USER MANUAL

SEP 2022

NANDFCTRL2 IP Core User's Manual

Sep 2022, Version 2022.2

Cobham Gaisler AB
Kungsgatan | SE-411 19 | Goteborg | Sweden
+46 31 7758650 | www.caes.com/gaisler

Table of contents

1 NANDFCTRL2 - NAND Flash Memory Controller with DMA 3
1.1 Overview 3
1.2 Interface ONFI 6
1.3 Memory accesses 8
1.4 Operation - Typical use case 10
1.5 Registers 11
1.6 Vendor and device identifiers 30
1.7 Implementation 30
1.8 Configuration options 30
1.9 Signal descriptions 31
1.10 Signal definitions and reset values 32
1.11 Library dependencies 32
1.12 Instantiation 33

1 NANDFCTRL2 - NAND Flash Memory Controller with DMA

1.1 Overview

Note: This IP core manual describes a NAND Flash controller that is in development.
Some features are described as not implemented and will be implemented over time together with additional extensions. See individual sections.

The NANDFCTRL2 core is a memory controller for NAND Flash memory devices. The memory controller is an Open NAND Flash Interface (ONFI) 4.0 compatible core and it can communicate with multiple parallel Flash memory devices. The core provides an AMBA AHB master DMA to perform write and read data accesses. 32 individually addressable targets via CE_n signals and 8 R/B# signals can be handled. For write accesses it is possible to issue commands to several targets at once (e.g. configuration and reset) by selecting multiple CE_n signals. The core implements a BCH EDAC which can be bypassed to allow ECC in SW. The controller also has a data randomizer, which can be bypassed. ECC functionality and randomization functionality requires a fixed layout of page area into data area, ECC area and flag area.The NANDFCTRL2 supports some Flash memory configuration in page size when it handles ECC and randomization.

Configuration of NANDFCRTL2 is done through a set of AMBA APB registers. The APB interface is also used for configuring Flash memory via ONFI commands.

A SEFI block monitors R/B# signals for time-outs and also have one external output signal (on/off) for any user functionality.

For details about the actual Flash memory interface, Flash memory architecture and ONFI 4.0 command set please refer to the *Open NAND Flash Interface specification, revision 4.0*, hereafter called the ONFI 4.0 specification.

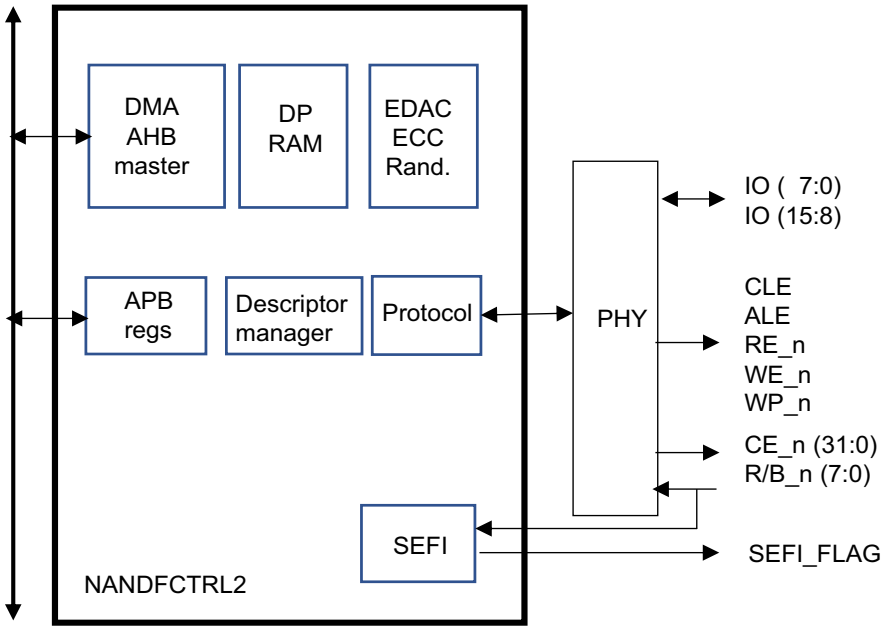

Figure 1. Block diagram

1.1.1 Functionality and limitations of current revision, and planned features

The core is currently available as a first version that is targeted for FPGA implementation and have functionalities and limitations in its features according to below:

- ONFI commands
  All mandatory commands are supported and some optional commands are implemented.
  The final version of the core is expected to support all ONFI optional commands,
  as well as vendor-specific commands.
- Asynchronous / SDR data interface, 8-bit data bus, timing mode 0-5 is supported, with EDO.
- ECC/BCH supports 2 different en/decoding schemes (configured at design time via generic)
  60 bit correctability and,
  16 bit correctability (depending on chunk size).
- Randomization on byte position (allows for free unaligned accesses).
- 3 different memory configurations of NAND flash are supported:
  16 Kb data/1024 byte chunk/60 bits ECC,
  8 Kb data/512 byte chunk/16 bits ECC,
  4 Kb data/512 byte chunk/16 bit ECC
- SEFI (Single Event Functional Interrupt) is supported by 2 means:
  - An SEFI_flag output signal can be used to toggle power-on/power-off
  (or any other user defined functionality) via a register bit.
  - All R/B# signals have configurable time-out monitoring.
  Each R/B# signal is mapped to one or several CE signal/s and when one of it's CE is active for an access on PHY a timer is started if R/B# signal shall toggle for that access.
  The timer takes its values from configurable time-out registers depending on the access type on PHY.

- Die size of 16 bits are not supported i.e. data lane IO(15-8) is currently not supported.
- Synchronous data interfaces are not supported (NV-DDR, NV-DDR2, NV-DDR3).
  NV-DDRs are planned.
- Fault Tolerance option is not supported for this revision. It is planned to be added.
- MBIST support is not implemented on this revision of the controller. It is planned to be added.

1.1.2 Supported ONFI commands

Mandatory commands are supported and some optional commands are also supported.

Table 1. Implemented ONFI commands for NANDFCTRL2.

