System Boot from NAND Flash

# Contact Information

- Contact Information
  - Wesley A. Prouty
  - Micron Technology, Inc.
  - Senior Applications Engineer
  - waprouty@micron.com

System Considerations

- Although generic is some aspects, this presentation covers a typical boot process for a system with the following characteristics:
  - ARM9 or ARM11 processor core with internal ROM code and SRAM to support boot.
  - Processor includes NAND controller hardware.
  - No specific operating system but Linux OS is implied for this presentation.
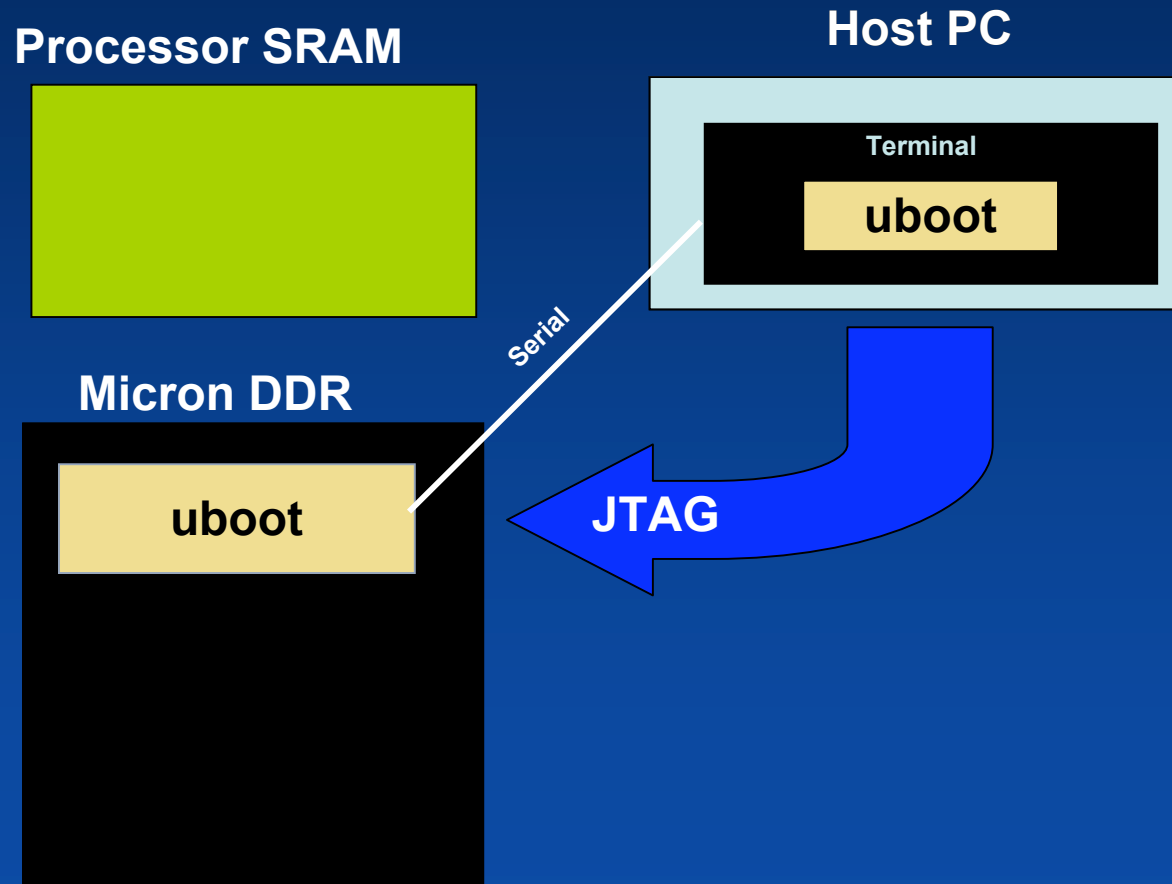
Boot from NAND Steps

Stage 4 Boot (Operating System)

- The stage 3 Boot Loader code is referred to as U-boot in this presentation.

