CoreAhbNvm Datasheet

DirectCore

Intended Use

- Provides AHB Hardware Interface and CFI Software Interface to the Embedded Nonvolatile Memory (NVM) Blocks within Fusion Devices

Key Features

- Supplied in SysBASIC Core Bundle
- Provides an Industry-Standard Software Interface to Actel Fusion® Flash Memory
- Implements a Subset of the Common Flash Memory Interface Specification Release 2.0
- Implements Standard Slave AHB Bus Hardware Interface
- Supports Read, Automatic Write and Erase, and Status Operations
- 32-Bit Interface, Allowing Byte, Half-Word, or Word Accesses to NVM
- Low Tile Count (resource utilization)
- Ability to Logically Merge Multiple Fusion NVM Blocks into One Large Area of NVM
- NVM Spare Page Access

Supported Families

- Fusion (AFS, M1AFS, M7AFS)

Core Deliverables

- VHDL and Verilog Delivered as Plain Text or Obfuscated RTL via Actel CoreConsole IP Deployment Platform
- Unit Test Delivered as CoreMP7 Bus Functional Model (BFM) Scripts, and Example AHB-Based System
- Core Verification
- User Can Easily Modify User Testbench Using Existing Format to Add Custom Tests

Contents

General Description

CoreAhbNvm provides an AHB bus interface to the embedded flash memory blocks within Fusion devices. The software, running on an AHB-based microprocessor, will be able to communicate to the embedded flash memory (read, write, and erase). This IP core is targeted to provide a functional subset of the Common Flash Interface (software interface only) with a design emphasis given to minimize design size. CoreAhbNvm supports all devices in the Fusion family. Note that this datasheet focuses on the operation of the CoreAhbNvm and does not provide detail on the structure or the behavior of the Fusion flash memory. Refer to the Actel *Fusion Family of Mixed-Signal FPGAs* datasheet for details on the Fusion flash memory.

Device Utilization and Performance

Table 1 • **CoreAhbNvm Device Utilization and Performance**

Tiles	Performance (MHz)
436	62 MHz (256, 512 k memory) 55 MHz (1 M memory)

I/O Signal Descriptions

The port signals for the CoreAhbNvm macro are given in Table 2.

Table 2 • **CoreAhbNvm Signal Descriptions**

Signal	Direction	Description
HCLK	Input	Bus Clock. This clock times all bus transfers and all signal timings.
HRESETn	Input	Reset. The bus reset signal is active LOW and is used to reset the system and the bus. This is the only active low AHB signal.
HADDR[20:0]	Input	Address. Bit 20, when 1, denotes a spare page access.
HWRITE	Input	Transfer direction. When high, this signal indicates a write transfer and when low a read transfer.
HSIZE[2:0]	Input	Transfer size. Indicates the size of the transfer, which can be byte (8-bit), halfword (16-bit), or word (32-bit). 000 – Byte 001 – Half word 010 – Word
HWDATA[31:0]	Input	32-bit data from the master
HREADYIN	Input	Ready signal from all other AHB slaves
HSEL	Input	Combinatorial decode of HADDR, which indicates that this slave is currently selected.
HRDATA[31:0]	Output	32-bit data written back to the master
HREADY	Output	Transfer done. When high, the HREADY signal indicates that a transfer has finished on the bus. This signal can be driven low to extend a transfer.
HRESP[1:0]	Output	Transfer response, which has the following meanings: 00 – Okay 01 – Error 10 – Retry 11 – Split
HTRANS[1:0]	Input	Type of transfer 00 – Idle 01 – Busy 10 – Non-sequential 11 – Sequential

Generic/Parameter Descriptions

CoreAhbNvm has one generic (VHDL) or one parameter (Verilog), called NVM_INSTANCES. This can take on the values shown in Table 3.