Command	Optional/ Mandatory	1st Cycle	2nd Cycle	Supported
Read	M	00h	30h	YES
Read Multi-plane	O	00h	32h	NO
Copyback Read	O	00h	35h	NO
Change Read Column	M	05h	E0h	YES
Change Read Column Enhanced	O	06h	E0h	NO
Read Cache Random	O	00h	31h	NO
Read Cache Sequential	O	31h		NO
Read Cache End	O	3Fh		NO
Block Erase	M	60h	D0h	YES
Block Erase Multi-plane	O	60h	D1h	NO
Read Status	M	70h		YES
Read Status Enhanced	O	78h		NO
Page Program	M	80h	10h	YES
Page Program Multi-plane	O	80h	11h	NO
Page Cache Program	O	80h	15h	NO
Copyback Program	O	85h	10h	NO
Copyback Program Multi-plane	O	85h	11h	NO
Small Data Move	O	85h	11h	NO
Change Write Column	M	85h		YES
Change Row Address	O	85h		YES
Read ID	M	90h		YES
Volume Select	O	E1h		NO
ODT Configure	O	E2h		NO
Read Parameter Page	M	ECh		YES
Read Unique ID	O	EDh		YES
Get Features	O	EEh		YES
Set Features	O	EFh		YES
LUN Get Features	O	D4h		YES
LUN Set Features	O	D5h		YES
ZQ Calibration Short	O	D9h		NO
ZQ Calibration Long	O	F9h		NO
Reset LUN	O	FAh		YES
Synchronous Reset	O	FCh		NO
Reset	M	FFh		YES

1.2 Interface ONFI

1.2.1 SDR interface

ONFI specification allows for two independent 8-bit data buses in some ONFI packages. “Implementations may tie the data lines and control signals (RE_n, CLE, ALE, WE_n, WP_n, and DQS) together for the two independent 8-bit data buses externally to the device.“

NANDFCTRL2 does not handle multiple data interfaces explicitly, such feature is expected to be handled either at system level with use of data muxes or at PCB.

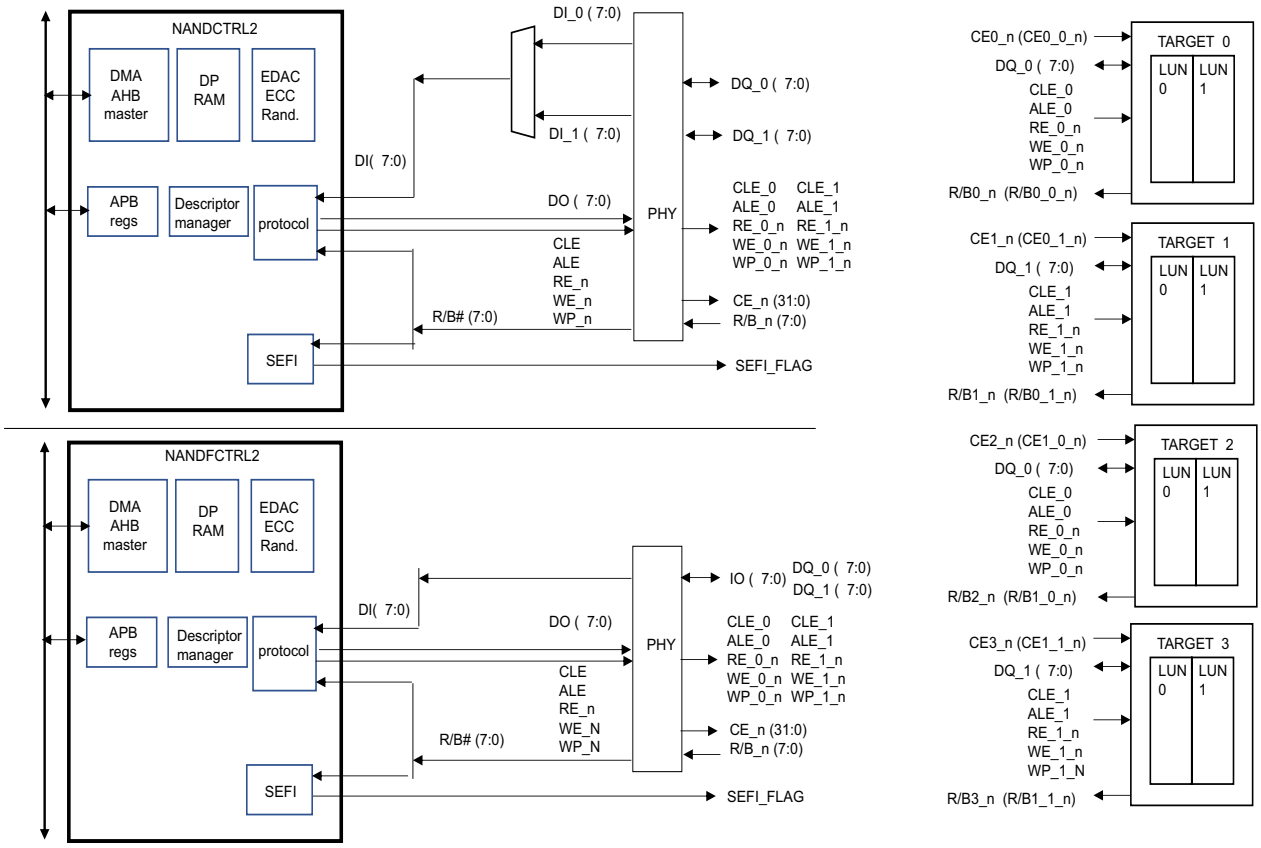

Figure 2. Interfaces

1.2.2 Data interface type

Current implementation supports interface type Single Data Rate (SDR).

1.2.3 SDR timing parameters

The ONFI timing parameters that the core explicitly handles are:

- tCS CE setup time
- tWW WP_n transition to WE low

- tRHW RE_n high to we WE low
- tRR Ready to RE_n low (data only)
- tWB WE_n high to SR[6] low

- tADL ALE to data loading time
- tWHR WE_n high to RE low
- tCCS Change Column setup time

- tWH WE_n high hold time
- tWP WE_n pulse width
- tREH RE_n high hold time
- tRP RE_n pulse width

All other timing requirements are either fulfilled through design, or are handled by the Flash memory devices. See the ONFI specification for details about the different timing parameters.

The data interface timing are fully programmable via AMBA APB registers.

1.2.4 SDR timing modes

NANDFCTRL2 supports SDR timing modes 0-5 via programmable timing parameter registers and Extended Data Output (EDO) is also supported.

Care must be taken to assure that the timing parameter registers are setup according to system clock as they are counters running on system clock cycle period.

1.3 Memory accesses

1.3.1 Flash memory page layout

Support for ECC generation and data randomization in hardware are two options that can be enabled separately. When ECC is enabled the controller use a page layout that divides the page area into partitions (termed chunks or sectors), 16 or 8 chunks per page, depending on flash memory configuration. Each chunk has a data area and an ECC area. Current implemented ECC capability allows for some extra bytes in the end of a page (in spare area) for any purposes (termed Flags area).

NANFDCTRL2 supports 3 memory configurations:

- page size of 16 kbytes data size and 2208 bytes spare size,
  16 partitions of 1024 bytes chunk size and ECC 60 bits correctability.
- page size of 8 kbytes and 448 bytes spare size,
  16 partitions of 512 bytes chunk size and ECC 16 bits correctability.
- page size of 4 kbytes and 224 bytes spare size,
  8 partitions of 512 bytes chunk size and ECC 16 bits correctability.

