

Boot an Entire Embedded Linux System Using Macronix High Density NOR Flash

Table of Contents

TABLE OF CONTENTS	1
1 PRODUCT CHARACTERISTIC.....	3
1.1 TRADITIONAL NOR FLASH VS HIGH DENSITY NOR FLASH.....	3
1.2 HIGH DENSITY NOR FLASH PARAMETERS	4
1.3 THE ADVANTAGES AND BENEFITS OF HIGH DENSITY NOR FLASH.....	4
1.4 HIGH DENSITY NOR FLASH USE CASES.....	5
1.5 MARKET SUBSTITUTION POTENTIAL	5
1.6 CONCLUSION.....	6
2. HIGH DENSITY NOR FLASH IN EMBEDDED LINUX BOOT SCENARIOS	6
2.1 EMBEDDED LINUX BOOT	6
2.1.1 EMBEDDED LINUX BOOT PROCESS	6
2.1.2 COMMON EMBEDDED LINUX BOOT METHODS	6
2.2 EMBEDDED LINUX SYSTEM BOOTING FROM HIGH DENSITY NOR FLASH	7
2.2.1 FEATURES.....	7
2.2.2 FLOW	7
2.2.3 ADVANTAGES.....	8
2.3 EMBEDDED LINUX SYSTEM BOOTING FROM SD CARD.....	9
2.3.1 FEATURES	9
2.3.2 FLOW	9
2.3.3 ADVANTAGES.....	9
2.3.4 DISADVANTAGES	10
2.4 EMBEDDED LINUX SYSTEM BOOTING FROM NOR + EMMC	10
2.4.1 FEATURES	10
2.4.2 FLOW	10
2.4.3 ADVANTAGES.....	11
2.4.4 DISADVANTAGES	11
2.5 EMBEDDED LINUX SYSTEM BOOTING FROM NOR + NAND.....	12
2.5.1 FEATURES.....	12
2.5.2 FLOW	12
2.5.3 ADVANTAGES.....	13
2.5.4 DISADVANTAGES	13
2.6 COMPARISON OF EMBEDDED LINUX BOOT	14



3. THEORETICAL PERFORMANCE COMPARISON	14
3.1 THEORETICAL PERFORMANCE COMPARISON TABLE OF HIGH DENSITY NOR FLASH.....	14
4. ACHIEVEMENT EXHIBITION	15
4.1 HARDWARE COMPONENTS	15
4.2 SYSTEM BLOCK DIAGRAM.....	16
4.3 BOOT FROM 3D NOR FLASH.....	17
4.4 COMPARISON OF BOOT TIME	18
5. REVISION HISTORY	18

1 Product Characteristic

In embedded systems, the boot device for Linux images plays a crucial role during the startup process. Common options include SD cards, eMMC, and NOR Flash. While SD cards and eMMC are favored for their larger storage capacities, NOR Flash stands out for its high reliability and fast random access, making it an ideal choice for boot code and compact systems. Traditional NOR Flash is often limited to a capacity of tens of megabytes, making it challenging to accommodate a complete Linux operating system and large applications. However, high density Flash products from Macronix, including Octa Flash and 3D NOR Flash, overcome the limitations of Traditional NOR Flash. These products offer capacities of 256MB or higher and support high-frequency operation exceeding 250MHz, enabling the storage of both Linux images and file systems on a single chip. With exceptional performance and high reliability, they provide an ideal high density solution for efficient boot and data storage in embedded systems.

1.1 Traditional NOR Flash vs High Density NOR Flash

In the NOR Flash market, "high density" is typically defined relative to the capacity range of Traditional NOR Flash. The general classification standards in the industry are as follows:

1. Traditional NOR Flash

a) Typically ranges from 4MB to 256MB. These products are primarily used for storing boot code (bootloader), configuration data, or simple embedded applications.

2. High Density NOR Flash

a) Generally refers to NOR Flash with capacities of 256MB, 512MB, or 1GB, especially products with capacities of 256MB or higher.

b) These products are suitable for storing operating systems (e.g., Linux images), file systems (e.g., UBIFS, YAFFS), and some application data.