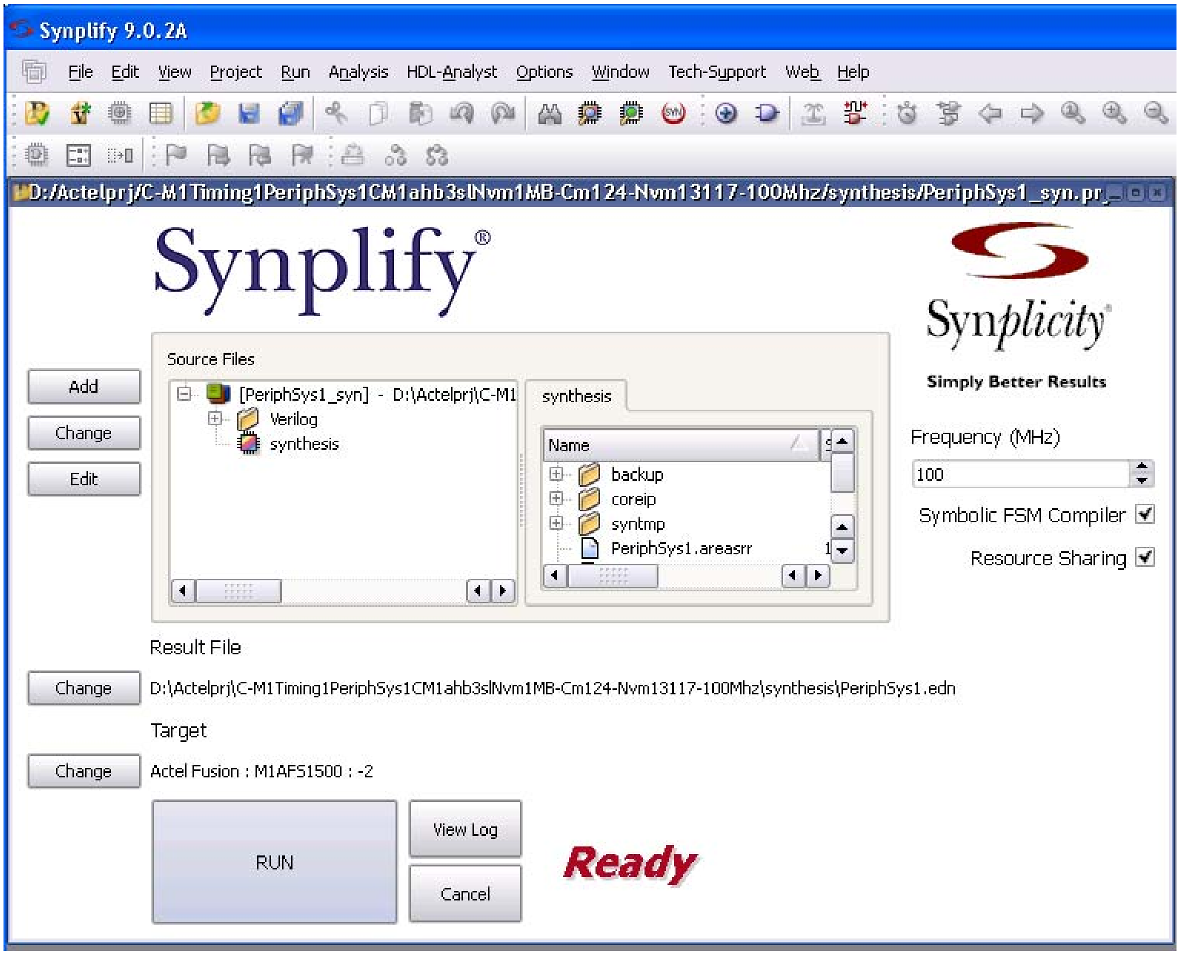
If instantiated within CoreConsole, this generic/parameter is set by selecting a value from a pull-down box within the configuration window.

