

# Pre-programming Guide for NAND Flash memory

*Rev 0.0*  
*Feb. 22. 2003*



***Product Planning & Application Eng. Team***

## ■ The method of Pre-programming

- Gang Programming/In-situ Programming
- Basic Concepts for NAND Formatting

## ■ Gang Programming

- Gang Programming Procedure (Step 1-2-3)
- NAND Formatter on Gang Programmer
- Who does What for Gang Programming ?
- Gang Programmer examples
- ROM Writer Manufacturers

## ■ In-situ Programming

- In-situ Programming Procedure (Step 1-2-3)
- In-situ Programming examples (Multi-ICE, JTAG)

# The Methods of Pre-programming

Confidential

## ■ Gang Programming

- For large-scale production using 3rd party Gang-programmer
- Write master image file to NAND with formatter code running on ROM Writer

## ■ In-situ Programming

- For developers and may be adopted in low volume production
- Write image data to NAND using Multi-ICE, Trace32 or JTAG connected PC
- Multi-ICE/Trace32 controls MCU on the system and image data is programmed NAND
- Small bootstrap code under 16KB may be factory-preprogrammed and main OS image under FTL can be written using E-Boot

# Basic Concepts for NAND-Formatting

## ■ Taking care of Invalid Blocks

- NAND Flash has invalid blocks when it is shipped to user
- User should skip the Invalid Blocks when user writes data to NAND

## ■ Generation of Mapping Table

- Scan NAND array to identify Invalid Blocks
- To re-locate addresses for Invalid Blocks, it is needed to generate address replacement table

## ■ Add ECC & Logical meta data

- To use ECC function, ECC data should be generated and written to spare array
- Should be written adequate meta data in case of using file system

# Gang Programming Procedure(1)

Confidential

## ■ Step 1 : Preparation of O/S image

- System Developer generates O/S image
- O/S image include Boot code, O/S, Applications, graphic data, etc.

## ■ Step 2 : Conversion to Master image

- Meta data(eg. ECC) added according to physical format of FTL(Flash Transition Layer), which is written to NAND
- Master image is created on PC because Gang-programmer does not have the MIPS for generating on-the-fly ECC

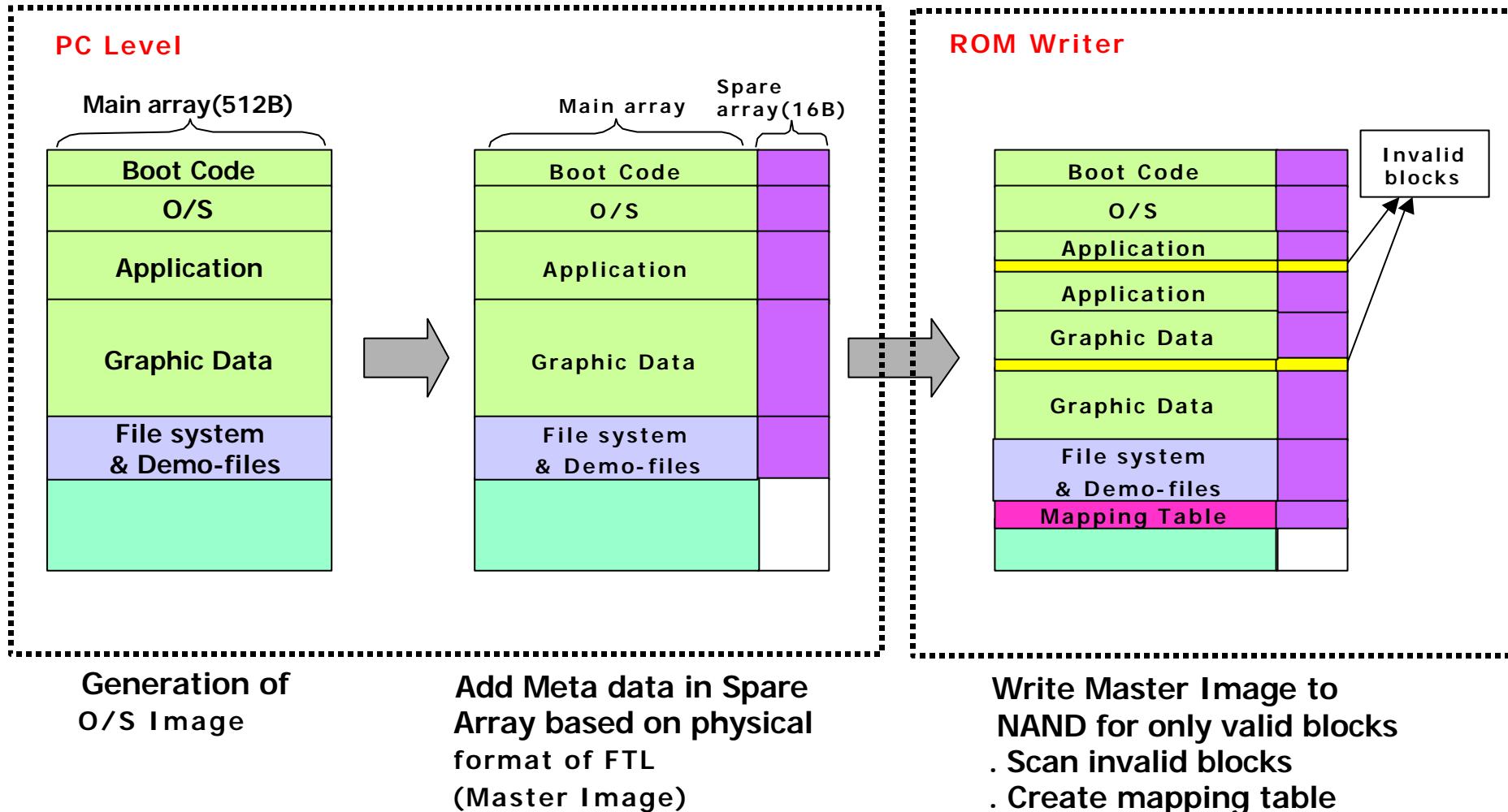
## ■ Step 3 : Writing Master image to NAND