The ECC bits are placed in the spare area and avoids using the first 2 bytes (to preserve original Bad Block Markings).

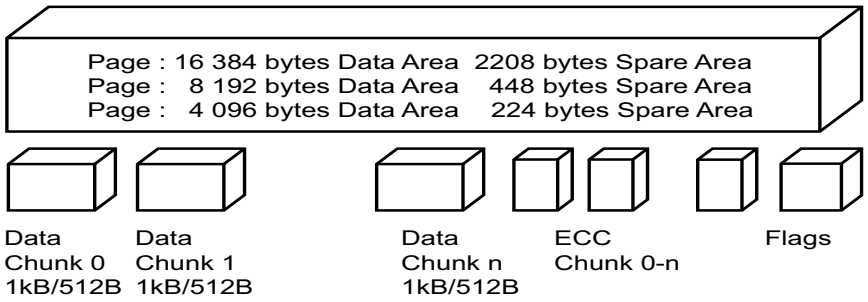

Figure 3. Page - page layout

1.3.2 Configuration and status accesses

Currently all configuration and status commands uses APB data-in/data-out registers except READ PARAMETER PAGE that is accessed over DMA:

- READ ID
- READ UNIQUE ID
- READ STATUS
- GET FEATURES
- LUN GET FEATURES
- SET FEATURES
- LUN SET FEATURES

1.3.3 Data accesses

The NANDFCTRL2 allows for different access modes and with optional hardware support for both ECC generation and randomization. If any of these support features are enabled the NANDFCTRL2 will automatically fill up the spare area. If ECC generation and randomization are disabled the spare area is not filled with data. If ECC is enabled and randomization is disabled the spare area will be used to store ECC bits for each chunk. If ECC is disabled and randomization is enabled the spare area will be filled with randomized data.

NANFDCTRL2 supports 3 different data accesses

- Access the whole data area, e.g. 16K data transfer size.
- Access data area with a chunk size, which must be on a chunk aligned column address.
- Access page register freely. For such accesses to the data area the ECC must be turned off. Free accesses are allowed to spare area with no restriction.
  If randomization is enabled the supplied data will be randomized. The randomizer will align randomized data to selected column address, i.e. randomizer is using column address as input to assure that randomization algorithm will produce correct randomization pattern dependent on given position.

When the whole data area is accessed the NANDFCTRL2 will loop internally on chunk size to fill the whole data area. When accesses are at chunk sizes, multiple accesses must be setup and commands Change_Write_Column and Change_Read_Column must be used to access page register at aligned chunk number for each access (except for the first chunk which is using ordinary command and setting skip cmd2 bit SC2, see register DCMD).

1.3.4 Single Event Functional Interrupt - SEFI

Note: SEFI functionality may be subjected to future changes and are preliminary.

The SEFI block handles timeouts in the event that a target does not respond.
All commands that use R/B# signal to detect when a command has been executed are monitored with its specific timing parameters.
It is possible to set both a counter value 1- 1000 and a resolution (off, ns, us, ms) for each timing parameter (tBERS, tPLEBSY, tPLPBSY, tPLRBSY, tPCBSY, tPROG, tR, tRCBSY).

The SEFI block monitors up to 8 R/B# signals and for each R/B# signal it is possible to assign it to one or several CE signals.

The protocol block issues a start counting and also informs about access type. Timing parameters are taken from APB registers and the counter starts counting down in specified resolution.
When a counter hits 0 an interrupt is issued for that R/B# counter, unless transaction finish via R/B#.

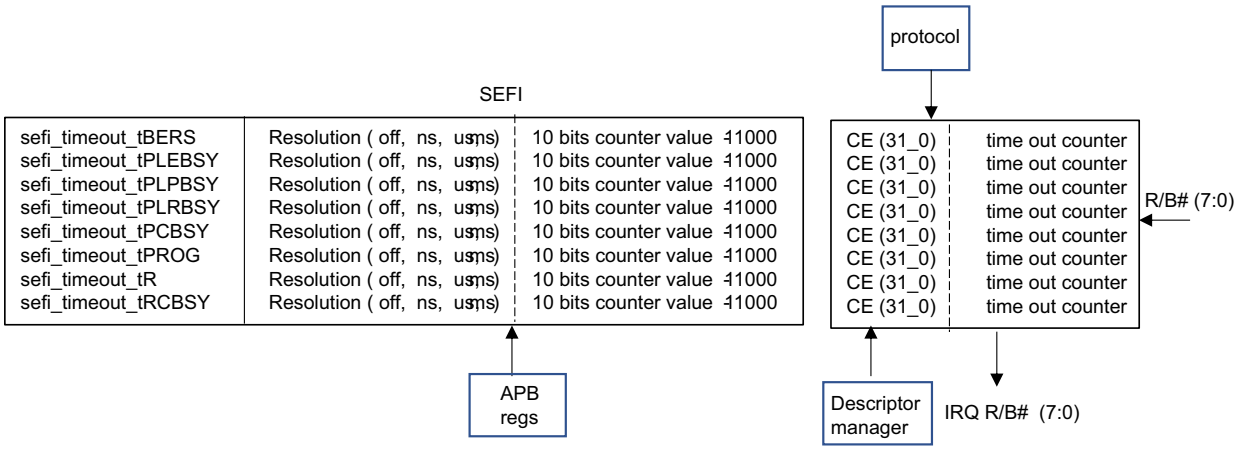

Figure 4. SEFI block

1.4 Operation - Typical use case

The steps that needs to be taken to access (i.e. send an ONFI 4.0 command to) the Flash memory devices are described below.
Refer to the Register section for more information about a specific register or register field.

The mandatory commands can be divided into following categories and are the only one necessary to use NANDFCTRL2 in its simplest mode:
- RESET
- READ ID and READ PARAMETER PAGE
- BLOCK ERASE
- READ STATUS
- PAGE PROGRAM and CHANGE WRTE COLUMN
- PAGE READ and CHANGE READ COLUMN

All commands are issued as pairs of CMD1 (*cmd1* - 1st Cycle) and CMD2 (*cmd2* - 2nd Cycle).
It is possible to set *skip_command_2* by a bit for some commands (see register) to chain with more commands, e.g. before issue an ending programming command.

0. Setup Core control 0 &1 register and timing register for tCCS and SEFI timeout.

1. Issue a reset CMD to all or selected targets. This can be done using either the RST bit in core control register 2 that resets all targets or select individual targets using *target_sel* and CMD1 = 0xFF

2. Perform a READ_ID for selected target.

3 Perform a READ PARAMETER PAGE for selected target (step 10 below is necessary).

4. Setup Core Control register accordingly with information from parameter page.
Including SEFI timeout parameters and other timing parameters.