Table 3 • **Generic/Parameter Description**

NVM_INSTANCES	NVM Size
1	256 kbytes
2	512 kbytes
4	1 Mbyte

Maximizing Performance for Large Memories

System performance degrades for NVM size greater than 256 kbytes. To maximize performance, constrain the design to 100 MHz within the Synplicity® synthesis tool (Figure 1). Under Layout Options, Figure 2 on page 4, select **Advanced**. Then choose **High effort layout** in the Advanced Layout Options, Figure 3 on page 4.



D:/Actelprj/C-M1Timing1PeriphSys1CM1ahb3slNvm1MB-Cm124-Nvm13117-100Mhz/synthesis/PeriphSys1_syn.pr

[PeriphSys1_syn] - D:\Actelprj\C-M1
Verilog
synthesis

PeriphSys1.areasrr

Frequency (MHz): 100

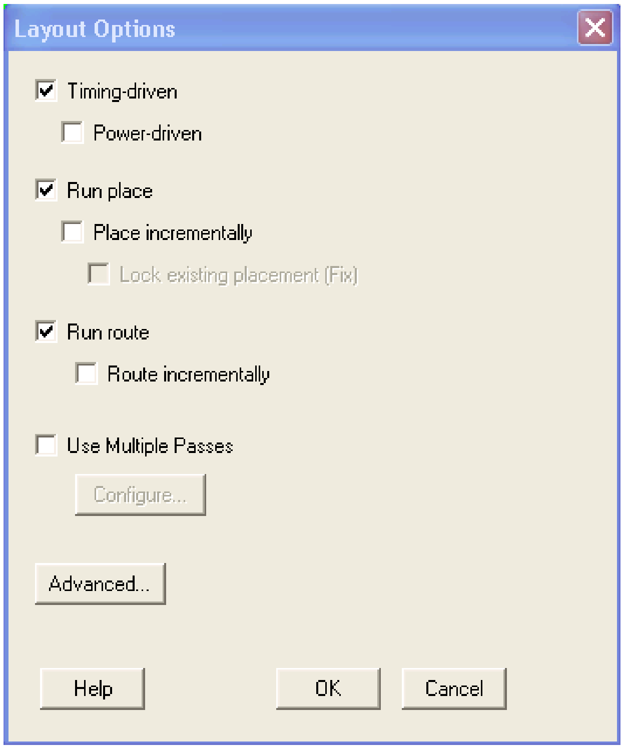
Symbolic FSM Compiler ☑

Resource Sharing ☑

Result File: D:\Actelprj\C-M1Timing1PeriphSys1CM1ahb3slNvm1MB-Cm124-Nvm13117-100Mhz\synthesis\PeriphSys1.edn

Target: Actel Fusion : M1AFS1500 : -2

Figure 1 • **Synplicity Constraints**



Layout Options

☑ Timing-driven

☐ Power-driven

☑ Run place

☐ Place incrementally

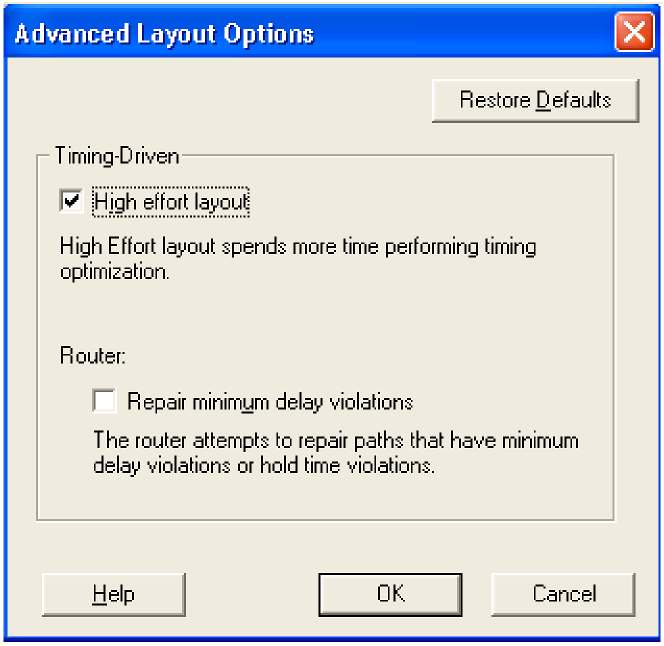
☐ Lock existing placement (Fix)

☑ Run route

☐ Route incrementally

☐ Use Multiple Passes

Figure 2 • **Layout Options**



Advanced Layout Options

Timing-Driven

☑ High effort layout

High Effort layout spends more time performing timing optimization.