- Scan NAND array to identify invalid blocks
- Create mapping table
- Write master image to NAND while taking care of invalid blocks

# Gang Programming Procedure(2)

Confidential

## Image Conversion and Write Process



# NAND-Formatter on Gang-Programmer

- Gang-programmer should run a formatter, which writes master image into NAND while skipping invalid blocks
- In many cases, demo-files under file-system need to be included as part of pre-programming
- Programming files into file-system requires needs a format-process of file-system storage, according to its designed meta-data format
- The formatter should be prepared by Flash file-system developers and delivered to Gang-programmer manufacturers

# Who Does What for Gang-Programming?

Classification		Contents
System Developer	Software ENG' R	. Prepare Master Image File
	MCU Manufacturer (or FTL Developer)	. Should provide and explain Formatter to ROM Writer manufacturer
ROM Writer Manufacturer		. Generates ROM Writer codes according to Developer's requirements
Samsung		. Coordinate between ROM Writer manufacturer and MCU manufacturer(or FTL developer)

# Gang Programmer Examples

ROM Writer Manufacturer	Device	Implemented algorithm
E.E.Tools Inc.	128M NAND	<ul style="list-style-type: none"> <li>. Programming at only good blocks (Invalid blocks skip)</li> <li>. No use parity area</li> <li>. 1<sup>st</sup> block skip</li> </ul>
	256M NAND	<ul style="list-style-type: none"> <li>. Use Mapping Table(@last block)</li> <li>. Use parity area(for ECC)</li> </ul>
Wavetech	128M NAND	<ul style="list-style-type: none"> <li>. Programming at only good blocks (Invalid blocks skip)</li> <li>. No use parity area</li> <li>. 1<sup>st</sup> block skip</li> </ul>
Data I/O	128M NAND 256M NAND	<ul style="list-style-type: none"> <li>. Programming at only good blocks (Invalid blocks skip)</li> <li>. Using parity area is optional</li> </ul>
System General	256M NAND	<ul style="list-style-type: none"> <li>. Use GBBM algorithm</li> <li>. GBBM is used in Samsung S/W (PocketStore, TFS)</li> </ul>