Stage 3 Boot (Boot Loader)

- The stage 2 Boot-strap code is referred to as x-loader in this presentation.

Stage 2 Boot (Boot-strap)
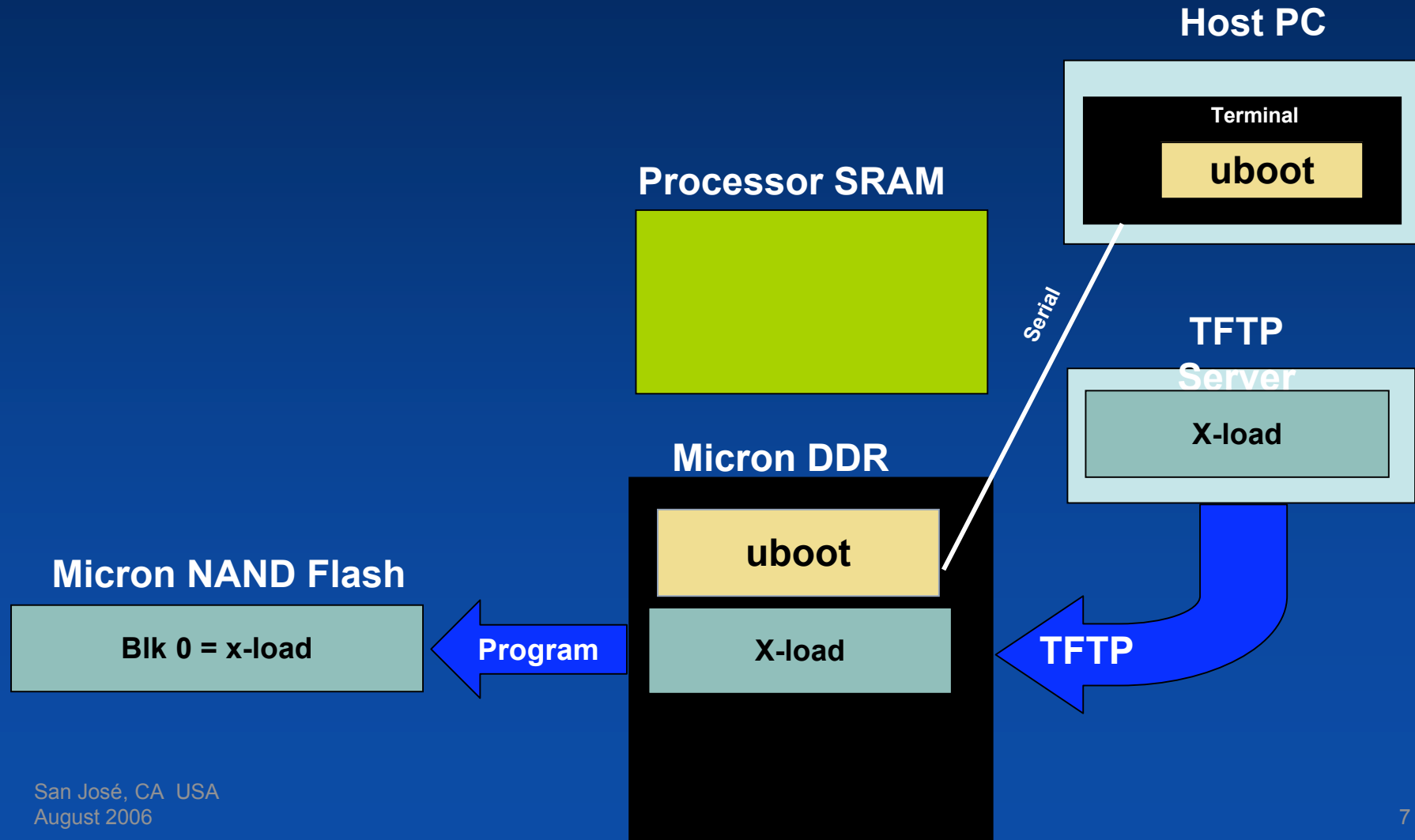
Stage 1 Boot (ROM Code)
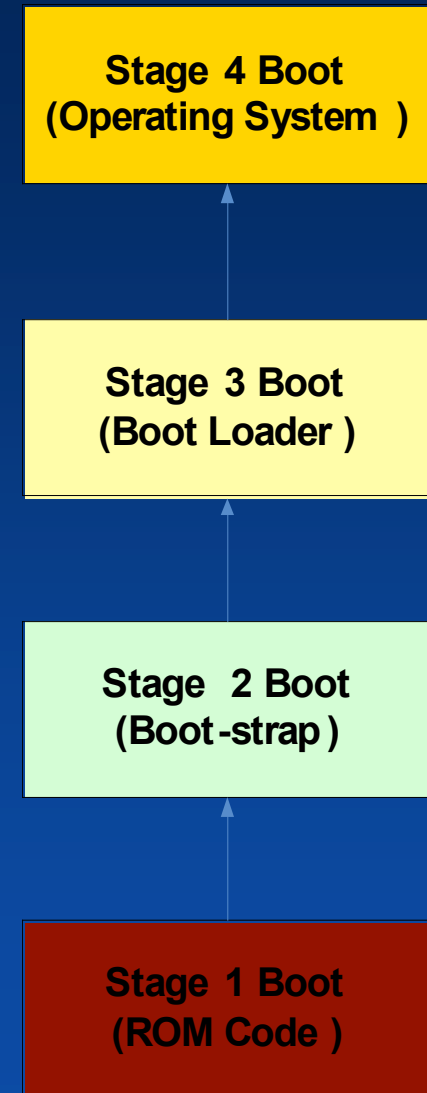
NAND Flash Considerations