Router:

☐ Repair minimum delay violations

The router attempts to repair paths that have minimum delay violations or hold time violations.

Figure 3 • **Advanced Layout Options**

Supported CFI Commands

CoreAhbNvm supports the Read, Automatic Erase, Automatic Write, and Status CFI Operations. The command descriptions are summarized in Table 4.

Table 4 • **Command Descriptions**

Command[9]	No. of Bus Cycles	First Bus Cycle			Second Bus Cycle			Notes
		Operation	Address	Data	Operation	Address	Data	
Read Array	1 or ≥ 2	Write	X	FFh	Read	AA	AD	1
Read Status	2	Write	X	70h	Read	X	SD	
Clear Status	1	Write	X	50h				
Erase Page	2	Write	PA	20h	Write	PA	D0h	2
Single-Write	2	Write	PA	40h	Write	AA	AD	3
Multi-Write	≥ 2	Write	PA	E8h	Write	PA	N	4, 5

Legend: X = any address within the device, QA = Query Address, QD = Query Data, IA = Identifier Address,
ID = Identifier Data, AA = Array Address, AD = Array Data, SD = Status Data, PA = Any Address within the page

Notes:

1. The Write portion of the Read Array command is only needed if not already in Read Array mode.
2. The Erase Page operation will fail if the page is locked.
3. The page portion of the address is ignored for the second bus cycle.
4. The page specified by AA is the page the data will be written to. The page portion of the address is ignored once the multi-write command has been sent (note that this means that writes will wraparound onto the same current page if the page address goes outside the PA specified with the first bus cycle).
5. N is the number of elements (bytes/words/double words) – 1 to be written to the write buffer. Expected count ranges are N = 00h to N = 7Fh (e.g., 1 to 128 bytes) in 8-bit mode, N = 00h to N = 003Fh in 16-bit mode, and N = 00h to N = 1Fh in 32-bit mode. Bus cycles 3 and higher are for writing data into the write buffer. The confirm command (D0h) is expected after exactly N + 1 Write cycles; any other command at that point in the sequence will prevent the transfer of the buffer to the array (the write will be aborted).
6. All new commands are ignored while the device is busy.

Reads/Writes

CoreAhbNvm Read operations, other than Read Array, are always preceded by a Write command to set up the read sequence. A preceding Write is only required for the Read Array operation when the device is not already in Read Array mode. The Fusion flash memory device contains a 16-byte read page buffer that enables fast data transfers.

Accessing Spare Pages

Assert the upper address bit, HADDR[20], to access a spare page. In this instance, the page address (HADDR[11:7]) is ignored and the spare page for the sector denoted by HADDR[17:12] is accessed.

Read Array Command

The algorithm for the Read command is shown in Figure 4. CoreAhbNvm comes out of reset in Read Array mode and the Read Array command is not required to read the array after reset.

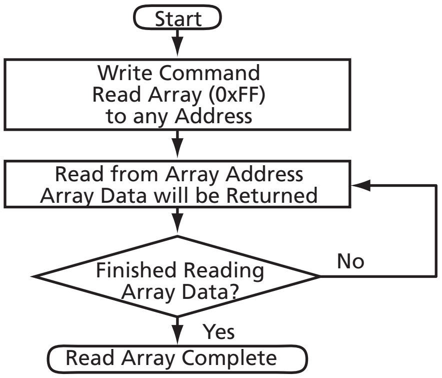


Figure 4 • **Read Array Flow Diagram**

Read Status Command

The algorithm for the Read Status command is shown in Figure 5. The status register shown in Table 5 may be read to determine the success of writing or page erase commands. After writing the Read Status command, all subsequent read operations output data from the status register until another valid command is written. When error conditions cause status register bits S5, S4, or S1 to be set, they can only be reset by the Clear Status command.