# ROM Writer Manufacturers

Confidential

Manufacturer	Contact points	Headquarter
E.E. Tools. Inc.	E.E. Tools. Inc. 549 Weddell Drive Sunnyvale, CA 94089 USA Tel : (408) 734-8184 Fax : (408) 734-8185 contact person : John Kim email : johnkim@eetools.com Web : www.eetools.com	USA
DATA I/O Corp.	Data I/O Corporation 10525 Willows Road NE PO Box 97046, Redmond, WA 98073-9746 Tel : (425) 881-6444 Fax : (425) 882-1043	USA
Wavetech	B&M SYSTEM - located in Korea 25-1 Dongsomun Dong 6Ga Sungbuk-gu Seoul Korea Tel : +82-2-927-7575, 922-3620 Fax : +82-2-927-7520 email : bmsys103@yahoo.co.kr cdw0706@korea.com	Japan
System General	System General Taiwan 5F, No. 9, Alley 6, Lane 45 Bao-Shing Road, Shin-Dian City, Taipei Hsien, Taiwan Tel : +886-2-2917-3005 FAX: +886-2-2911-1283	Taiwan

## ■ Step 1 : Preparation of O/S image

- System Developer generates O/S image
- O/S image include Boot code, O/S, Applications, graphic data, etc

## ■ Step 2 : Conversion to Master image

- Meta data(eg. ECC) added according to physical format of FTL(Flash Transition Layer), which is written to NAND
- Can be skipped, if MCU has high level operation  
( In this case, formatter generates Meta data directly)

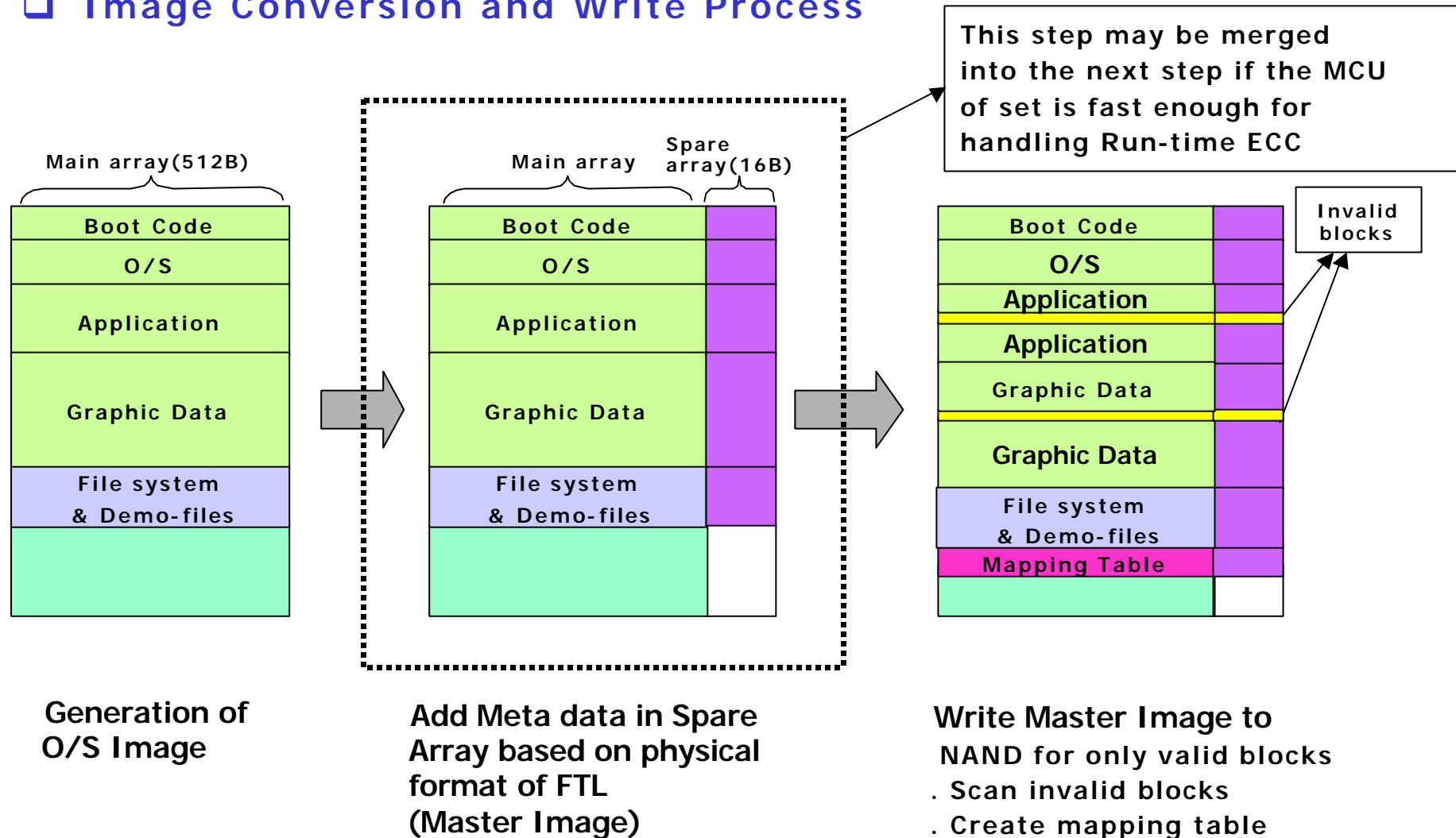
## ■ Step 3 : Writing Master image to NAND