5. Get/Set Feature (Target/LUN) if needed (step 8 below is necessary).

6. Make sure the Flash memory is not in write protect mode by reading the *WP* field in the *Core Status register 0,* otherwise the WP bit can be cleared to deselect Write Protection signal.

7. If the command requires a row address to be sent, write the row address to the *Descriptor row register*. Otherwise this step can be skipped.

8. If the command requires a column address to be sent, write the column address to the *coladdr* field of the *Descriptor column & size register*. Bits 7:0 of the *coladdr* field also need to be written with the one byte address used for SET FEATURES, GET FEATURES, READ ID, READ UNIQUE ID, and READ PARAMETER PAGE commands. Otherwise this step can be skipped.

9. If the command should include a data phase then set the size of the data by writing to the *size* field of the *Descriptor Column & Size register*. NOTE: See limitations for data accesses to page area.
This step is not necessary for the SET FEATURES, GET FEATURES, READ STATUS, and READ STATUS ENHANCED commands since they always have a fixed size data phase.

10. If data should be read or written over the AHB bus, i.e. Page Write or Page Read commands, the *Descriptor data pointer register* needs to be set with the correct memory address for DMA transfers.

11. Select which target the command should be sent to by writing to the *targsel* field in the *Descriptor target select register (which CE to activate)*

12. Write the command CMD1 or commands values CMD1 & CMD2 to the *Descriptor command register*, and set the control bits accordingly to usage.

13. Select if an interrupt should be generated when the command is finished, and start execution by writing to ‘1’ to *DT* bits in the *Core control 2 register.*

Once command execution has been started software can wait for an interrupt if the core was configured to generate one to monitor when the command is finished.

1.5 Registers

The core is programmed through registers mapped into APB address space.

Table 2. NANDFCTRL2 registers

APB address offset	Register
0x00	Core control 0 register
0x04	Core control 1 register
0x08	Core control 2 register
0x10	Core status 0 register
0x14	Core status 1 register
0x20	Capability 0 register
0x30	Programmable timing 0 register
0x34	Programmable timing 1 register
0x38	Programmable timing 2 register
0x3c	Programmable timing 3 register
0x40	SEFI timeout tBERS register
0x44	SEFI timeout tPLEBSY register
0x48	SEFI timeout tPLPBSY register
0x4c	SEFI timeout tPLRBSY register
0x50	SEFI timeout tPCBSY register
0x54	SEFI timeout tPROG register
0x58	SEFI timeout tR register
0x5c	SEFI timeout tRCBSY register
0x60	SEFI R/B# 0 CEs register
0x64	SEFI R/B# 1 CEs register
0x68	SEFI R/B# 2 CEs register
0x6c	SEFI R/B# 3 CEs register
0x70	SEFI R/B# 4 CEs register
0x74	SEFI R/B# 5 CEs register
0x78	SEFI R/B# 6 CEs register
0x7c	SEFI R/B# 7 CEs register
0x80	Descriptor next pointer low register
0x84	Descriptor next pointer high register
0x88	Descriptor data pointer low register
0x8c	Descriptor data pointer high register
0x90	Descriptor command register
0x94	Descriptor target select register
0x98	Descriptor row register
0x9c	Descriptor column & size register
0xa0	Descriptor status register
0xa4	Descriptor ECC status register
0xc0	Data-in 0 register
0xc4	Data-in 1 register
0xc8	Data-in 2 register
0xcc	Data-in 3 register

Table 2. NANDFCTRL2 registers

APB address offset	Register
0xe0	Data-out 0 register
0xe4	Data-out 1 register
0xe8	Data-out 2 register
0xec	Data-out 3 register

1.5.1 Core control 0 register

Table 3. 0x00 - CTRL0 - core control 0 register

31 25	24 16
RESERVED	C100
0	*
r	r

15 14	13	12	11	10	9	8	7	6 5 4 3 2	1	0
RESERVED	EE	RE	RSVD	LLM	DW	SEFI FLAG	EDO	RESERVED	WP	RES
0	1	1	0	0	0	0	0	0	1	0
r	rw	rw	r	rw	rw	rw	rw	r	rw*	r

31:25 Reserved.

24:16 Counter 100 ns (C100)
Number of clock cycles -1 it takes for system clock to count 100 ns.
0 - system clock of 10 Mhz
NOTE: Base counter for SEFI block. A generic constant.

15:14 Reserved

13 EDAC enable (EE)
0- EDAC (ECC) off
1- EDAC (ECC) on

12 Data randomization enable (RE)
0- Data randomization off
1- Data randomization on

11 Reserved

10 Linked list mode enable (LL) via descriptor registers
0 = inactive
NOTE: THIS FUNCTION IS NOT SUPPORTED FOR THIS REVISION OF THE IP

9 Data width (DW) - Sets the default data lane width.
Value taken from parameter page
0 = Core uses 8-bit data lanes.
This can be overridden for individual commands by setting the *dw* bit in the *Descriptor command* register. This bit is only available if the *dw16* bit in the *Capability register* is 1.
NOTE: THIS FUNCTION IS NOT SUPPORTED FOR THIS REVISION OF THE IP

8 SEFI flag output (SEFI FLAG)
0 = sets SEFI_flag signal to low
1 = sets SEFI_flag signal to high

7 EDO data output (EDO)

0 = EDO off
1 = EDO on

6:2 Reserved

1 Write protect (WP)
When this bit is set to 1 the core puts the Flash memory devices in write protect mode by asserting the *wp* signal. In write protect mode, the memories won't respond to PROGRAM or ERASE commands. If the core is active when software writes this bit there is a delay before the actual write protect signal goes low. Software can use the *wp* field in the *Core status register* to see when the signal has changed value. Reset value: 1

0 Reserved

1.5.2 Core control 1 register

Table 4. 0x04 - CTRL1 - Core control 1 register

31	16
RESERVED	
0	
r	

15	7	6	5	4	3	2	1	0
RESERVED		IRQ SEFI	IRQ CMD	IRQ DL	IRQ UL	IRQ ECC	IRQ DS	IRQ
0		1	1	1	1	1	1	1
r		rw	rw	rw	rw	rw	rw	rw

31:7 Reserved.

6 SEFI interrupt (IRQ-SEFI)
0 interrupt is disabled,
1 interrupt is enabled.
This interrupt is issued when there is an SEFI timeout for current transaction.

5 Invalid command interrupt enable (IRQ-CMD)
0 interrupt is disabled.
1 interrupt is enabled.
This interrupt is issued when there is an invalid command.