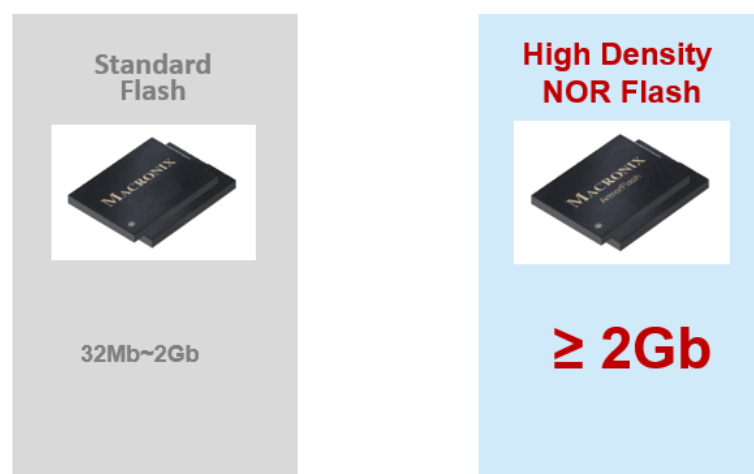


Figure 1-1: Traditional NOR Flash vs High Density NOR Flash

1.2 High Density NOR Flash Parameters

- Support Serial Peripheral Interface – Mode 0
- Protocol Support
 - 1-1-1, 1-1-8, 8-8-8 mode
 - Support DTR mode
- Support Simultaneous Read while Write operation
- Multiple bank architecture
 - Four uniform banks
- Fast frequency support
 - Support clock frequency up to
 - ◆ 1-1-1 mode 133Mhz
 - ◆ 1-1-8 mode 166Mhz
 - ◆ 8-8-8 mode 200Mhz
- Support Execute-in-place
- Configurable Sector Structure
 - 64 * 4KB sector on top / or bottom
 - 64KB uniform equal Blocks architecture
- Write Buffer
 - 256byte page buffer
 - Support interruptible write
 - Support write buffer read
- Reset Feature
 - H/W Reset # Pin
 - S/W Reset Command
- Data Integrity Check function
 - ECC
 - Bus CRC
- Typical 100000 erase/program cycles
- Compiles with AEC-Q100
- Input Data Format
 - SPI
 - OPI
- DOPI Output Data Format: Byte mode data sequence
- Program/Erase Suspend and Resume operation

1.3 The Advantages and Benefits of High Density NOR Flash

1. High Capacity

a) Advantage: Provides capacities of 256MB, 512MB, or higher, far exceeding the 64MB limit of traditional NOR Flash.

b) Benefit: Enables storage of a complete Linux operating system, file systems (e.g., UBIFS, YAFFS), and user applications on a single chip, simplifying design and storage management.

2. High Performance

a) Advantage: Supports operating frequencies of up to 250MHz or higher, as well as high-speed Octal SPI interfaces.

b) Benefit: Significantly improves data throughput, reduces boot time and data read latency, and meets the demands of embedded applications requiring high real-time performance.

3. High Reliability

- a) Advantage: Supports numerous program/erase cycles (typically 100,000 cycles) and long data retention periods (typically 20 years).
- b) Benefit: Suitable for scenarios with high requirements for data reliability and durability, such as industrial control and automotive electronics.

4. Low Power Consumption

- a) Advantage: Lower standby and operating power consumption compared to eMMC and NAND Flash.
- b) Benefit: Ideal for low-power applications such as IoT devices and smart terminals.

5. Simplified Design

- a) Advantage: Does not require complex Error Correction Code (ECC) management, reducing controller complexity.
- b) Benefit: Lowers system design costs and shortens product development cycles.

1.4 High Density NOR Flash Use Cases

1. Automotive Electronics:

- a) High Density NOR Flash can be used to store ADAS data, in-vehicle dashboard operating systems, and configuration files. Its high reliability and fast boot capability make it ideal for the automotive industry.

2. Industrial Automation:

- a) In PLCs (Programmable Logic Controllers) and HMIs (Human-Machine Interfaces), high density NOR Flash provides reliable real-time data storage solutions.

3. Consumer Electronics:

- a) Used in smart home devices, AR/VR devices, and other compact smart terminals to enable high-speed data processing and low-power operation.

4. IoT Devices:

- a) Supports firmware updates (OTA), and its high reliability makes it a preferred storage solution for edge devices.

1.5 Market Substitution Potential

Due to its unique characteristics, High Density NOR Flash products can gradually replace other storage solutions in the following areas:

1. Replacement of Traditional NOR Flash

- a) Scenario: For applications requiring larger storage capacities, traditional NOR Flash (4MB-256MB) can no longer meet the demand.

b) Example: More complex boot code, multi-operating system support, and large firmware updates.

2. Partial Replacement of eMMC

a) Scenario: In scenarios with high real-time requirements or low capacity demands, the latency and power consumption of eMMC may become bottlenecks.

b) Example: Industrial automation equipment, automotive electronics such as dashboards and ADAS (Advanced Driver Assistance Systems).

3. Partial Replacement of NAND Flash

a) Scenario: In small embedded systems, NAND Flash requires a complex controller to manage ECC, while High Density NOR Flash provides a plug-and-play solution.

b) Example: Consumer electronics and IoT terminals where reliability and low latency are critical.