- Use Multi-ICE, Trace32, JTAG, etc
- Scan NAND array to identify invalid blocks and create mapping table
- Write master image to NAND while taking care of invalid blocks

# In-situ Programming(2)

Confidential

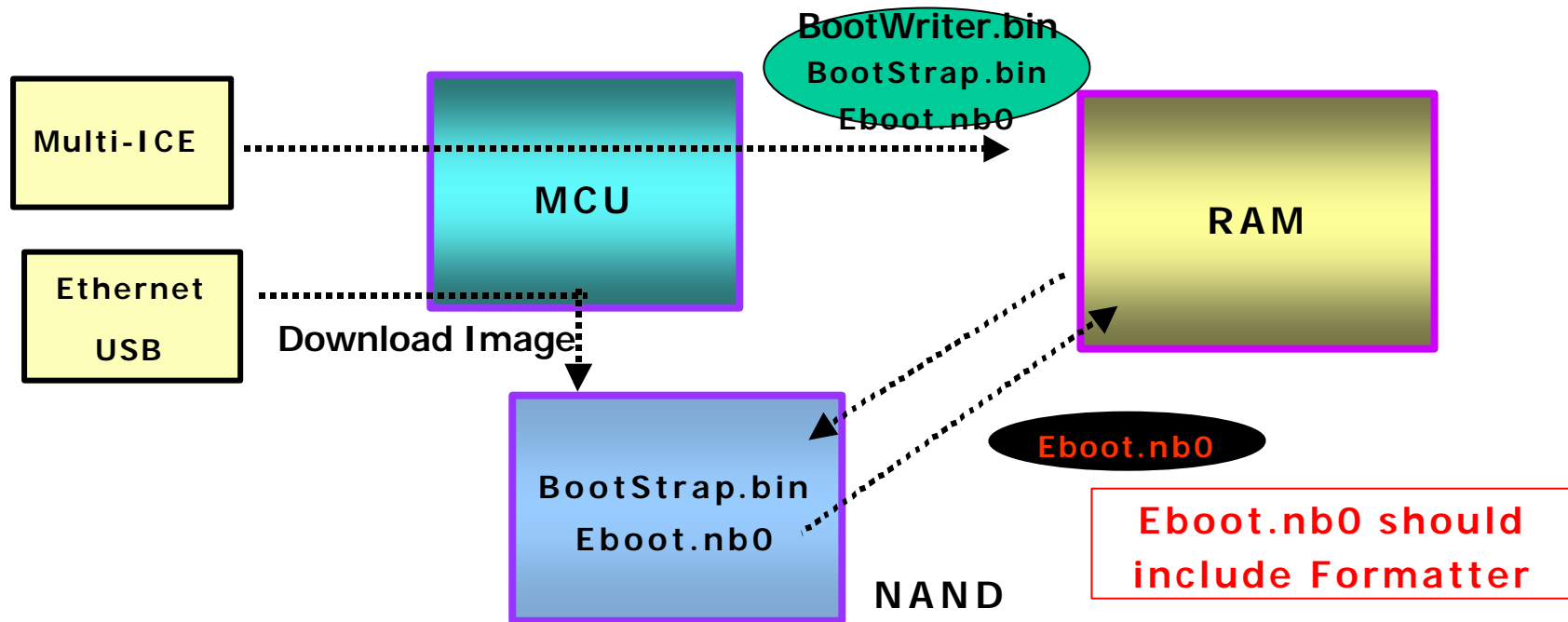
## Image Conversion and Write Process



# In-situ Programming(3) - example 1

Confidential

## Code-Burning into NAND using Multi-ICE



Load 1<sup>st</sup> set of programs into RAM by Multi-ICE

BootWriter loads BootStrap.bin and Eboot.nb0 into NAND