- NAND flash is not an eXecute In Place (XIP) memory so when booting from NAND, code must be copied (or shadowed) from NAND to RAM before it can be executed.

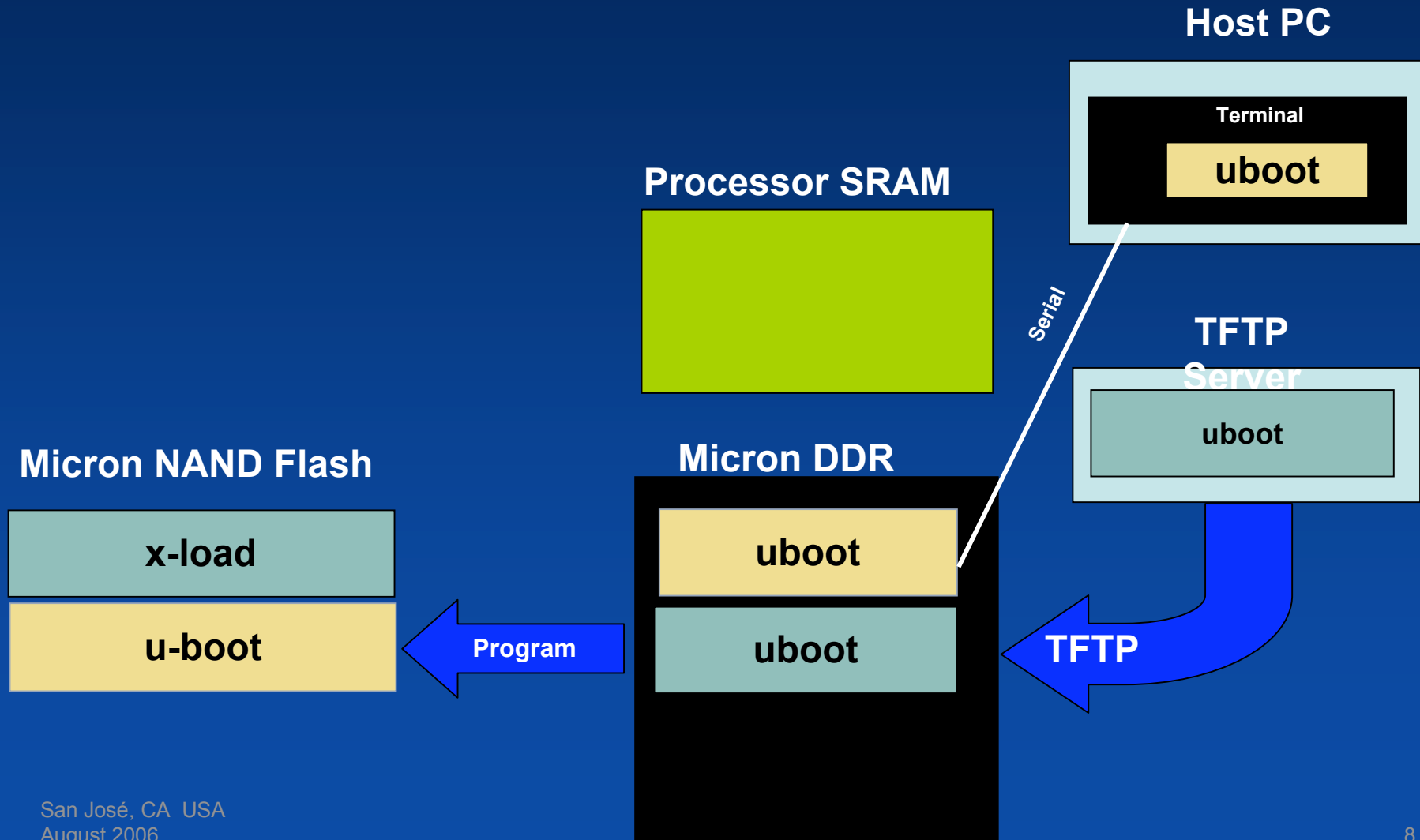- Designers must also consider how the code will be initially programmed to NAND flash.