4 DMA downlink interrupt (IRQ-DL)
0 interrupt is disabled.
1 interrupt is enabled.
This interrupt is issued when there is an error in DMA downlink

3 DMA uplink interrupt (IRQ-UL)
0 interrupt is disabled.
1 interrupt is enabled.
This interrupt is issued when there is an error in DMA uplink

2 ECC interrupt (IRQ-ECC)
0 interrupt is disabled.
1 interrupt is enabled.
This interrupt is issued when there is an uncorrectable ECC error.

1 Descriptor interrupt (IRQ-DS).
0 interrupt is disabled.
1 interrupt is enabled.
This field should be set together with the descriptor *irq* field in *Descriptor command register.*

0 Interrupt enable (IRQ)
0 - all interrupts are disabled.
1 - all interrupts are enabled.

1.5.3 Core control 2 register

Table 5. 0x08 - CTRL2 - core control 2 register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														DT	RST
0														0	0
r														rw*	rw*

31:2 Reserved.

1 Descriptor trigger (DT)
1- Trigger descriptor command. When set the information in descriptor registers are executed.
This bit is cleared automatically when command has been executed.

0 Software reset (RST)
1- The NANDFCTRL2 is reset, and a RESET (0xFF) command is issued to all targets on all attached Flash memory devices.
This bit is cleared automatically when a reset command has been executed.

1.5.4 Core status 0 register

Table 6. 0x10 - STS0 - Core status 0 register

31	16
RESERVED	
0	
r	

15	3	2	1	0
RESERVED	DA	RSVD	WP	RDY
0	0	0	1	*
r	r	r	r	r

31:4 Reserved.

3 Descriptor active (DA)
1 - Descriptor active.

2 Reserved

1 Write protect (WP) - Shows if the Flash memory devices are in write protect mode or not.
0 = Not in write protect mode.
1 = In write protect mode.
Reset value: 1

0 Core ready (RDY) - After a power up / reset this bit will be cleared.
Once the core is done with it's reset procedure (waiting for the ready signal from the Flash memory device or issuing a RESET command) this bit is set to 1.
Reset value is 0 while reset is active and 1 shortly after it is released.

1.5.5 Core status 1 register

Table 7. 0x14 - STS1 - Core status 1 register

31	16
RESERVED	
0	
r	

15	7	6	5	4	3	2	1	0
RESERVED		SEFI	CMD	DL	UL	ECC	DF	IRQ
0		0	0	0	0	0	0	0
r		wc	wc	wc	wc	wc	wc	wc

31:7 Reserved.

6 SEFI timeout (SEFI)
This is set when there is an SEFI timeout.

5 Invalid command (CMD)
This is set when there is an invalid command.

4 Donwlink DMA error (DL).
This is set when there is an error in DMA Downlink

3 Uplink DMA error (UL).
This is set when there is an error in DMA Uplink

2 ECC error (ECC).
This is set when there is an ECC error.

1 Descriptor finished (DF).
This is set when the descriptor is finished.

0 Interrupt (IRQ)
This is set when an interrupt have been issued.

1.5.6 Capability 0 register

Table 8. 0x20 - CAP0 - Capability 0 register

31 28	27 16
REV	RESERVED
*	0
r	r

15 8	7	6	5 0
RESERVED		DW16	NTARGS
0		*	*
r		r	r

31:28 Revision (REV) - Indicates the revision of the core.

27:7 Reserved.

6 16-bit memory support (DW16) Determined by the generic *dwidth16*.
If this bit is 0 the core only support 8-bit memories.
If this bit is 1 the core support both 8-bit and 16-bit memories.

NOTE: THIS FUNCTION IS NOT SUPPORTED FOR THIS REVISION OF THE IP!

5:0 Number of targets per device (NTARGS) - This field indicates how many individual targets the core can access, i.e. number of chip enable signals.
Determined by the generic *ce*.
A Flash memory device can have one or more targets, each with an individual chip enable signal.
Value 1-32

1.5.7 Programmable timing 0 register

Table 9. 0x30 - TME0 - Programmable timing 0 register

31 29	28 20	19	18 16
RESERVED	tCS	RES	tCS3 (8:6)
0	0	0	0
r	rw	r	rw

15 10	9	8 0
tCS3 (5:0)	RES	tWW
0	0	0
rw	r	rw

31:29 Reserved.

28:20 CE_n setup time (tCS).

Length of tCS in clock cycles, minus one.

19 Reserved

18:10 CE_n setup time (tCS3) for data input after CE_n has been high for greater than 1us.

Length of tCS in clock cycles, minus one.

NOTE: THIS FUNCTION IS NOT SUPPORTED FOR THIS REVISION OF THE IP! (Errata 5, ONFI 4.0)

9 Reserved

8:0 WP_n transition to WE low (tWW).

Length of tWW in clock cycles, minus one.

1.5.8 Programmable timing 1 register

Table 10. 0x34 - TME1 - Programmable timing 1 register

31 30	29 20	19	18 16
RESERVED	tRHW	RES	tRR (7:5)
0	0	0	0
r	rw	r	rw

15 11	10	9 0
tRR (4:0)	RES	tWB
0	0	0
rw	r	rw

31:30 Reserved.

29:20 RE_n high to WE_n low (tRHW).

Length of tRHW in clock cycles, minus one.

19 Reserved.

18:11 Ready to RE_n low (tRR).

Length of tRR in clock cycles, minus one.

10 Reserved.

9:0 WE_n high to SR[6] low (tWB)

Length of tWB in clock cycles, minus one.

1.5.9 Programmable timing 2 register

Table 11. 0x38 - TME2 - Programmable timing 2 register

31	30	20	19	16
RES	tADL		tWHR (8:5)	
0	0		0	
r	rw		rw	

15	11	10	0
tWHR (4:0)		tCCS	
0		0	
rw		rw	

31	Reserved.
30:20	ALE to data loading time (tADL). Length of tADL in clock cycles, minus one.
19:11	WE_n high to RE_n low (tWHR). Length of tWHR in clock cycles, minus one.
10:0	Change Column setup time (tCCS). Length of tCCS in clock cycles, minus one.

1.5.10 Programmable timing 3 register

Table 12. 0x3C - TME3 - Programmable timing 3 register

31	24	23	16
tWH		tWP	
0		0	
rw		rw	

15	8	7	0
tREH		tRP	
0		0	
rw		rw	

31:24	WE_n high hold time (tWH). Length of tWH in clock cycles, minus one.
23:16	WE_n pulse width (tWP). Length of tWP in clock cycles, minus one.
15:8	RE_n high hold time (tREH). Length of tREH in clock cycles, minus one.
7:0	RE_n pulse width (tRP). Length of tRP in clock cycles, minus one.

1.5.11 SEFI timeout registers

These registers holds the resolution and timer value for different R/B# timing parameters,
(8 registers in order : tBERS, tPLEBSY, tPLPBSY, tPLRBSY, tPCBSY, tPROG, tR, tRCBSY).

