Application Report
SPRABA1B–January 2011

TMS320DM355/DM335 Migration Guide (silicon revision 1.1, 1.3 and 1.4)

Adam Barnett — DSP Software Applications

ABSTRACT

This application report describes the differences between Silicon revision 1.1, 1.3 and 1.4 of the TMS320DM355/DM335 digital media system-on-chip (DMSoC). This document discusses behavior different from that described in the *TMS320DM355 Digital Media System-on-Chip (DMSoC) ARM Subsystem User's Guide* (SPRUFB3) and the *TMS320DM335 Digital Media System-on-Chip (DMSoC) ARM Subsystem User's Guide* (SPRUFX7).

Contents

List of Figures

List of Tables

1 Overview

1.1 Differences Between Silicon Revision 1.1, 1.3 and 1.4

The only difference between silicon revision 1.1, 1.3 and 1.4 is the replacement of the ROM-boot loader (RBL) with a different version with different functional behavior. There are no other functional changes to the device.

1.2 ROM-Boot Loader (RBL)

The ROM-boot loader (RBL) is firmware that is stored in ROM on the DM355/DM335; it is responsible for handling the boot process. When the boot process is initiated, it senses the state of the BOOTSEL[0:1] pins and, based on that state, loads a user-boot loader (UBL) from external media and branches to the entry point of the UBL.

1.3 ROM-Boot Loader (RBL) Change Summary

This section summarizes the changes in the RBL.

The RBL used on silicon revision 1.3:

- Makes several changes to the NAND boot mode support
- Makes several changes to its support for UART boot

The RBL used on silicon revision 1.4:

- Makes several changes to the NAND boot mode support
- Adds SPI boot as a fall back for NAND boot

1.3.1 Changes to the NAND Boot Functionality

Silicon revision 1.3 changes the following changes to NAND boot functionality:

- A bug in the ECC error correction has been fixed.
- The assumed layout of the data in the boot NAND has changed.
- The NAND device ID table has been extended.
- Support for 4K and 8K page size devices has been added.
- An attempt to decode the characteristics of NAND devices not contained in the NAND device ID table will be attempted.

Silicon revision 1.4 adds the following changes to NAND boot functionality:

- The following has been added to overcome the 4th ID byte variations:
  - ONFI support
  - Reading of NAND parameters from SPI EEPROM
  - Samsung 4th ID support
- A bug in the 4-bit ECC error correction has been fixed.
- The NAND device ID table has been extended.
- Support for 16K page size devices has been added.

Due to these changes, the fallback MMC/SD bootmode will be delayed in silicon revision 1.4. The standalone MMC/SD bootmode mode is not impacted because of these NAND changes.

1.3.2 Changes to the UART Boot Functionality

Silicon revision 1.3 and later includes the following changes to UART boot functionality:

- Only 7 characters of the *ACK* sequence are read instead of 8. The trailing *\0* is not used.
- An additional UART boot option has been added that shortens the boot sequence.

1.3.3 SPI Boot Functionality

Silicon revision 1.4 adds SPI boot functionality while running NAND boot. If an EEPROM exists and has a relevant SPI boot mode UBL header descriptor, the UBL is downloaded from EEPROM and the UBL code is executed. Otherwise the NAND boot continues executing as earlier.

2 NAND Boot Changes

This section discusses the differences between the ROM-boot loader (RBL) support for NAND boot on silicon revision 1.1 and 1.3 and changes between silicon revision 1.3 and 1.4. The changes between silicon revision 1.1 and 1.3 are summarized as follows:

- Silicon revision 1.3 expands the number of NAND devices supported compared to silicon revision 1.1.
- An RBL error correction bug has been fixed.
- The assumed data layout on the NAND device has changed.
- The Magic number functions differently

The changes between silicon revision 1.3 and 1.4 are summarized as follows:

- Silicon revision 1.4 expands the number of NAND devices supported compared to silicon revision 1.3.
- An RBL 4-bit error correction bug has been fixed.
- New methods to detect/read 4th byte information has been added.

2.1 *Additional Devices Supported*

The ROM-boot loader (RBL) used in silicon version 1.3 has added support for additional NAND devices.