1.6 Conclusion

High-Density NOR Flash products (e.g., 3D NOR Flash and Octa Flash) offer key advantages in embedded Linux boot processes. With capacities of 256MB to 512MB or more, they support the storage of complete operating systems, file systems, and applications on a single chip, streamlining design and management.

Operating at up to 250MHz with high-speed interfaces, they enhance data throughput and boot performance to meet real-time demands. Their high reliability, long data retention, and extensive program/erase cycles make them ideal for applications like ADAS, dashboards, PLCs, and HMIs.

With low power consumption and no need for complex controllers, High-Density NOR Flash is also perfect for IoT and smart devices. It can partially replace traditional NOR, NAND Flash, and eMMC, providing a robust, efficient solution for embedded Linux boot needs.

2. High Density NOR Flash In Embedded Linux Boot Scenarios

2.1 Embedded Linux Boot

2.1.1 Embedded Linux Boot Process

During the Embedded Linux boot process, the U-Boot Bootloader first initializes the hardware and then loads the Kernel along with the Device Tree Blob (DTB), which provides the Kernel with hardware configuration details necessary for initializing each device correctly. The Kernel then mounts the Root Filesystem (RootFS), transitions to user space, and completes the system boot, bringing it into a fully operational state.

2.1.2 Common Embedded Linux boot methods

1. U-Boot, Linux Kernel, RootFS can be stored in SD card.
2. U-Boot in NOR Flash, and others in eMMC.

3. U-Boot in NOR Flash, and others in NAND.
4. U-Boot, Linux Kernel, RootFS can be stored in High Density NOR Flash.

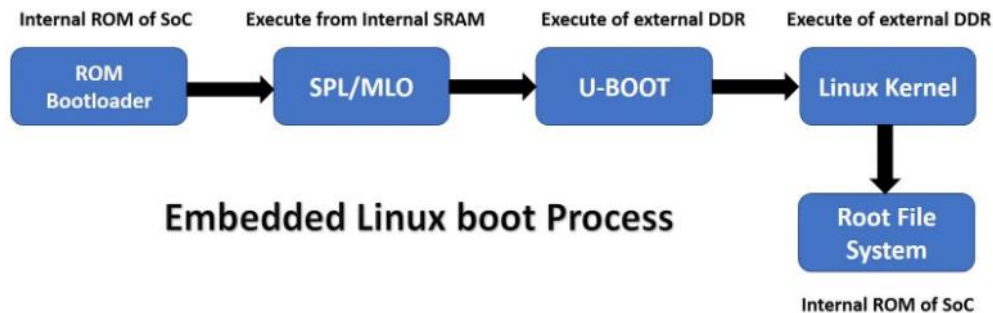


Figure 2-1: Embedded Linux boot process

2.2 Embedded Linux System Booting from High Density NOR Flash

2.2.1 Features

In a High Density NOR Flash Embedded Linux boot, the NOR Flash typically uses 8 IO lines and operates at speeds up to 250MHz, providing fast read access for boot code and Kernel data. This high-speed, wide-bus setup enables efficient direct booting from NOR, suitable for systems requiring high reliability.

2.2.2 Flow

In a High Density NOR Flash Embedded Linux boot, the MCU/MPU reads U-Boot from NOR into RAM, where U-Boot initializes and loads the Kernel and Device Tree Blob (DTB) from NOR. The Kernel then runs from RAM, mounts the root Filesystem directly from NOR, and executes applications from the RootFS to complete the boot process.

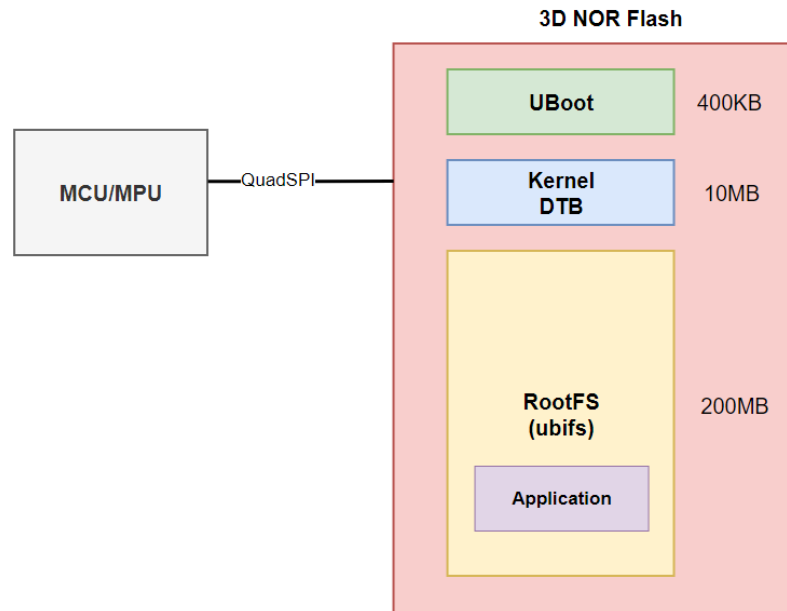


Figure 2-2: Boot From High Density NOR Flash

2.2.3 Advantages

1. High reliability, particularly for boot and critical system components.
2. Simplified storage and boot process without the need for wear leveling.
3. Suitable for read-heavy, low-write systems.
4. Streamlined design without the need for multiple storage devices.