Table 13. 0x40 -0x5c - SEFI timeout registers

31	18	17	16
RESERVED		Resolution	
0		0	
r		rw	

15	9	0
RESERVED	Counter value	
0	0	
r	rw	

31:17	Reserved
17:16	Resolution 0 - off 1 - ns 2 - us 3 - ms
15:8	Reserved
9:0	Counter value 1-1000

1.5.12 SEFI R/B# registers (R/B# 0 - R/B# 7)

These 8 register maps the CE information about which CE each R/B# (7:0) is connected to,
(8 registers in order RB_n_0 to RB_n_7)

Table 14. 0x60 - 0x7c - SEFI R/B# 0 - SEFI R/B# 7 registers

31	16
CE	
0	
rw	

15	0
CE	
0	
rw	

31:0	CE (31:0) CE signals 0 - CE is not attached to R/B# 1 - CE is attached to R/B#

1.5.13 Descriptor next pointer low register

Table 15. 0x80 - DNPL - Descriptor next pointer low register

31	16
Next pointer(31:16)	
0000000000000000	
rw	

15	0
Next pointer(15:0)	
0000000000000000	
rw	

31:0 Next pointer low (31:0) - Least significant part of memory address.
NOTE: THIS FUNCTION IS NOT SUPPORTED FOR THIS REVISION OF THE IP

1.5.14 Descriptor next pointer high register

Table 16. 0x84 - DNPH - Descriptor next pointer high register

31	16
Next pointer(63:48)	
0000000000000000	
rw	

15	0
Next pointer(47:32)	
0000000000000000	
rw	

31:0 Next pointer high (63:32) Most significant part of memory address.
NOTE: THIS FUNCTION IS NOT SUPPORTED FOR THIS REVISION OF THE IP

1.5.15 Descriptor data pointer low register

Table 17. 0x88 - DDPL - Descriptor data pointer low register

31	16
Data pointer(31:16)	
0000000000000000	
rw	

15	0
Data pointer(15:0)	
0000000000000000	
rw	

31:0 Data pointer low (31:0) - Least significant part of memory address. Address in memory from where DMA reads or writes data.

1.5.16 Descriptor data pointer high register

Table 18. 0x8c - DDPH - Descriptor data pointer high register

31	16
Data pointer(63:48)	
0000000000000000	
rw	

15	0
Data pointer(47:32)	
0000000000000000	
rw	

31:0 Data pointer high (63:32) Most significant part of memory address. Address in memory from where DMA reads or writes data.

1.5.17 Descriptor command register

Table 19. 0x90 - DCMD - Descriptor Command register

31 24	23 16
CMD2	CMD1
00000000	00000000
rw	rw

15 12	11	10	9	8	7	6	5 4	3 2	1	0
RESERVED	DD	ED	RD	DWO	RSVD	SC2	SUB CMD	RESERVED	IRQ DS	EN
r	0	0	0	0	0	0	0	r	0	0
0	rw	rw	rw	rw	r	rw	rw	0	rw	rw

31:24 Second command phase (CMD2)
If the command to execute is a two byte (two phase) command then software should write the second byte of the command to this field.
The core ignores this field for commands that only have one command phase, or if SC2 is set.

23:16 First command phase (CMD1)
Software should write this field with the first byte of the command to execute.

15:12 Reserved

11 Data DMA disable (DD)
0 - enable data read /write over DMA
1 - disable data read/write over DMA.
Data is read via data-in registers or data is written via data-out registers via APB when DD=1.

10 EDAC disable (ED)
0 - the *EE* bit in the *Core control register* 0 is used as specified in its description.
1 - the *EE* bit in the *Core control register* 0 is ignored and EDAC is disabled.for the current descriptor.

9 Data Randomization disable (RD)
0 - the *RE* bit in the *Core control register* 0 is used as specified in its description.
1 - the *RE* bit in the *Core control register* 0 is ignored and data randomization is disabled.for the current descriptor.

8 Data width override (DWO)
0 - the *dw* bit in the *Core control register* 0 is used as specified in its description.
1 - the *dw* bit in the *Core control register* 0 is ignored and 8 bits access is forced for the current descriptor.
This bit is only available if the *dw16* bit in the *Capability register* is 1.
Reset value 0.
NOTE - THIS FEATURE IS NOT SUPPORTED FOR THIS REVISION OF THE IP.

7 Reserved

6 Skip second command phase (SC2)
If this bit is set and a program command (PAGE PROGRAM, COPYBACK PROGRAM, or CHANGE WRITE COLUMN) is being executed the core skips the second command phase for that command and jumps back to idle state once all data has been written.
This is done in order to support the CHANGE WRITE COLUMN command, which needs to be executed in between the first and second phase of the program command
If a CHANGE WRITE COLUMN command is required during a PAGE PROGRAM, the PAGE PROGRAM command is issued with SC2 set to 1.
After that when issuing a CHANGE WRITE COLUMN command if more than one CHANGE WRITE COLUMN commands needs to be issued all the commands but the last one is issued with SC" set to 1.

5:4 Sub command (SUB_CMD)
There are different combinations of CMD1 = 0x80 or 0x85 and they have different timing parameters.
0 - Ordinary usage of CMD1 (all different PAGE PROGRAM commands)
1 - Small Data Move
2 - Change Write Column
3 - Change Row Address

3:2 Reserved.

Table 19. 0x90 - DCMD - Descriptor Command register

1	Request IRQ (IRQ-DS) 0 - don't send IRQ when command is finished 1 - send IRQ when command is finished.
0	Descriptor Enable (EN) 0 - The descriptor is disregarded by the core and the descriptor registers can be updated freely. 1 - The descriptor is owned by the core and no descriptor registers should be updated until the core clears it. Setting the DT bit of the Core Control 2 register will start execute the descriptor.

1.5.18 Descriptor target select register

Table 20. 0x94 - DTARSEL - Descriptor target select register

31	16
TARGSEL	
0	
rrw	

15	0
TARGSEL	
0	
rw	

31:0	Target select (TARGSEL) The core uses this field to select which targets from to send the command to (which chip enable (CE) signals to assert). The least significant bit in this field corresponds to the first target (CE(0)) etc. The actual number of bits implemented equals the *ntargs* field in the *Capability register*.

1.5.19 Descriptor row register

Table 21. 0x98- DROW - Descriptor row register

31	24	23	16
RESERVED		ROWADDR(23:16)	
0		00000000	
r		rw	

15	0
ROWADDR(15:0)	
0000000000000000	
rw	

31:24 Reserved.