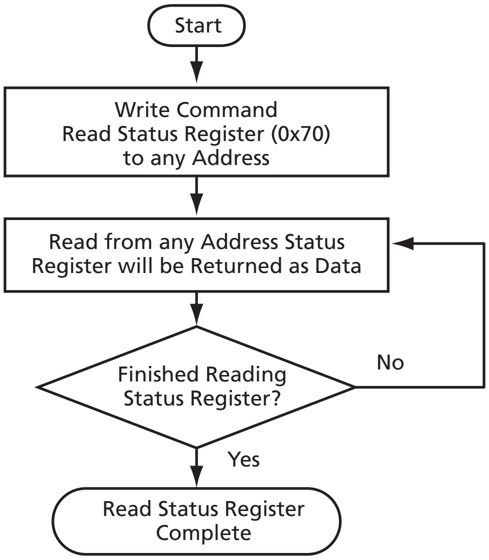


Figure 5 • **Read Status Flow Diagram**

Table 5 • **Status Register**

Status Bit	Description
S7	Ready (real-time)
S6	–
S5	Program/Erase Error (sticky)
S4	Write Error (sticky)
S3-S2	–
S1	Read Error (sticky)
S0	–

Clear Status Command

The algorithm for the Clear Status command is shown in Figure 6. When error conditions cause status register bits S5, S4, or S1 to be set, they can only be reset by the Clear Status command.

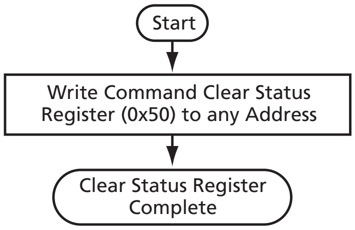


Figure 6 • **Clear Status Flow Diagram**

Erase Page Command

The algorithm for the Erase Page command is shown in Figure 7. The Erase Page requires two bus cycles to start: the command itself and a confirm command. Once the erase starts, it cannot be interrupted (any subsequent commands are ignored while the erase is in progress). CoreAhbNvm handles the required sequences. The user can determine when the erase is complete by monitoring status bit S7 until ready status is indicated (note that the status is updated automatically and the Read Status command sequence is not required). Once the Erase Page command has completed, status bits S1, S4, and S5 should be checked to determine if any page erase error occurred. If any of the error status bits are set, they can only be cleared by a Clear Status command.

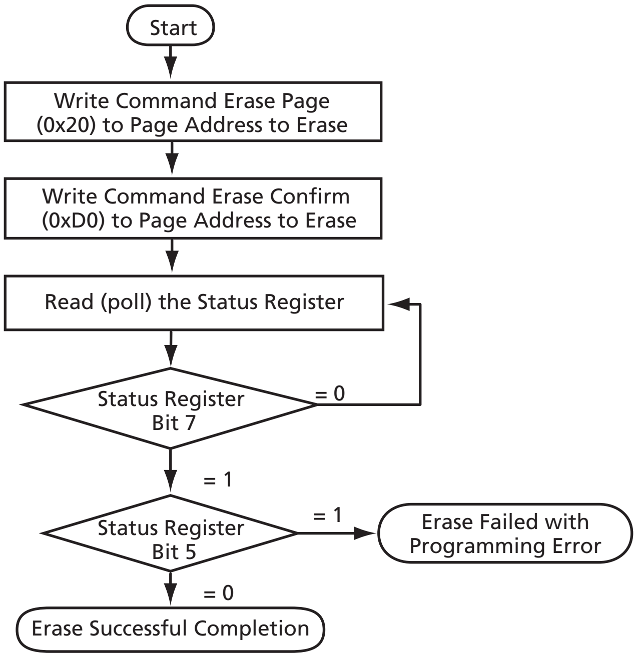

Start
Write Command Erase Page
(0x20) to Page Address to Erase
Write Command Erase Confirm
(0xD0) to Page Address to Erase
Read (poll) the Status Register
Status Register
Bit 7
= 0
= 1
Status Register
Bit 5
= 1
Erase Failed with
Programming Error
= 0
Erase Successful Completion