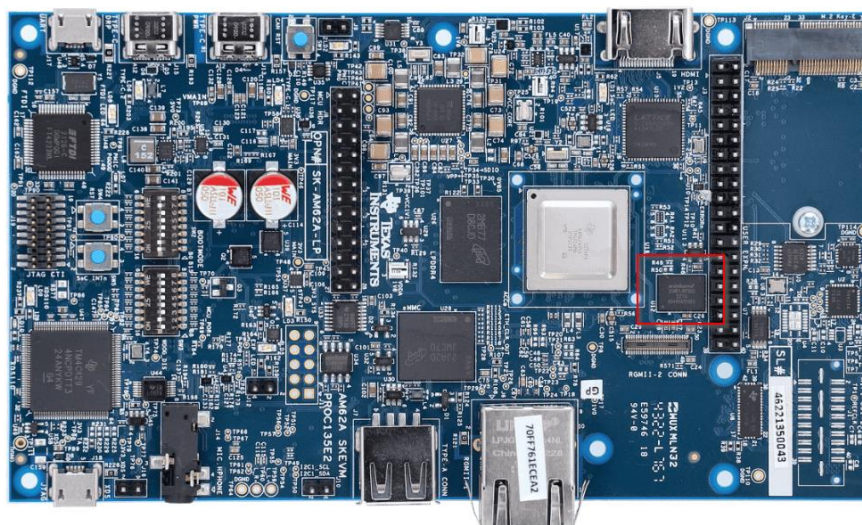


Figure 2-3: Boot From High Density NOR Flash

*On the SK-AM62A-LP development board, we replaced the NAND Flash with Octa NOR Flash,
enabling the system to boot Linux directly from the Octa NOR Flash.*

2.3 Embedded Linux System Booting from SD Card

2.3.1 Features

In Embedded Linux booting from an SD card, the SD card typically operates with 4 data lines (4 IO) to enhance data transfer rates and supports up to 50MHz clock speed in high-speed mode, achieving faster boot times and efficient data handling.

2.3.2 Flow

In the SD card boot process, the MCU/MPU first reads U-Boot from the SD card into RAM, where U-Boot initializes and then loads the Kernel and Device Tree Blob (DTB) into RAM. The Kernel then executes from RAM, mounting the RootFS directly from the SD card to complete the boot.

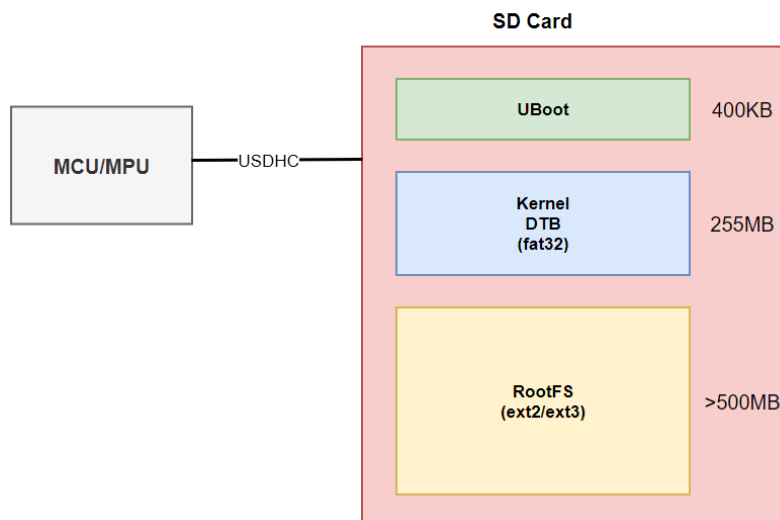


Figure 2-4: Boot from SD Card

2.3.3 Advantages

1. Low cost, ease of replacement, and availability.
2. Simple boot process with no need for complex Flash management.
3. Useful for development, prototyping, or low-cost applications.

2.3.4 Disadvantages

1. Limited write endurance and risk of data corruption
2. Simple Slower performance compared to other storage options like eMMC or NAND.
3. Less reliable and secure, making it less ideal for production systems.