2.1.1 Changes in How the Magic Number Works

The ROM-boot loader (RBL) used in silicon version 1.1, 1.3 and 1.4 uses magic numbers to identify NAND layout and boot options. A magic number of the form 0xA1AC EDxx was used in revision 1.1. Using a magic number in this form for revision 1.3 and 1.4 places the DM355/DM335 in compatibility mode. This forces the device to operate exactly as if it were silicon revision 1.1. You cannot use the larger NAND devices when operating in compatibility mode.

Using a magic number of the form 0xA1BC EDxx places the DM355/DM335 in standard mode. Standard mode for revision 1.3 and 1.4 allows for the use of the larger NAND devices explained in the next section.

2.1.2 Support for 4K and 8K NAND Devices Added

The ROM-boot loader (RBL) used in silicon revision 1.1 supported NAND page sizes of 512 bytes and 2048 bytes.

The RBL used in silicon revision 1.3 supports page sizes of 512 bytes, 2048 bytes, 4096 bytes, and 8192 bytes

NOTE: When this document was prepared, no 8K devices were available for testing, so 8K has not been tested. However, the code contains support for these devices.

2.1.3 Support for 16K NAND Devices Added

The RBL used in silicon revision 1.4 supports page sizes of 512 bytes, 2048 bytes, 4096 bytes, 8192 bytes and 16384 bytes.

NOTE: When this document was prepared, no 8K and 16K devices were available for testing, so 8K and 16K has not been tested. However, the code contains support for these devices.

2.1.4 NAND Device ID Table Updated

The RBL contains an internal table containing a list of known NAND devices. This list has been updated on Silicon Revision 1.3 and Silicon Revision 1.4. Table 1 and Table 2 show the devices contained in the tables. If the NAND device is not found in this table, then the RBL will read the fourth byte of the NAND ID table and attempt to decode this to obtain the necessary parameters.

For the purpose of determining NAND block size and page size the information from the fourth byte is considered as follows:

- Bits 5 and 4 determine the block size
  - Bits 5,4 = 00: 64KB
  - Bits 5,4 = 01: 128KB
  - Bits 5,4 = 10: 256KB
  - Bits 5,4 = 11: 512 KB

- Bits 1 and 0 determine the page size
  - Bits 1,0 = 00: 1KB
  - Bits 1,0 = 01: 2KB
  - Bits 1,0 = 10: 4KB
  - Bits 1,0 = 11: 8KB

In silicon revision 1.4, the latest Samsung (manufacturer ID: 0xEC) 4th ID definition has been added which is as follows:

- Bits 5 and 4 determine the block size
  - Bits 5,4 = 00: 128KB
  - Bits 5,4 = 01: 256KB
  - Bits 5,4 = 10: 512KB
  - Bits 5,4 = 11: 1024 KB
- Bits 1 and 0 determine the page size
  - Bits 1,0 = 00: 2KB
  - Bits 1,0 = 01: 4KB
  - Bits 1,0 = 10: 8KB
  - Bits 1,0 = 11: reserved

Table 1. NAND Devices in NAND Device ID Table in Silicon Revision 1.1

Device ID	Pages per Block	Bytes pr Page	Block Shift Value for Address	Number of Address Cycles
0xE3	16	512+16	12	3
0xE5	16	512+16	12	3
0xE6	16	512+16	12	3
0x39	16	512+16	13	3
0x6B	16	512+16	13	3
0x73	32	512+16	13	3
0x33	32	512+16	13	3
0x75	32	512+16	13	3
0x35	32	512+16	13	3
0x43	32	512+16	13	4
0x45	32	512+16	13	4
0x53	32	512+16	13	4
0x55	32	512+16	13	4
0x76	32	512+16	13	4
0x36	32	512+16	13	4
0x79	32	512+16	13	4
0x71	32	512+16	13	4
0x46	32	512+16	13	4
0x56	32	512+16	13	4
0x74	32	512+16	13	4
0xF1	64	2048+64	22	4
0xA1	64	2048+64	22	4
0xAA	64	2048+64	22	5
0xDA	64	2048+64	22	5
0xAC	64	2048+64	22	5
0xDC	64	2048+64	22	5
0xB1	64	2048+64	22	5
0xC1	64	2048+64	22	5

