**NAND Boot**

**1.1 Basic Boot Operation**

Boot the device from an external NAND memory. Although the memory is interfaced with EMIF, it is not the same as EMIF boot. Non-secure NAND booting requires the boot image to be in the form of a boot table.

In order for information to be read off a raw NAND, the geometry of the device is required. This includes:

• 8-bit or 16-bit data width

• Page size

• Number of pages per block

• Number of address cycles

Because the NAND market moves at such a fast rate, it cannot be guaranteed that all devices will be compatible. At run time the RBL will attempt to check for compatibility with the NAND. The RBL also supports a 4b ECC. The structure of the ECC is stored in the EMIF16 hardware ECC block and the format depends on the EMIF16 hardware design.

The RBL will first check for ONFI compliance. Most ONFI compliant devices should be supported. A read ID command will be issued to address 0x20 and four bytes will be read. If the four bytes match the word 0x49464E4F (or ‘O’ ‘N’ ‘F’ ‘I’) the RBL will attempt to read the ONFI parameter page for the device geometry.

After which, the RBL will request 4 bytes of identifier code from the NAND by issuing a read ID command to address 0x00. The first and second bytes are the manufacture ID and device ID, respectively, and the fourth byte has information on the NAND parameters. If the device was found not to be ONFI compliant, the RBL will try the following to determine geometry:

• Compare device ID with IDs and their parameters stored in ROM. It is impossible to have all devices listed on the table, and some devices may share IDs. If ID is matched, the configuration is used and the next step is skipped.

• Read 4th byte data. This happens when the RBL finds no match with the device ID. The RBL will assume an 8 bit data bus width and check the device Manufacture ID to indicate “Samsung format” or “Common format” when reading the 4th byte parameters for NAND geometry. If non-Samsung the RBL will be able to change the width to 16-bit if necessary. Unfortunately, Samsung devices will remain in 8-bit mode.

Refer to flowchart below on a visual of how the RBL configures geometry.

In this mode the boot ROM will first check the 1st block set in the device configuration bits for bad block error. The boot ROM checks the Out of Band (OoB) region’s first 6 bytes of page 0 and page 1 of each block. If the bytes are all set to 0xFF, then the ROM marks the block as good and starts reading the block and load the sections in the internal memory accordingly. It is up to the application to make sure that the bad blocks are marked for ONFI spec and ROM boot. After transfer is complete the boot ROM will branch to the load address and begin executing.

NAND boot geometry flow



**1.2 Boot Mode setting**

Devstat bootstrap settings.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DEVSTAT BITS | | | | | | | | | | | | | | |
| 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Device  Name |
| Pll Config | | | Device Configuration | | | | | | Boot Device | | | |  |
| Extended Dev Cfg | | |
| PLL2 | PLL1 | PLL0 | Wait | Width | CS1 | CS0 | Sub-Mode | rsvd | 0 | 0 | 0 | 0 | ENDIAN | Sleep  EMIF16 |
| Lane | Rate1 | Rate0 | R CLK1 | R CLK0 | Rsvd | 0 | 0 | 0 | 1 | SRIO |
| S CLK1 | S CLK0 | EXT1 | EXT0 | DevID2 | DevID1 | DevID0 | 0 | 1 | 0 | ETHERNET |
| 1st block | 1st block | 1st block | 1st block | 1st block | I2c | 0 | 0 | 1 | 1 | NAND |
| R CLK | BAR3 | BAR2 | BAR1 | BAR0 | Rsvd | 0 | 1 | 0 | 0 | PCIe |
| 0 | ADDR1 | ADDR0 | Speed | Param5 | Param4 | Param3 | Param2 | Param1 | Param0 | 1 | 0 | 1 | I2C Master |
| 1 | ADDR6 | ADDR5 | ADDR4 | ADDR3 | ADDR2 | ADDR1 | ADDR0 | Rsvd | 0 | 1 | 0 | 1 | I2C Slave |
| Mode1 | Mode0 | 4/5 pin | Addr Width | CS1 | CS0 | Param3 | Param2 | Param1 | Param0 | 1 | 1 | 0 | SPI |
| PLL2 | PLL1 | PLL0 | rsvd | Rate1 | Rate0 | R CLK1 | R CLK0 | Rsvd | 0 | 1 | 1 | 1 | HyperLink |
| speed | speed | parity | parity | port | Rsvd | 1 | 0 | 0 | 0 | UART |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NAND Boot Device Configuration Bit Fields | | | | | |
| 10 | 9 | 8 | 7 | 6 | 5 |
| 1St Block | | | | | I2C |

Table 13. NAND Boot Device Configuration Bit Fields

|  |  |  |
| --- | --- | --- |
| NAND Configuration Bit Field Descriptions | | |
| Bit Field | Value | Description |
| 1st Block | 0-31 | The initial block read from the NAND |
| I2C | 0 | Parameters are not read from the I2C |
| 1 | Parameters are read from the I2C |

Table 14. NAND Configuration Bit Field Descriptions

The I2C NAND configuration structure is formatted as 16 bit words. Each word is read from the I2C in big endian format, most significant byte first, regardless of the endianness of the device. Even though this provision is available it is recommended to do a second stage I2C boot where you load the entire NAND boot parameter table.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I2C NAND Configuration Structure | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAGIC A = 0x10B3 | | | | | | | | | | | | | | | |
| MAGIC B = 0x57A6 | | | | | | | | | | | | | | | |
| Column Bytes | | | | Row Bytes | | | | Page Size | | | | Pages Per Block | | | |
| BusWidth | | | | Reserved | | | | | | | | | | | |

Table 86. I2C Nand Configuration Structure

|  |  |  |
| --- | --- | --- |
| I2C NAND Field Description | | |
| Field | Value | Description |
| Column Bytes | 0-15 | Number of bytes used to form the column address |
| Row Bytes | 0-15 | Number of bytes used to form the row address |
| Page Size | 0-15 | Bytes per page is 2page size field |
| Pages per Block | 0-15 | Number of pages per block is 2pages per block field |
| Bus Width | 0 | Bus is 8 bits |
| 1-15 | Bus is 16 bits |

Table 87. I2C NAND Field Descriptions