Figure 2-5: Raspberry Pi uses an SD card to boot Embedded Linux

2.4 Embedded Linux System Booting from NOR + eMMC

2.4.1 Features

In a NOR + eMMC Embedded Linux boot setup, the NOR Flash generally supports 4 or 8 I/O lines and operates at speeds of 100-200MHz, allowing for rapid boot code execution. This configuration often requires dual controllers to enable efficient communication with both NOR Flash and eMMC, striking a balance between fast boot performance and ample storage capacity.

2.4.2 Flow

During the Embedded Linux boot process, the U-Boot bootloader first initializes the hardware and then loads the Kernel along with the Device Tree Blob (DTB), which provides the Kernel with hardware configuration details necessary for initializing each device correctly. The Kernel then mounts the Root Filesystem (RootFS), transitions to user space, and completes the system boot, bringing it into a fully operational state.

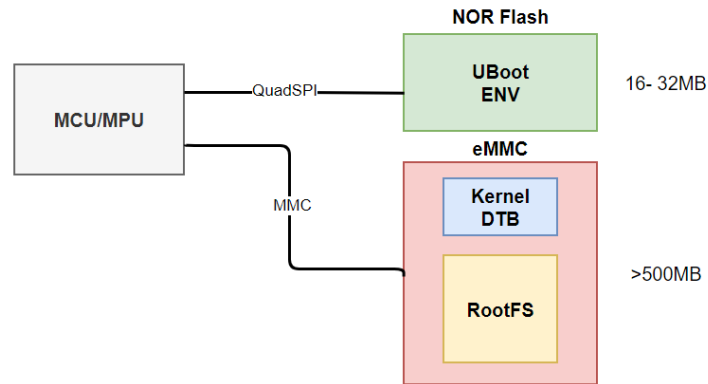


Figure 2-6: Boot from NOR + eMMC

2.4.3 Advantages

1. Fast boot speed, enabling quicker kernel and file system loading.
2. High data stability with built-in error correction.
3. Suitable for high-reliability applications like automotive and industrial automation.

2.4.4 Disadvantages

1. More complex hardware design and resource requirements.
2. Limited lifespan with frequent writes, less ideal for high-write environments.

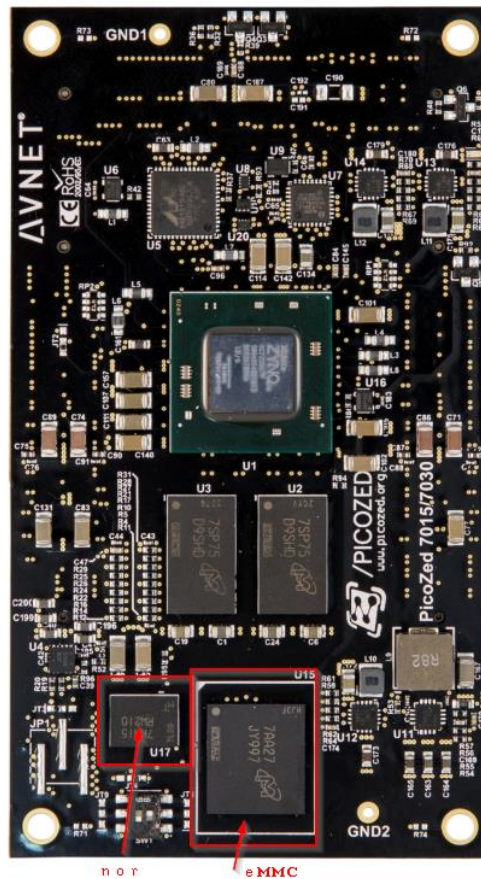


Figure 2-7: Avnet PicoZed can boot from NOR + eMMC

2.5 Embedded Linux System Booting from NOR + NAND

2.5.1 Features

In a NOR + NAND Embedded Linux boot setup, the NOR Flash typically supports 4 I/O lines and operates at speeds of 133MHz, enabling quick boot code execution. This configuration may require dual controllers for efficient communication with both NOR Flash and NAND, balancing fast initial boot with high density, cost-effective storage.

2.5.2 Flow

In a NOR + NAND Linux boot process, the MCU/MPU first reads U-Boot from NOR Flash into RAM, where U-Boot initializes and then loads the Kernel and Device Tree Blob (DTB) from NAND into RAM. The Kernel then runs from RAM, mounting the root filesystem directly from the eMMC to complete the boot.