Table 2. NAND Devices in NAND Device ID Table in Silicon Revision 1.3 and 1.4

Device ID	Pages per Block	Bytes pr Page	Block Shift Value for Address	Number of Address Cycles
0xE3	16	512+16	12	3
0xE5	16	512+16	12	3
0xE6	16	512+16	12	3
0x6B	16	512+16	13	3
0x73	32	512+16	13	3
0x33	32	512+16	13	3
0x75	32	512+16	13	3
0x35	32	512+16	13	3
0x43	32	512+16	13	4
0x45	32	512+16	13	4
0x53	32	512+16	13	4
0x55	32	512+16	13	4
0x76	32	512+16	13	4
0x36	32	512+16	13	4
0x79	32	512+16	13	4
0x71	32	512+16	13	4
0x46	32	512+16	13	4
0x56	32	512+16	13	4
0x74	32	512+16	13	4
0xF1	64	2048+64	22	4
0xA1	64	2048+64	22	4
0xAA	64	2048+64	22	5
0xDA	64	2048+64	22	5
0xAC	64	2048+64	22	5
0xDC	64	2048+64	22	5
0xB1	64	2048+64	22	5
0xC1	64	2048+64	22	5

2.2 ECC Bug Fixed

The ROM-boot loader (RBL) used in this silicon revision contained a bug that prevented the ECC hardware from detecting and correcting bit stream errors while the UBL was read from the NAND device. This bug was fixed in silicon revision 1.3.

The RBL in silicon revision 1.3 has a 4-bit ECC mode limitation. If more than 4 errors are detected, the ECC error bit is not reset, leading to failure to correct subsequent errors. This bug was fixed in silicon revision 1.4.

2.3 NAND Layout Changed

The position of data on the NAND device assumed by the ROM-boot loader (RBL) has changed.

2.3.1 NAND Layout Used by Silicon Revision 1.1

Silicon Revision 1.1 assumed the following data layout shown in Table 3:

Table 3. NAND Layout Used by Silicon Revision 1.1

512 Byte Page size	2048 Byte Page Size
512 bytes Data	512 bytes Data
16 bytes ECC Data	16 bytes ECC Data
	512 bytes Data
	16 bytes ECC Data
	512 bytes Data
	16 bytes ECC Data
	512 bytes Data
	16 bytes ECC Data

2.3.2 NAND Layout Used by Silicon Revision 1.3 and Silicon Revision 1.4

Silicon revision 1.3 and 1.4 assumes that all the data on a NAND page is in a single block with the ECC and bad block data contained in the spare byte areas as shown in Table 4.

Table 4. NAND Layout Used by Silicon Revision 1.3 and 1.4

512 Byte Page size	2048 Byte Page Size	4096 Byte Page Size	8192 Byte Page Size
512 bytes data	2048 bytes data	4096 bytes data	8192 bytes data
16 bytes ECC data	64 bytes ECC data	128 bytes ECC data	256 bytes ECC data

2.4 Additional Methods to Detect NAND Parameters in Silicon Revision 1.4

The silicon revision 1.4 adds new methods to detect NAND parameters apart from the existing 4th ID byte information and device table. These are:

- ONFI support
- Reading of NAND parameters from SPI EEPROM

2.4.1 ONFI support

ONFI format is used by many NAND manufacturers including Micron, Numonyx and Hynix. It provides a standard parameter format as compared to the varying 4th ID byte across vendors and NAND generations. If a NAND is detected to be ONFI compatible, the information from the ONFI parameter page is used to interface with NAND.

2.4.2 NAND Parameters From SPI EEPROM

To provide more flexibility to obtain the correct NAND parameters, the NAND parameters can be read from a user written EEPROM over SPI. The format of the NAND parameters in EEPROM (as expected by RBL) is defined in Table 5.

Table 5. NAND Parameters Format in EEPROM

Offset (in bytes) From Base [1]	Parameter
8	Magic number (0xA1ACEDAA)
12	Page count (Number of pages per block)
16	Number of address cycles
20	Page size (Number of bytes in a page)
24	Spare size (Number of spare bytes in a page)
28	Block shift (Number of bits by which block address is to be shifted)

[1] Note that the first 8 bytes might be used for MAC address and is not used by RBL/UBL.