Writing code to NAND Flash:
Step 1 - Load and run "u-boot" in DRAM

Processor SRAM

Host PC

Terminal

uboot

Serial

Micron DDR

uboot

JTAG

Writing code to NAND Flash:
Step 2 - Write x-load to NAND

Host PC

Terminal

uboot

Processor SRAM

Serial

TFTP Server

X-load

Micron DDR

uboot

Micron NAND Flash

Blk 0 = x-load

Program

X-load

TFTP

Writing code to NAND Flash:
Step 3 - Write u-boot to NAND

Host PC

Terminal

uboot

Processor SRAM

Serial

TFTP Server

uboot

Micron NAND Flash

x-load

u-boot

Micron DDR

uboot

uboot

Program

TFTP

Writing code to NAND Flash:
Step 4 - Write OS kernel to NAND

Host PC

Terminal

uboot

Processor SRAM

Serial

Micron NAND Flash

| x-load |
| u-boot |
| uboot environment data |
| Kernel |

TFTP Server

Kernel

Micron DDR

uboot

kernel

Program

TFTP
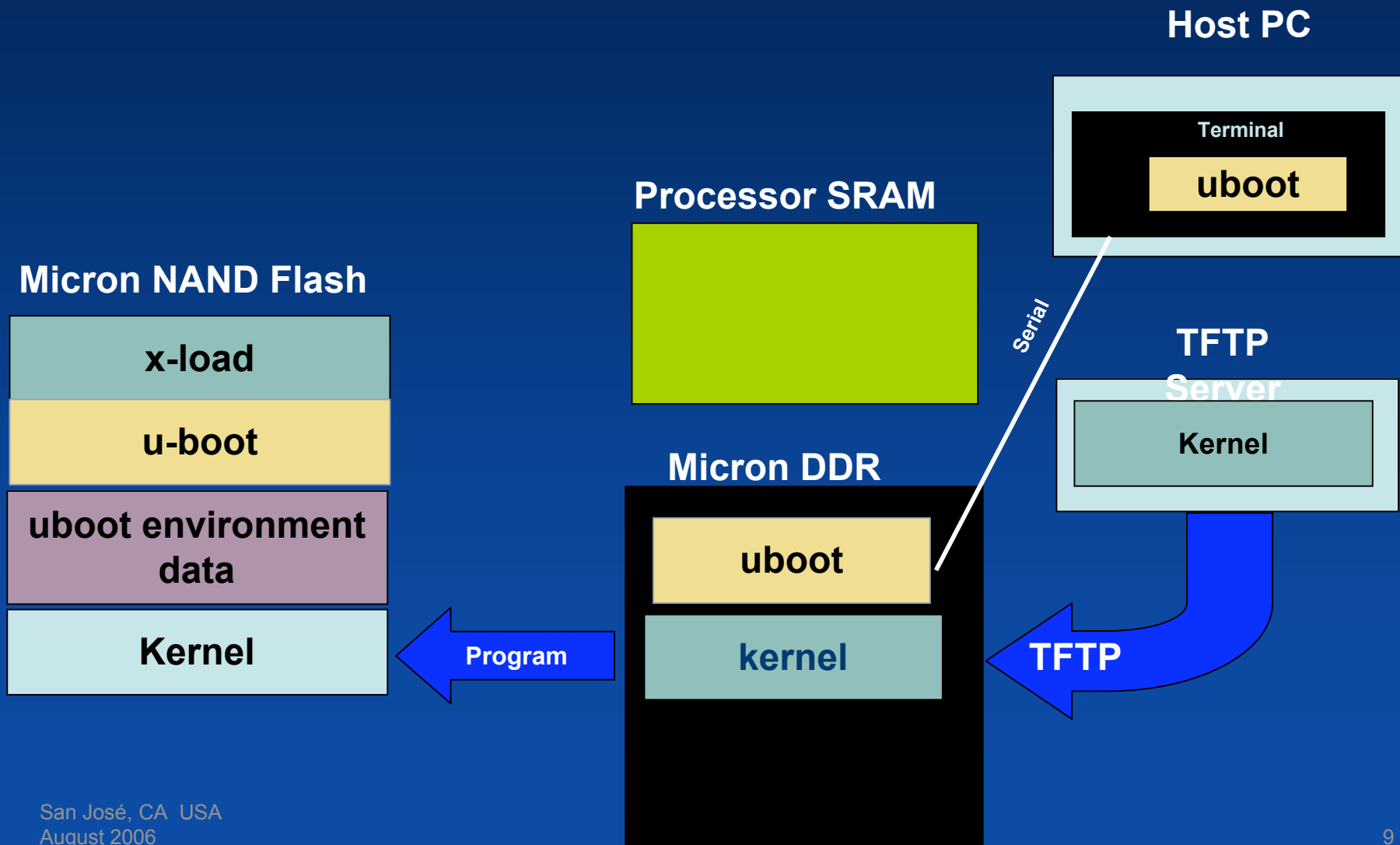
Writing code to NAND Flash:
Step 5 - Write File System to NAND