23:0 Row Address (ROWADDR)
This field sets the three byte row address, which is used to address LUNs, blocks and pages.
As described in the ONFI 4.0 specification the least significant part of the row address is the page address, the middle part block address, and the most significant part is the LUN address.
(7:0) first address byte at PHY,
(15:8) second address byte at PHY,
(23:16) third address byte at PHY.
Exactly how many bits that are used for each part of the address depends on the architecture of the Flash memory.
Software needs to write this field prior to issuing any command that has an address phase that includes the row address.
The core ignores this field if the command doesn't use the row address.

1.5.20 Descriptor column & size register

Table 22. 0x9c - DCOLSIZE - Descriptor column & size register

31	16
SIZE	
0000000000000000	
rw	

15	0
COLADDR	
0000000000000000	
rw	

31:16 Command data size (SIZE)
If a command has a data out or data in phase then software needs to set this field to the size of the data that should be read / written.
Software does not need to set this field for the commands SET FEATURES, GET FEATURES, READ STATUS, or READ STATUS ENHANCED their data phases are always the same size.
The core also ignores this field if the command issued doesn't have a data phase, as for example BLOCK ERASE.

15:0 Column address (COLADDR)
This field sets the two byte column address, which is used to address into a Flash memory page. See the ONFI 4.0 specification for more information about column address.
(7:0) first column byte at PHY
(15:8) second column byte at PHY
Software needs to write this field for those commands that have an address phase that includes the column address, as well as for those special commands that only have a one byte address phase (SET FEATURES; GET FEATURES; READ ID, READ UNIQUE ID, and READ PARAMETER PAGE). This field is ignored by the core if the command only uses the row address.

1.5.21 Descriptor status register

Table 23. 0xa0 - DSTS - Descriptor Status register

31	16
RESERVED	
0	
r	

15	2	1	0
RESERVED		UE	INV
0		0	0
r		r	r

31:2 Reserved.

1 Uncorrectable error (UE)
0 - No uncorrectable error
1 - Uncorrectable error

0 Invalid command (INV)
This bit is set to one if an invalid command is written to the *Descriptor command register* when the *DT* bit is written. This bit is cleared automatically once a new command is started. Reset value: 0

1.5.22 Descriptor ECC status register

Table 24. 0xa4 - DECCSTS - Descriptor ECC status register

31	16
RESERVED	
0	
r	

15	12	11	0
		CE	
		0	
		r	

31:12	Reserved.
11:0	Corrected errors (CE) Number of corrected errors

1.5.23 Data-in 0-3 register

Data-in register 0 holds the data bytes (Data 0-3) read from the Flash memory device for each command reading data.

Data-in register 1holds the data bytes (Data 4-7) read from the Flash memory device for each command reading data.

Data-in register 2 holds the data bytes (Data 8-11) read from the Flash memory device for each command reading data.

Data-in register 3 holds the data bytes (Data 12-15) read from the Flash memory device for each command reading data.

Data 0 is first byte read from Flash.

Note that this is independent if the data is also transferred to the AHB bus or not.

Table 25. 0xc0 - 0xcc - Data-in 0 - Data-in 3 registers little endian

31 – 24	23 – 16
DATA 3, 7, 11, 15	DATA 2, 6, 10, 14
00000000	00000000
r	r

15 – 8	7 – 0
DATA 1, 5, 9, 13	DATA 0, 4, 8, 12
00000000	00000000
r	r

31:24	DATA 3, 7, 11, 15
24:16	DATA 2, 6, 10, 14
15:8	DATA 1, 5, 9, 13
7:0	DATA 0, 4, 8, 12

Table 26. 0xc0 - 0xcc - Data-in 0 - Data-in 3 registers big endian

31 – 24	23 – 16
DATA 0, 4, 8, 12	DATA 1, 5, 9, 13
00000000	00000000
r	r

15 – 8	7 – 0
DATA 1, 5, 9, 13	DATA 3, 7, 11, 15
00000000	00000000
r	r

31:24	DATA 0, 4, 8, 12
24:16	DATA 1, 5, 9, 13
15:8	DATA 2, 6, 10, 14
7:0	DATA 3, 7, 11, 15

1.5.24 Data-out 0-3 register

Data-out register 0 holds the data bytes (Data 0-3) to be written to the Flash memory device for each command writing data.

Data-out register 1holds the data bytes (Data 4-7)to be written to the Flash memory device for each command writing data.

Data-out register 2 holds the data bytes (Data 8-11) to be written to the Flash memory device for each command writing data.

Data-out register 3 holds the data bytes (Data 12-15) to be written to the Flash memory device for each command writing data

Data 0 is first byte written from Flash.

Data out register are used when DMA is disabled and for commands SET (LUN) FEATURE.

Table 27. 0xe0 - 0xec - Data-out 0 - Data-out 3 registers little endian

31 – 24	23 – 16
DATA 3, 7, 11, 15	DATA 2, 6, 10, 14
00000000	00000000
r	r

15 – 8	7 – 0
DATA 1, 5, 9, 13	DATA 0, 4, 8, 12
00000000	00000000
r	r

31:24	DATA 3,7,11,15
24:16	DATA 2,6,10,14
15:8	DATA 1,5,9,13
7:0	DATA 0,4,8,12

Table 28. 0xe0 - 0xec - Data-out 0 - Data-out 3 registers big endian

31 – 24	23 – 16
DATA 0, 4, 8, 12	DATA 1, 5, 9, 13
00000000	00000000
r	r

15 – 8	7 – 0
DATA 2, 6, 10, 14	DATA 3, 7, 11, 15
00000000	00000000
r	r

31:24	DATA 0, 4, 8, 12
24:16	DATA 1, 5, 9, 13
15:8	DATA 2, 6, 10, 14
7:0	DATA 3, 7, 11, 15

1.6 Vendor and device identifiers

The core has vendor identifier 0x01 (Cobham Gaisler) and device identifier 0x0C5.
For description of vendor and device identifiers see GRLIB IP Library User's Manual.

1.7 Implementation

1.7.1 Reset

The core changes reset behavior depending on settings in the GRLIB configuration package (see GRLIB User's Manual).

1.7.2 Endianess

The core changes endianness behaviour depending on the settings in the GRLIB configuration package (see GRLIB User's Manual).

1.8 Configuration options

Table 29 shows the configuration options of the core (VHDL generics).