The NAND-boot mode code flow in silicon revision 1.4 is illustrated in Figure 1.

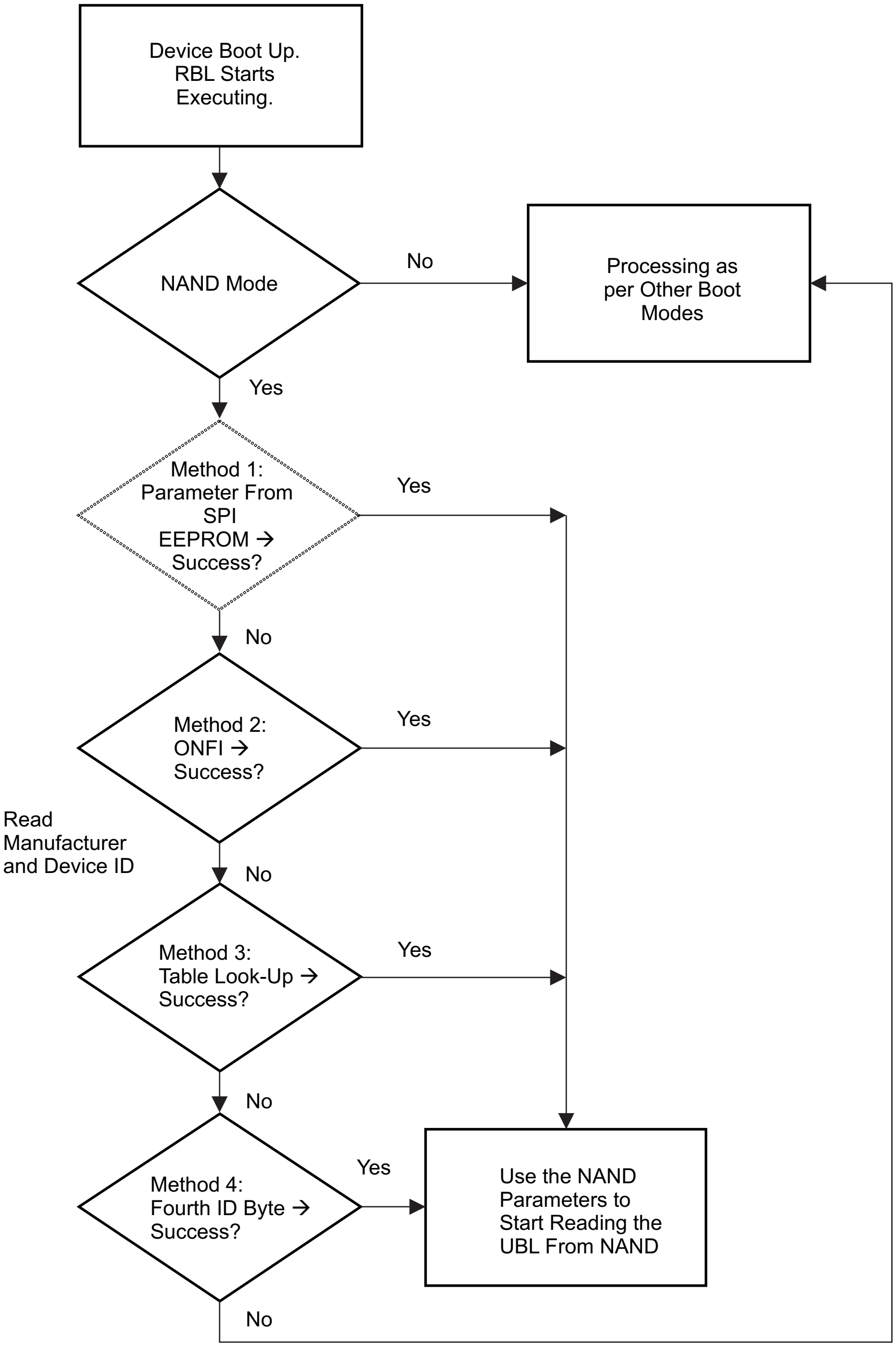

Figure 1. NAND Boot Process in Silicon Revision 1.4

3 UART-Boot Mode Changes

This section details the two functional changes made to UART-boot mode.

