One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

Flash Programming via MDIO—Protocol Type 8

by Eckart Hartmann

INTRODUCTION

This application note describes a protocol proposed for fully integrated single package parts that incorporate high performance analog peripherals together with digital peripherals controlled by an ARM® Cortex™-M3 processor and integral flash for code and data.

The parts must include an MDIO interface capable of operating at up to 4 MHz and the capability to simultaneously execute from one flash block and to write/erase the other flash block making it ideal for 40 G to 100 G optical applications.

The nonerasable kernel code plus flags in user flash provide assistance to allow user code to robustly switch between the two blocks of user flash code and data spaces as required for MDIO.

For further details regarding the flags and other part specific details, consult the specific part hardware reference manual.

The purpose of this application note is to specify a flash procedure for initial programming of the user flash memory with only the power, GND, MDIO, and MDC pins accessible. The MDIO pin is in push-pull mode and does not require an external pull-up. The programming sequence can be initiated and controlled by a host.

For details about the MDIO protocol, refer to the MDIO specification IEEE 802.3 and the specific Analog Devices, Inc., device hardware reference manual. Protocol Type 8 only implements the minimum frame types necessary to perform downloads and relies entirely on the verify mechanism described in this application note to check that the download was successful. For in service reprogramming, one can devise and implement their own reprogramming strategy.

TABLE OF CONTENTS

REVISION HISTORY

5/14—Revision 0: Initial Version

ENTERING DOWNLOAD

The device kernel enters download mode if P2.3 is low and both K2B0 and K2B1 are not zero. This is the case when the user code is blank or if the user has deliberately not written these keys as described in the hardware reference manual. This application note concentrates on describing the download protocol once the kernel has entered its downloader.

GENERAL REQUIREMENTS

MDIO Frame Structure

As per the MDIO specification, each frame consists of a preamble, ST, OP, PHYADR, DEVADD, TA, and 16 bits of data.

Of the access types defined by OP, only Address, Write, and Read are used.

The value of PHYADR is 5.

The value of DEVADD is 1.

Commands in Address Frames

Address frames use pseudo addresses that contain command information and partial addresses, if appropriate. The top nibble of the address is interpreted as commands as shown in Table 1.

Table 1. Commands in Address Frames

Command	Description
0	Reserved.
1 Download	Request download. Bit 0 to Bit 11 must be the last 3 nibbles of chip information.
2 SetAddress	Set flash address page to Bit 0 to Bit 11 of frame data.
3 PageErase	Set flash address page to Bit 0 to Bit 11 of frame data and then erase that page.
4 MassErase	Set flash address page to Bit 0 to Bit 11 of frame data and then erase entire user flash space.
5 Verify	Set flash address page to Bit 0 to Bit 11 of frame data and then verify that page.
6	Reserved.
7 Reset	Reset.
0x8 to 0xF	Reserved.

After each command is received, it is executed and progress can be monitored with Read Frames.

Command 1 is required to enter download mode. If the chip information is incorrect, a subsequent read returns 0x0000. If any Erase, Write, or ReadInc frame is received before a correct Request Download, the part locks up until reset.

Write Frames

Write frames contain the data to be programmed to the part. Programming starts at the beginning of the flash page set by the last Command 2, Command 3 or Command 4. After each Write Frame, the data is saved. After every four Write Frames, the collected data is programmed into flash at the correct flash address. Thus, programming must be in multiples of four frames. A read back after a write frame returns the number of bytes processed for this page. This number is only updated after the frame is processed and is divisible by 8 when a location has been flashed.

If user flash protection is active (0x1FFF4 = 0x3A or 0x3FFF4 = 0x3A) writing fails and the read frame returns 0x8BAD.

Programing should always be in multiples of full pages.

Read Frames

After every frame, a reply is prepared to be returned by the next Read Frame (see Table 2).

Table 2. Reply Description

Response to	Reply Description
Command 0	0x0BAD
Command 1	Chip information. See Table 3 for details.
Command 2	0x0002
Command 3	0 if erase not complete; 0x0003 if erase complete. 0x3BAD if error.
Command 4	0 if erase not complete; 0x0004 if erase complete.
Command 5	Verify data. See the description following Table 3.
Command 6	0x6BAD
Command 7	Undefined after reset.
Command 8 to Command 0xF	0x*BAD where * is replaced by the command number.
Write Frame	Refer to the Write Frames section.
Read Frame	Refer to the description following Table 3.

The chip information consists of three nibbles as shown in Table 3.

Table 3. Chip Information

Nibble	Chip Information Description	Defined Values
3	Unused	0
2	Category	0x3 for ADuCM
1	Family	0x2 for 32x
0	Family member	0x0 for 320

Additional reads after the chip ID may contain further chip information. This information is implementation dependent. The first read back after Command 5 returns the sum of the last 4 half words of the page being verified. The second read back returns the low 16 bits of the CRC generated by the flash sign command.

The third read back returns the high 16 bits of the CRC generated by the flash sign command. Together, the three frames check all bytes of the page.

TYPICAL DOWNLOAD SEQUENCE

Table 4 shows a typical download sequence.

Table 4. Typical Download Sequence

Step	Frame Type	Description	Frame Data	Comment
0	Address	Download	0x1320	
1	Read	Chip information	0x0320	Check correct chip is accessed.
2	Address	Erase	0x3000 + page number	(Pages need not be consecutive.)
3	Read	Erase done?	0x0000	Repeat until 0x0003.
4	If desired, repeat Step 2 to Step 3.			
5	Address	Set address	0x2000 + page number	(Pages need not be consecutive.)
6	Write	Data	Next 2 bytes to program	Increment flash address by 2.
7	Write	Data	Next 2 bytes to program	Increment flash address by 2.
8	Write	Data	Next 2 bytes to program	Increment flash address by 2.
9	Write	Data	Next 2 bytes to program	Increment flash address by 2.
10	Read	Flashing done?	Byte count	Repeat until divisible by 8.
11	Repeat Step 5 to Step 10 another 255 times to complete flash page.			
12	If desired, do a page verify as described in Step 14 to Step 18.			
13	Repeat Step 5 to Step 12 until all pages are downloaded. Include Step 4 if next page not previously erased.			
14	Address	Verify	0x5000 + page number	(Pages need not be consecutive.)
15	Read	Verify	4 × 16-bit sum	16 bit sum of last 4 half words.
16	Read	Verify	CRC LSBs	Lower 16 bits of FEESIG.
17	Read	Verify	CRC MSBs	Upper 16 bits of FEESIG.
18	Check that verify values equal expected verify values.			
19	Repeat Step 14 to Step 18 for all pages as final check of correct download. (Pages need not be consecutive.)			
20	Address	Reset	0x7000	Software reset restarts part.

The host can terminate a download with Command 7 if, for instance, it takes too long to get a suitable read back value or a bad reply is received.

REFERENCES

IEEE 802.3 standard, Clause 45, is described online.