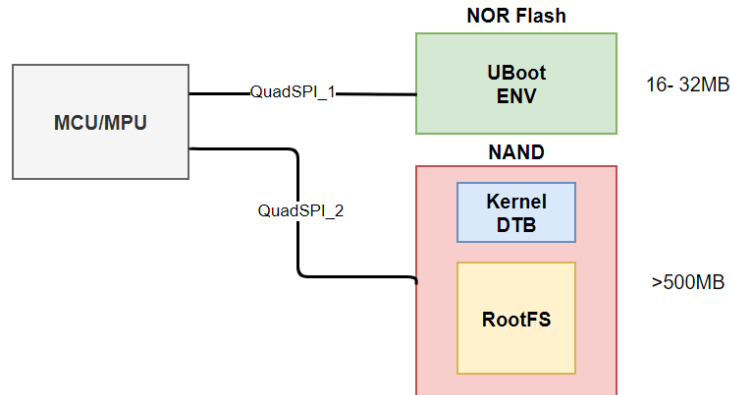


Figure 2-8: Boot from NOR + NAND flash

2.5.3 Advantages

1. Cost-effective, ideal for high density storage in budget-sensitive devices.
2. Suitable for applications with lower boot speed requirements.
3. High capacity at lower costs, making it practical for consumer devices.

2.5.4 Disadvantages

1. Slower boot speed and longer load times.
2. Lower data stability, with potential for bad blocks over time.
3. Weaker error correction, less suitable for high-integrity data needs.
4. Requires Flash Translation Layer (FTL) management in software to handle wear leveling and bad block management.

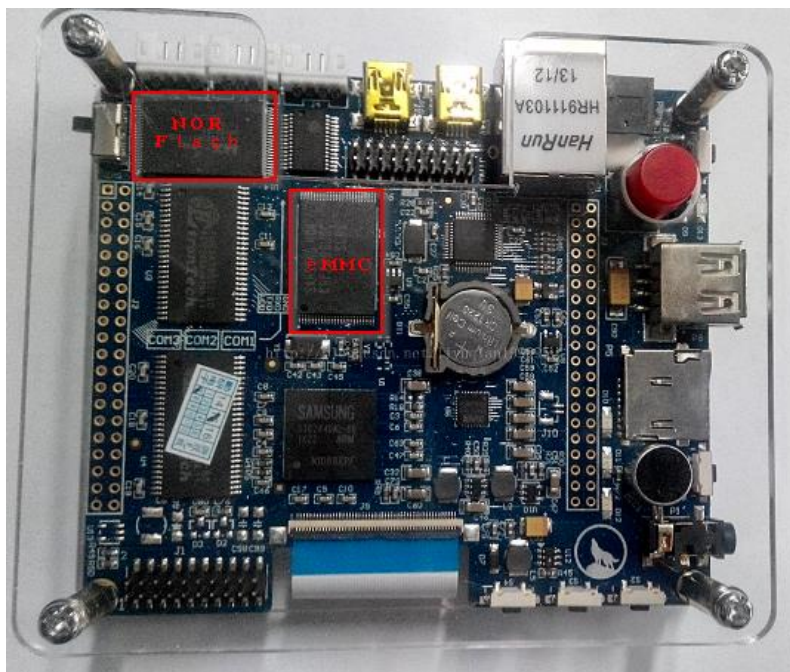


Figure 2-8: JZ2440 board can boot from NOR + NAND

2.6 Comparison of Embedded Linux boot

Index	Item	SD	NOR + eMMC/NAND	High Density NOR Flash
1	Storage Capacity	High	Middle	Middle
2	Data Read Speed	Low	Middle	High
3	System Boot Speed	Low	Middle	High
4	System Stability	Low	Middle	High
5	Data Integrity	Low	Middle	High
6	Power Loss Protection	Low	Middle	High
7	Application Scenarios	Middle	Middle	Middle
8	Cost	Low	High	Middle

3. Theoretical Performance Comparison

3.1 Theoretical Performance Comparison Table of High Density NOR Flash

Item	Frequency	STR/DTR	IO NUM	Bandwidth(Max)
SD	50Mhz	STR	4	25MB/s
Standard NOR Flash	133Mhz	STR	4/8	133MB/s
NAND	133Mhz	STR	4	66MB/s
eMMC	50~200Mhz	DTR	8	400MB/s
High Density NOR Flash(Octabus)	250Mhz	DTR	8	500MB/s