Table 29. Configuration options

Generic name	Function	Allowed range	Default
hindex	AHB master index	0 - NAHBMST-1	0
pindex	APB slave index	0 - NAPBSLV-1	0
pirq	APB irq number.	0 - NAHBIRQ-1	0
paddr	ADDR field of the APB BAR.	0 - 16#FFF#	0
pmask	MASK field of the APB BAR.	0 - 16#FFF#	16#FFF#
memtech	Memory technology used for the buffers.	0 - NTECH	0
flash_mem	Flash memory configuration 1- Memory 16 kB data area & 2208 B spare area 2 - Memory 8 kB data area & 448 B spare area 3 - Memory 4 kB data area & 448 B spare are	1 - 3	1
ce	Number of CE_n = Number of chip select signals connected to the core.	1 - 32	32
dwidth16	0 = Core implements 8-bit data lanes. 1 = Core implements 16-bit data lanes.	0 - 1	0
cycles_100ns	Number of system clock periods - 1 for 100 ns. (default 9 is for system clock of 100 Mhz)	0 - 511	9
nsync	Number of synchronization registers on R/B input. Data input is always synchronized through one set of registers.	0 - 3	2
cdc	Clock domain crossing (NOTE: for future use - shall be 2)	1 - 63	1
oepol	Polarity of pad output enable signal.	0 - 1	0

1.9 Signal descriptions

Table 30 shows the interface signals of the core (VHDL ports).

Table 30. Signal descriptions

Signal name	Field	Type	Function	Active
RSTN	N/A	Input	Reset	Logical 0
CLK	N/A	Input	Clock	-
APBI	*	Input	APB slave input signals	-
APBO	*	Output	APB slave output signals	-
AHBMI	*	Input	AHB master input signals	-
AHBMO	*	Output	AHB master output signals	-
SEFI_CTRL	N/A	Output	SEFI output control signal	Logical 1
NANDFI	rb_n (7:0)	Input	Ready/Busy signal	Logical 0
	di (7:0) di(15:8) [3]	Input	Data input (used both for 8-bit and 16-bit lanes)	-
NANDFO	ce_n(31:0) [2]	Output	Chip enable	Logical 0
	wp_n	Output	Write protect	Logical 0
	do (7:0) do(15:8) [3]	Output	Data output (used both for 8-bit and 16-bit lanes)	-
	cle	Output	Command latch enable	Logical 1
	ale	Output	Address latch enable	Logical 1
	we_n	Output	Write enable	Logical 0
	re_n	Output	Read enable	Logical 0
	oe	Output	Output enable	[1]

* see GRLIB IP Library User's Manual

[1] The polarity of the output enable signal is implementation dependent.

[2] The number of CE is set as generic *ce*

[3] The number of data bits i.e. die x8 or x16 is set by generic *dwidth16*

1.10 Signal definitions and reset values

The signals and their reset values are described in table 31.

Table 31. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
rb_n (7:0)	Input	Ready/Busy signal	-	-
d(7:0)	Input/Output	Data (used both for 8-bit and 16-bit lanes)	-	-
d(15:8) [3]	Input/Output	Data (upper byte for 16-bit lanes)	-	-
ce_n (31:0) [2]	Output	Chip enable	Logical 0	Logical
wp_n	Output	Write protect	Logical 0	Logical 0
cle	Output	Command latch enable	Logical 1	Logical 0
ale	Output	Address latch enable	Logical 1	Logical 0
we_n	Output	Write enable	Logical 0	Logical 1
re_n	Output	Read enable	Logical 0	Logical 1
oe	Output	Output enable	[1]	[1]
sefi_ctrl	Output	SEFI output cntrl	Logical 1	Logical 0

[1] The polarity of the output enable signal is implementation dependent.

[2] The number of CE is set as generic *ce*

[3] The number of data bits i.e. die x8 or x16 is set by generic *dwidth16*

1.11 Library dependencies

Table 32 shows the libraries used when instantiating the core (VHDL libraries).

Table 32. Library dependencies

Library	Package	Imported unit(s)	Description
GRLIB	AMBA	Signals	AMBA signal definitions
GAISLER	NANDFCTRL2PKG	Signals, component	Component declaration
TECHMAP	GENCOMP	components	Technology mapping

1.12 Instantiation

This example shows how the core can be instantiated. The instantiated core has all its generics, except *hsindex, pindex*, *paddr*, and *pirq* at their default values. The impact of the generics can be seen in table 29.

```
library ieee, grlib, gaisler;
use ieee.std_logic_1164.all;
use grlib.amba.all;
use gaisler.nandfctrl2pkg.all;

entity nandfctrl2_ex is
  port (
         clk :           in    std_ulogic;
         rstn :          in    std_ulogic
         nandf_rb_n :    in    std_logic_vector (7 downto 0);
         nandf_d :       inout std_logic_vector(15 downto 0);
         nandf_ce_n :    out   std_logic_vector(31 downto 0);
         nandf_wp_n :    out   std_ulogic;
         nandf_ale :     out   std_ulogic;
         nandf_cle :     out   std_ulogic;
         nandf_we_n :    out   std_ulogic;
         nandf_re_n :    out   std_ulogic;
         nandf_sefi :    out   std_logic );
end;

architecture rtl of nandfctrl2_ex is

  -- AMBA signals
  signal apbi  : apb_slv_in_type;
  signal apbo  : apb_slv_out_vector := (others => apb_none);
  signal ahbmi : ahb_mst_in_type;
  signal ahbmo : ahb_mst_out_vector := (others => ahbm_none);

  -- NANDFCTRL signals
  signal nandfo : nandfctrl2_out_type;
  signal nandfi : nandfctrl2_in_type;
  signal nandf_sefi: std_logic;

begin

...

  -- NANDFCTRL core
  nand0 : nandfctrl2
    generic map (hindex => 1, pindex => 10, paddr => 10, pirq => 10, cdc => 2)
    port map (rstn, clk, apbi, apbo(10), ahbmi, ahbmo(10), nandfi, nandfo, nandf_sefi);

  -- Pads for NANDFCTRL core
  nf2_d : iopadv generic map (tech => padtech, width => 16)
                  port map (nandf_d, nandfo.do, nandfo.oe, nandfi.di);
  nf2_rb_n : inpad generic map (tech => padtech, width => 8)
                  port map (nandf_rb_n, nandfi.rb_n);
  nf2_ce_n : outpad generic map (tech => padtech, width => 32)
                  port map (nandf_ce_n, nandfo.ce_n);
  nf2_we_n : outpad generic map (tech => padtech)
                  port map (nandf_we_n, nandfo.we_n);
  nf2_re_n : outpad generic map (tech => padtech)
                  port map (nandf_re_n, nandfo.re_n);
  nf2_cle : outpad generic map (tech => padtech)
                  port map (nandf_cle, nandfo.cle);
  nf2_ale : outpad generic map (tech => padtech)
                  port map (nandf_ale, nandfo.ale);
  nf2_wp_n : outpad generic map (tech => padtech)
                  port map (nandf_wp_n, nandfo.wp_n);
  nf2_sefi_op: outpad generic map (tech => padtech)
                  port map (nandf_sefi_flag, nandf_sefi);

end;
```

Cobham Gaisler AB
Kungsgatan 12
411 19 Göteborg
Sweden
www.caes.com/gaisler
sales@gaisler.com
T: +46 31 7758650