `write_protect_all` input

Table 11: Revision History