Figure 7 • **Erase Page Flow Diagram**

Single Write Command

The algorithm for the Single Write command is shown in Figure 8. The Single Write is used to write a single byte, half-word, or word. It should be noted that the Single Write still results in the entire page (in which the Single Write data is contained) being written into memory. Therefore, the user should avoid using Single Writes where Multi-Writes are more appropriate (that is, when more than one location within a page is to be written). A single write is initiated by executing the Single Write command followed by a Write to the desired location (note that all other commands are ignored once the Write is in progress). Once the Write command has completed (that is, the status no longer indicates busy—note that the status is updated automatically and the Read Status command sequence is not required), status bit S4 should be checked to determine if any write error occurred. If any of the error status bits are set, they can only be cleared by a Clear Status command.

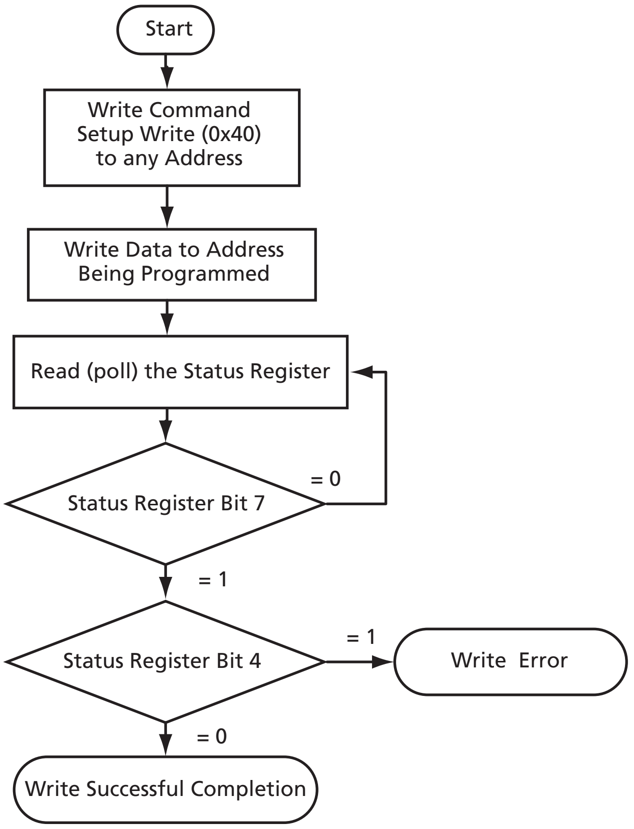


Figure 8 • **Single Write Flow Diagram**

Multi-Write Command

The algorithm for the Multi-Write command is shown in Figure 9 on page 8. The Multi-Write is used to write bytes, half-words, or words. A multi-write is initiated by executing the Multi-Write command and waiting for the write buffer to become available; that is, the status indicates ready. Note that the status is updated automatically and the Read Status command sequence is not required. Once the write buffer is available, the second Write with a data value of N is executed. N is the number of elements (bytes/words/double words) – 1 to be written to the write buffer—the expected ranges are N = 00h to N = 7Fh (e.g., 1 to 128 bytes) in 8-bit mode, N = 00h to N = 003Fh in 16-bit mode, and N = 00h to N = 1Fh in 32-bit mode. Once N is written, the multiple Writes to the desired locations within the page can then be made. Note that once the Multi-Write command has been issued, the page addresses for the subsequent data writes are ignored (this means that Writes will wrap-around onto the same current page if the page address goes outside the page address specified with the first bus cycle). Once the last data value has been written, the confirm command (D0h) is expected after exactly N + 1 write cycles; any other command at that point in the sequence will prevent the transfer of the buffer to the array (the write will be aborted). Note that all other command sequences are ignored once the confirm command is received and the write to the array is started. Once the Multi-Write command has completed (that is, the status no longer indicates busy – note that the status is updated automatically and the Read Status command sequence is not required), status bit S4 should be checked to determine if any write error occurred. If any of the error status bits are set, they can only be cleared by a Clear Status command.