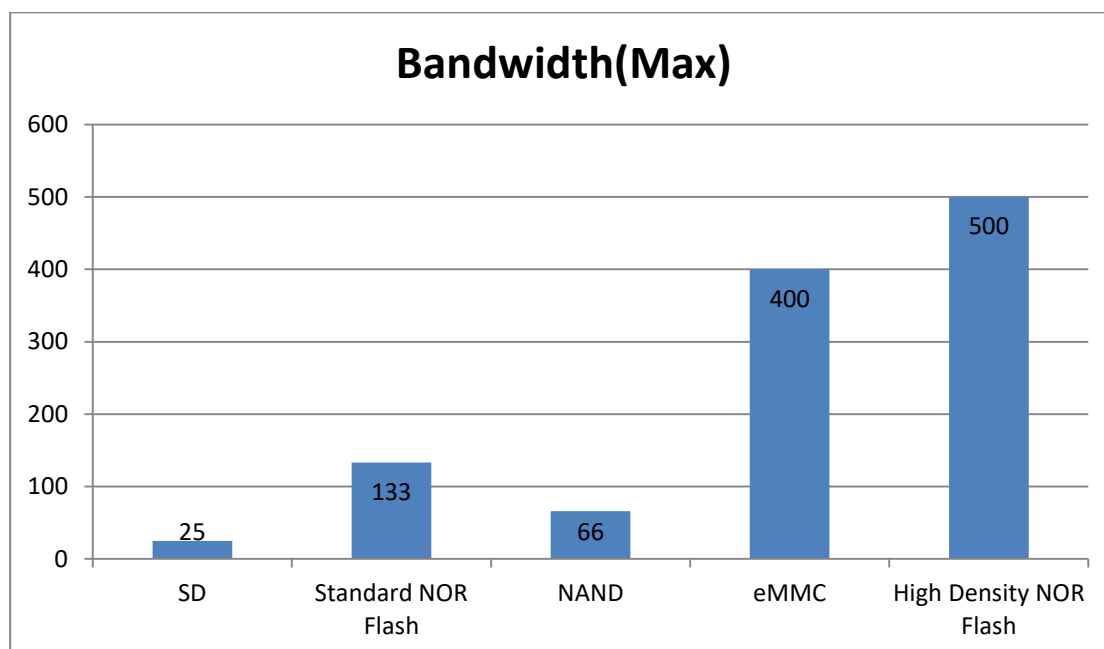


Figure 3-1: Theoretical Performance Comparison

4. Achievement Exhibition

4.1 Hardware components

1. TI SK-AM62A-LP
2. 1080p LCD
3. High Density NOR Flash



Figure 4-1: Demo

4.2 System Block Diagram

Traditional NOR Flash, with its limited capacity, typically holds only the bootloader, requiring external storage for the complete Embedded Linux system. The emergence of High Density NOR Flash, however, allows for the entire system—including the U-Boot, Kernel, DTB, and RootFS to be stored internally. This innovation simplifies design and improves performance and reliability.

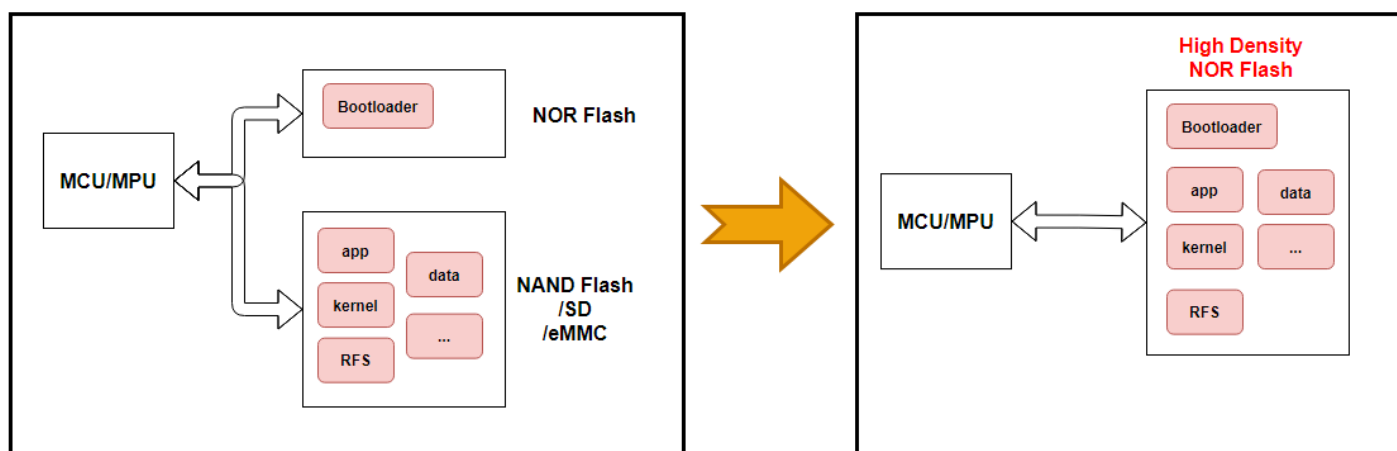


Figure 4-2: System Block Diagram

4.3 Boot from 3D NOR Flash

In Embedded development, we partition a 256MB Octa Flash chip into five sections, including a root filesystem (RootFS). To fit within storage limits, we streamline the RootFS by removing unnecessary files and optimizing its layout. This reduces size, enhances boot speed, and maintains core functionality, which is essential for adapting to hardware constraints.