Boot from NAND Process

Processor ROM

ROM Code

ROM Process

Micron NAND Flash

x-load

uboot

uboot environment data

Kernel

File System

xload Process

Processor SRAM

x-load

uboot Process

Micron DDR

uboot

Kernel Process

Kernel

San José, CA USA
August 2006

11
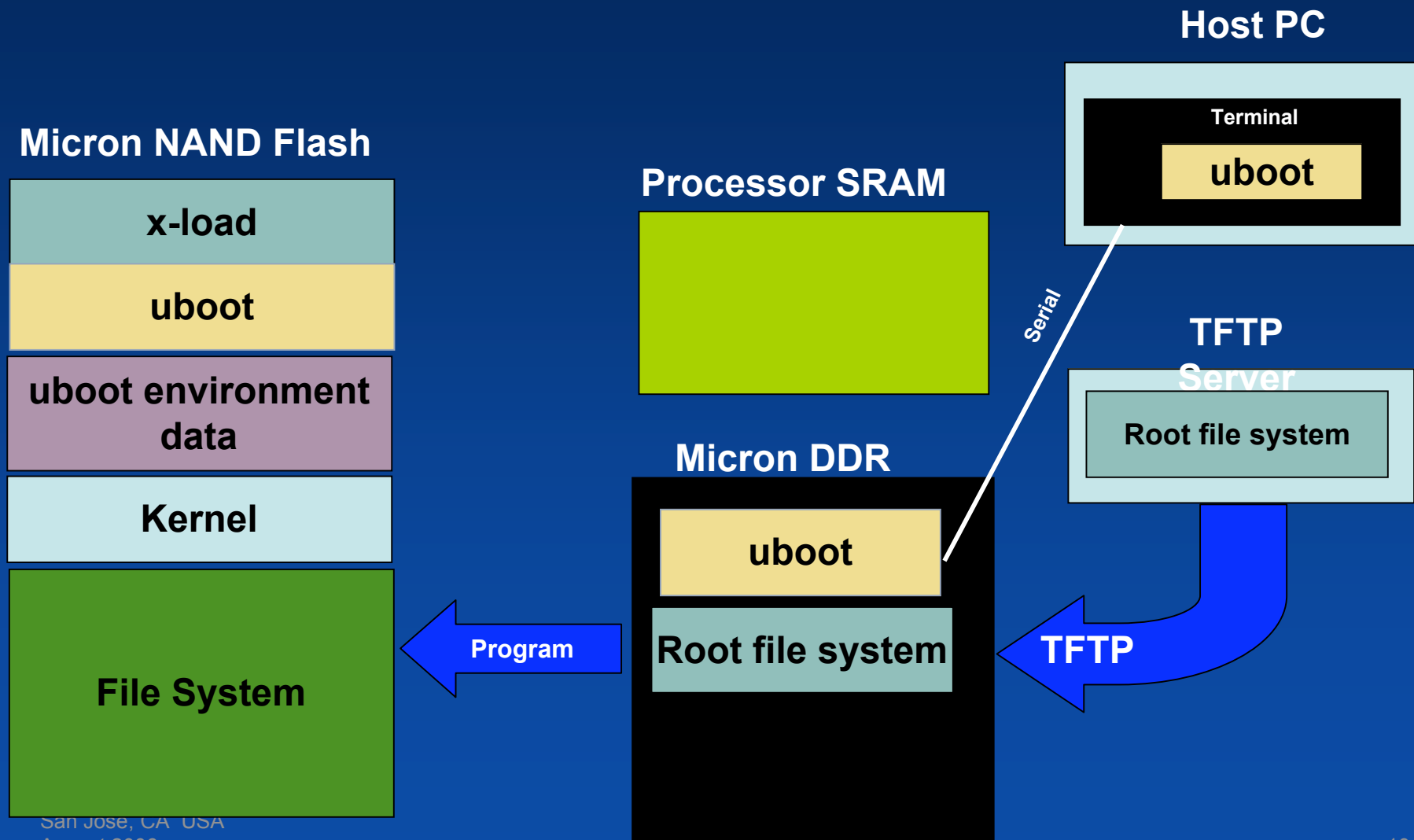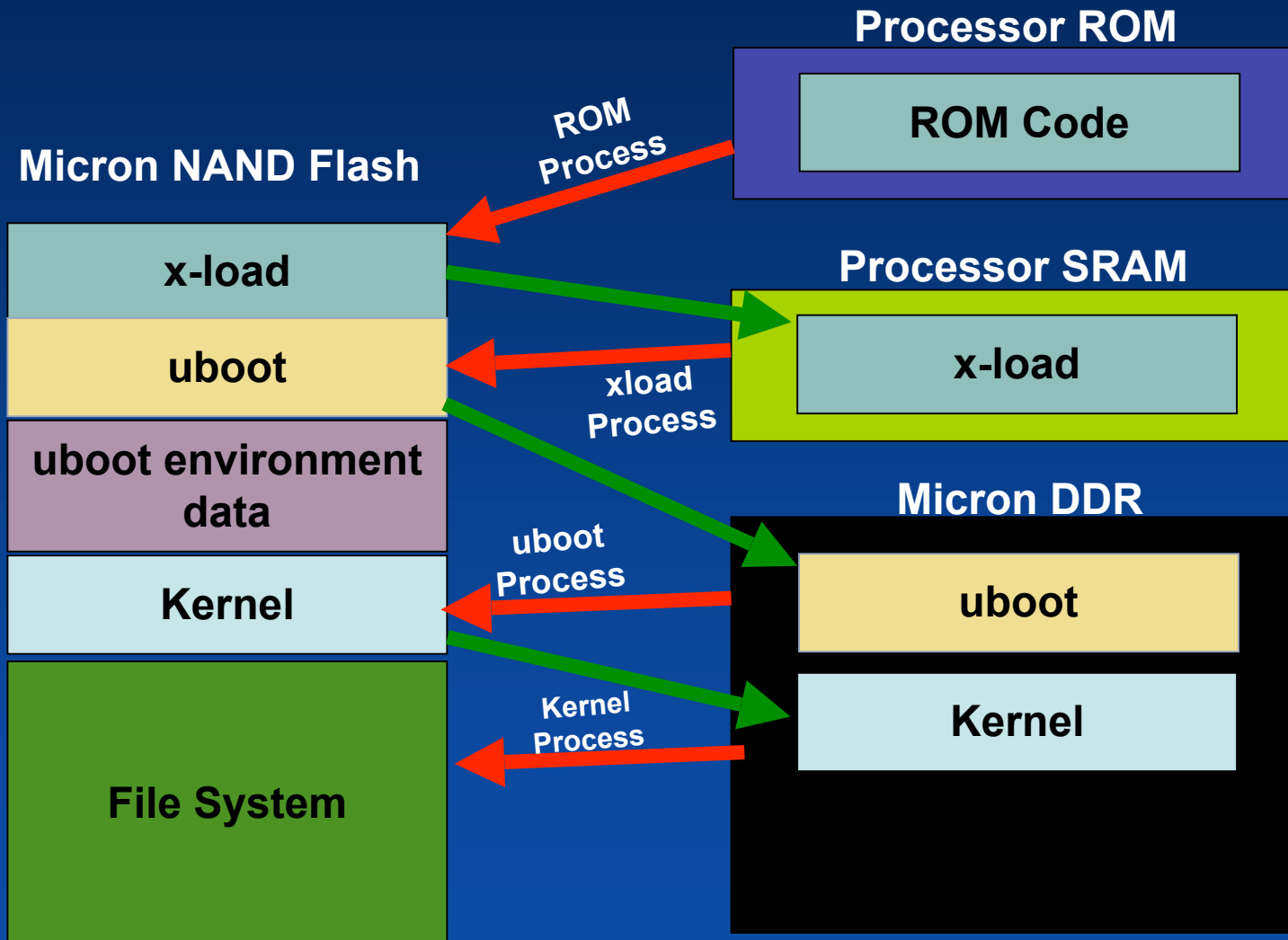
Recommendations to Maximize Boot Code Reliability

- Program each page of NAND flash in a single program operation.

- Compare boot code in NAND flash against original binary image to ensure programming was successful.

- Maximize error correction in code storage areas of NAND flash.

- Avoid excessive reads to blocks of NAND Flash which store code.

Biography

- Wes Prouty is a Sr. Applications Engineer for Micron Technology, Inc and is responsible for NAND flash wireless and software applications. He has a BSME from University of Idaho and an MSEE from Boise State University. Wes has 10 years of experience in design and test of embedded applications and memory devices.