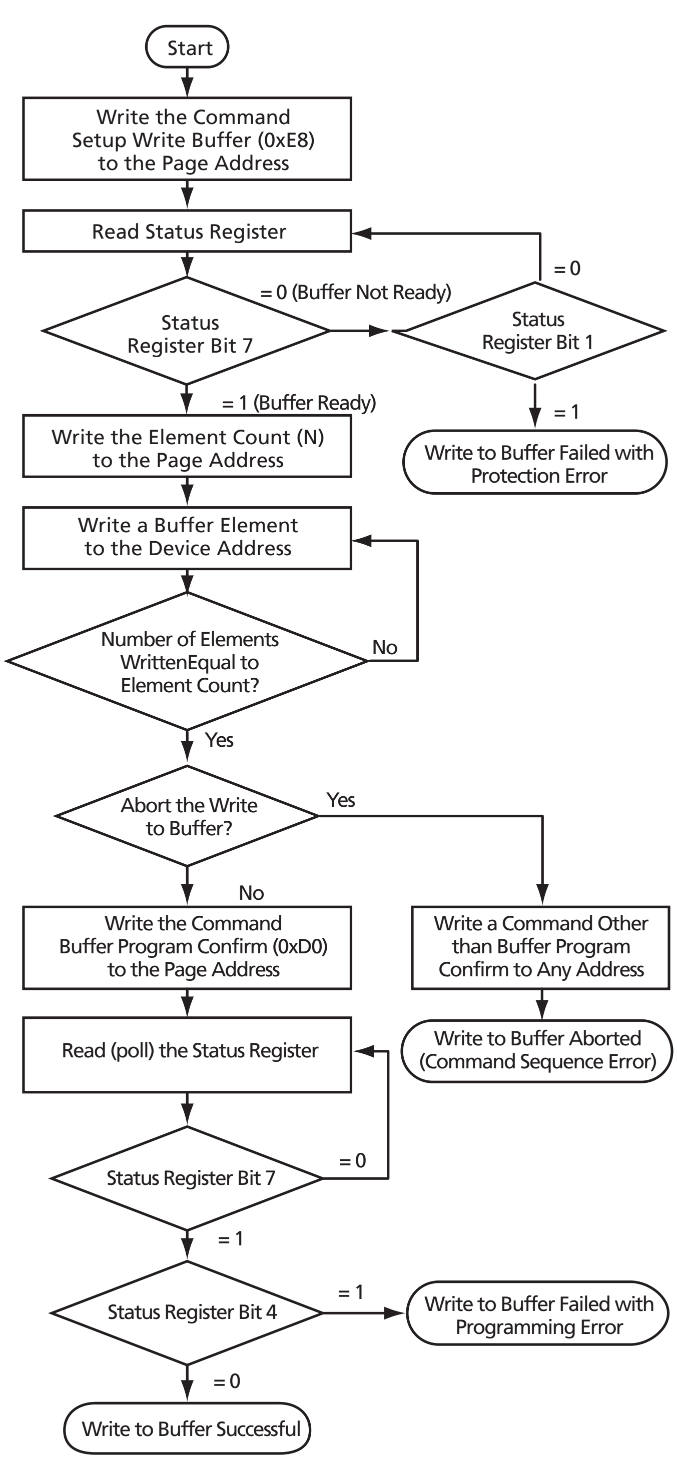

Start
Write the Command Setup Write Buffer (0xE8) to the Page Address
Read Status Register
Status Register Bit 7
= 0 (Buffer Not Ready)
Status Register Bit 1
= 0
= 1 (Buffer Ready)
= 1
Write the Element Count (N) to the Page Address
Write to Buffer Failed with Protection Error
Write a Buffer Element to the Device Address
Number of Elements WrittenEqual to Element Count?
No
Yes
Abort the Write to Buffer?
Yes
No
Write the Command Buffer Program Confirm (0xD0) to the Page Address
Write a Command Other than Buffer Program Confirm to Any Address
Read (poll) the Status Register
Write to Buffer Aborted (Command Sequence Error)
Status Register Bit 7
= 0
= 1
Status Register Bit 4
= 1
Write to Buffer Failed with Programming Error
= 0
Write to Buffer Successful

Figure 9 • **Multi-Write Flow Diagram**

Timing Diagrams

The timing diagrams for CoreAhbNvm are the normal AHB Read and Write timing diagrams available in the AHB specification from ARM.®

Ordering Information

CoreAhbNvm is included in the SysBASIC core bundle that is supplied with the Actel CoreConsole IP Deployment Platform tool. The obfuscated RTL version of SysBASIC (SysBASIC-OC) is available for free with CoreConsole. The source RTL version of SysBASIC (SysBASIC-RM) can be ordered through your local Actel sales representative. CoreAhbNvm cannot be ordered separately from the SysBASIC core bundle.

List of Changes

The following table lists critical changes that were made in the current version of the document.

Previous Version	Changes in Current Version (Advanced v0.2)	Page
Advanced v0.1 (April 2006)	The "Key Features" section was updated to include NVM Spare Page Access.	1
	The "Supported Families" section was updated to remove ProASIC®3 (M7A3P) and ProASIC3E (M7A3PE), which are not supported.	1
	Table 1 · CoreAhbNvm Device Utilization and Performance was updated.	1
	The title of Table 2 · CoreAhbNvm Signal Descriptions, was changed from "CoreAhbNvm Port Macro Signal Descriptions." The descriptions for the HADDR and HSIZE signals were revised. The description of the HRDATA signal was revised to change "date" to "data." The HTRANS signal was added.	2