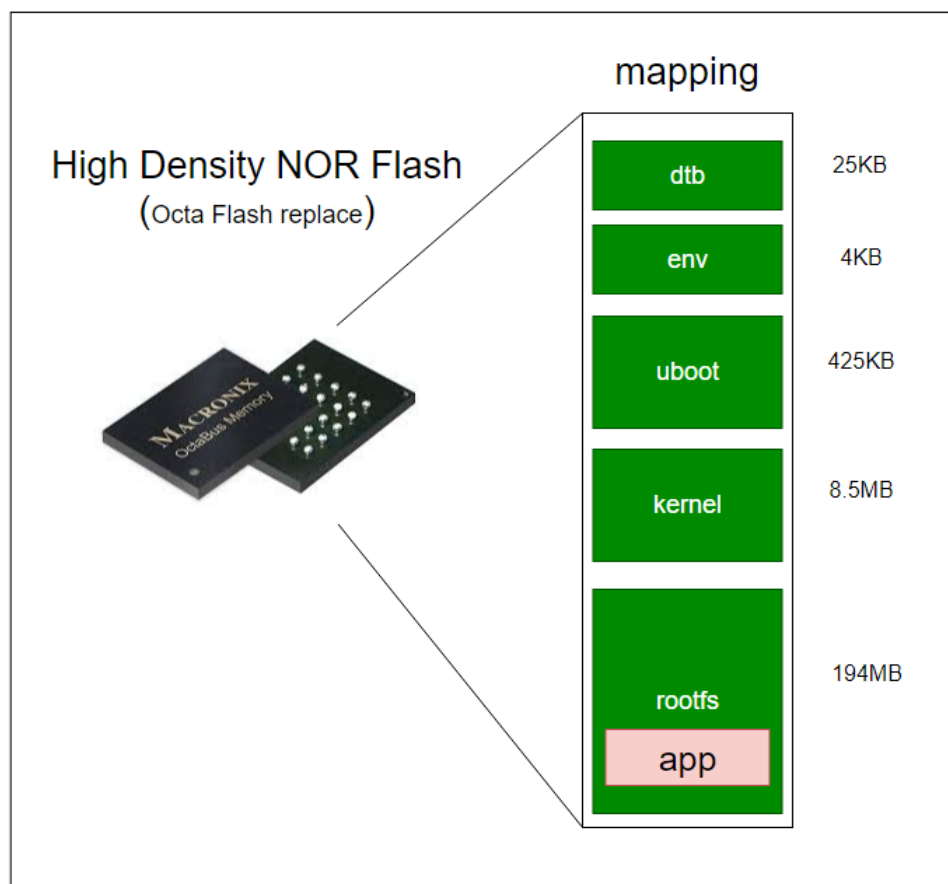


Figure 4-3: 3D NOR Flash storage structure

4.4 Comparison of Boot time

We have successfully implemented a complete Embedded Linux system on Octa Flash, achieving smooth application performance. Its loading speed for U-Boot is comparable to eMMC and SD cards. The upcoming 3D NOR Flash will enhance speeds and reduce startup times, optimizing performance for embedded applications.

Item	Boot time (SD)	Boot time (eMMC)	Boot time (High Density NOR Flash Octabus)
Protocol	4 I/O SDR	8 I/O SDR	8 I/O DTR
Frequency	50Mhz	50-200Mhz	200Mhz
U-Boot (include running)	2.31s	2.05	1.98s

5. Revision History

Revision No.	Description	Page	Date
Rev. 1	Initial Release	ALL	February 18, 2025
Rev. 2	Modify Boot time table	18	May 8, 2025



Except for customized products which have been expressly identified in the applicable agreement, Macronix's products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and not for use in any applications which may, directly or indirectly, cause death, personal injury, or severe property damages. In the event Macronix products are used in contradicted to their target usage above, the buyer shall take any and all actions to ensure said Macronix's product qualified for its actual use in accordance with the applicable laws and regulations; and Macronix as well as it's suppliers and/or distributors shall be released from any and all liability arisen therefrom.

Copyright© Macronix International Co., Ltd. 2025. All rights reserved, including the trademarks and tradename thereof, such as Macronix, MXIC, MXIC Logo, MX Logo, Nbit, Macronix NBit, KH Logo, MXSMIO, OctaBus, OctaFlash, ArmorFlash, LybraFlash, FortiX. The names and brands of third party referred thereto (if any) are for identification purposes only.

For the contact and order information, please visit Macronix's Web site at: <http://www.macronix.com>