- The number of characters tested by the DM355/DM335 when the Host sends the *ACK* string has changed from 8 to 7
- An optional change in the boot sequence has been added

3.1 UART Boot Functionality in Silicon Revision 1.1

The boot sequence used in silicon revision 1.1 is given below:

1. RBL repeatedly sends a *BOOTME* string until it receives a correct response from the host.
2. Host sends:
   (a) *ACK\0*
   (b) UBL Checksum
   (c) UBL size in bytes
   (d) UBL physical start address
   (e) *0000*
3. DM355/DM335 sends *BEGIN.*
4. Host sends the CRC-32 lookup table.
5. DM355/DM335 verifies the checksum for the lookup table and sends *DONE* if the checksum is correct. If the checksum is bad, DM355/DM335 sends *CORRUPT* and branches back to Step 1).
6. Host sends the UBL.
7. DM355/DM335 verifies the checksum for the UBL and sends *DONE* if the checksum is correct. If the checksum is bad, the DM355/DM335 sends *CORRUPT* and branches back to Step 1).
8. DM355/DM335 branches to the UBL start address.

3.2 UART Boot Functionality in Silicon Revision 1.3 and 1.4

The boot sequence used in silicon revision 1.3 and 1.4 has changed in two respects compared to silicon revision 1.1:

- The RBL tested for a terminating *\0* in the *ACK* string send in step 2a. The RBL only tests for 7 characters in silicon revision 1.3; so the terminating *\0* is ignored.
- If the *0000* string in step 2e is replaced by a *0001* string; then the DM355/DM335 will not check the checksum for the CRC-32 lookup table until the UBL has been sent. After the CRC-32 lookup table and the UBL have been sent, the DM355/DM335 will check both and send two *DONE* strings. The flow in this case is:

1. RBL repeatedly sends a *BOOTME* string until it receives a correct response from the host.
2. Host sends:
   (a) *ACK\0*
   (b) UBL Checksum
   (c) UBL size in bytes
   (d) UBL physical start address
   (e) *0001*
3. Host sends the CRC-32 lookup table.
4. Host sends the UBL.
5. DM355/DM335 verifies the checksum for the lookup table and sends *DONE* if the checksum is correct. If the checksum is bad, DM355/DM335 sends *CORRUPT* and branches back to Step 1).
6. DM355/DM335 verifies the checksum for the UBL and sends *DONE* if the checksum is correct. If the checksum is bad, the DM355/DM335 sends *CORRUPT* and branches back to Step 1).
7. DM355/DM335 branches to the UBL start address.

4 SPI-Boot Mode in Silicon Revision 1.4

To provide serial boot support, SPI boot support has been added. Due to the limitation of boot pins, the SPI-boot mode has been added as a part of NAND boot. If an EEPROM is detected with a pre-defined magic pattern during NAND boot, it indicates SPI boot. In this scenario, SPI boot is performed. The header format for SPI boot is defined in Table 6.

Table 6. SPI Descriptor/Header Format

Offset (in bytes) From Base [1]	Parameter
8	Magic number (0xA1ACED00)
12	UBL Entry Point
16	UBL Size in bytes
20	Prescaler
24	UBL start address
28	UBL load address

[1] Note that the first 8 bytes might be used for MAC address and is not used by RBL/UBL.

After the header is read, the UBL code is copied as per the details from the UBL header/descriptor and the control jumps to the starting address as specified in the header.

5 ROM version ID

The ROM ID is stored at address 0x00009FFC and is four bytes long. To identify which revision of the silicon you are running, examine the values at this address and compare with those in Table 7.

Table 7. ROM Version IDs

ROM Version	ID Stored at 0x00009FFC
Silicon Revision 1.1	0x10040101
Silicon Revision 1.3	0x10040103
Silicon Revision 1.4	0x10040104

6 References

- *TMS320DM355 Digital Media System-on-Chip (DMSoC) ARM Subsystem User's Guide* (SPRUFB3)
- *TMS320DM335 Digital Media System-on-Chip (DMSoC) ARM Subsystem User's Guide* (SPRUFX7)

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com	Wireless	www.ti.com/wireless-apps
RF/IF and ZigBee® Solutions	www.ti.com/lprf		
		TI E2E Community Home Page	e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265