	The "Maximizing Performance for Large Memories" section is new.	3
	The "Accessing Spare Pages" section is new.	5
	In the "Read Status Command" section, status register bit S3 was removed from the list of status register bits that can be set by error conditions, and bit S1 was added.	6
	In the "Clear Status Command" section, status register bit S3 was removed from the list of status register bits that can be set by error conditions, and S1 was added.	6
	Figure 7 · Erase Page Flow Diagram was updated.	6
	In the "Single Write Command" section, the text was revised to state that status bit S4 should be checked to determine if any write error occurred. Previously the text stated that status bits S4 and S5 should be checked.	7

Datasheet Categories

In order to provide the latest information to designers, some datasheets are published before data has been fully characterized. Datasheets are designated as "Product Brief," "Advanced," and "Production." The definitions of these categories are as follows:

Product Brief

The Product brief is a summarized version of an advanced or production datasheet containing general product information. This brief summarizes specific device and family information for unreleased products.

Advanced

This datasheet version contains initial estimated information based on simulation, other products, devices, or speed grades. This information can be used as estimates, but not for production.

Unmarked (production)

This datasheet version contains information that is considered to be final.

www.actel.com

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655
USA

Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.

River Court, Meadows Business Park
Station Approach, Blackwater
Camberley Surrey GU17 9AB
United Kingdom

Phone +44 (0) 1276 609 300
Fax +44 (0) 1276 607 540

Actel Japan

EXOS Ebisu Building 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan

Phone +81.03.3445.7671
Fax +81.03.3445.7668
http://jp.actel.com

Actel Hong Kong

Room 2107, China Resources Building
26 Harbour Road
Wanchai, Hong Kong

Phone +852 2185 6460
Fax +852 2185 6488
www.actel.com.cn