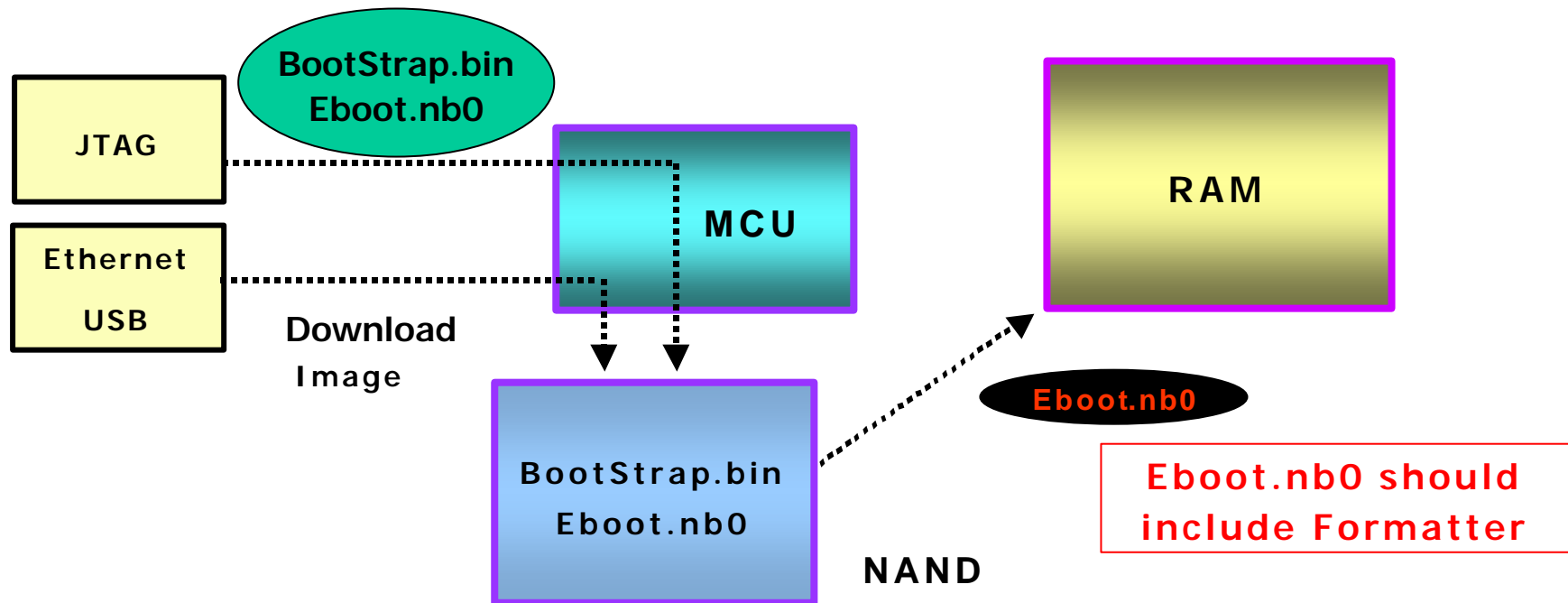
Upon power-on-reset, Eboot is selectively executed with Reset key and Interrupt pin 0

Eboot activates Ethernet and OS image is downloaded into NAND

# In-situ Programming(4) - example 2

Confidential

## Code-Burning into NAND using JTAG



Load 1<sup>st</sup> set of programs into NAND by JTAG

Upon power-on-reset, E-Boot is selectively executed with Reset key and Interrupt pin 0

Eboot activates Ethernet and OS image is downloaded into NAND