AM335x NAND Driver's Guide

AM335x NAND Driver's Guide

Translate this page to cs - Česky Translate

Google™ Custom Search Search

Navigation

- Main Page
- All pages
- All categories
- Popular pages
- Popular authors
- Popular categories
- Category stats
- Recent changes
- Random page
- Help
- Google Search

Print/export

- Create a book
- Download as PDF
- Printable version

Toolbox

- What links here
- Related changes
- Special pages
- Permanent link
- Browse properties

AM335x NAND Driver's Guide

Linux PSP

Contents [hide]

Introduction

The general-purpose memory controller (GPMC) is an unified memory controller dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (only available in non-multiplexed mode) burst NOR flash devices
- NAND Flash
- Pseudo-SRAM devices

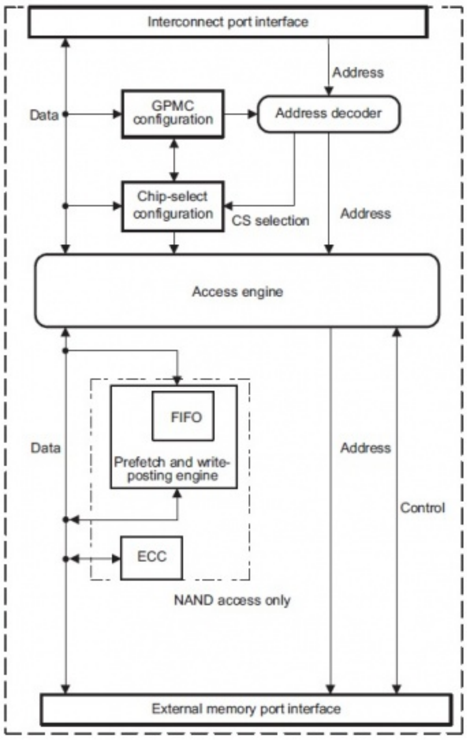
Features

Hardware Features

The GPMC can access various external devices through the L3 Slow Interconnect. The flexible programming model allows a wide range of attached device types and access schemes.

The GPMC consists of six blocks:

- Interconnect port interface
- Address decoder, GPMC configuration, and chip-select configuration register file
- Access engine
- Prefetch and write-posting engine
- Error correction code engine (ECC)
- External device/memory port interface



Software Features

- Access 8-bit NAND
- BCH8 ECC scheme with 8-bit error correction. BCH8 ECC Scheme used to make compatible with all other components (RBL, U-Boot..)
- Polled Prefetch Mode

AM335x NAND Device Layout

Layout for NAND flash device is described here.

ECC schemes usage table

ECC scheme used by different components is described in the table

Driver Configuration

To enable/disable NAND support, start the *Linux Kernel Configuration* tool:

```
host$ make menuconfig
```

Select *Device Drivers* from the main menu.

```
    ...
    ...
    [ ] Networking support --->
    Device Drivers --->
    File systems --->
    ...
    ...
```

- Enable below Configs to enable MTD Support along with MTD nand driver support

```
Device Drivers  --->
  <*> Memory Technology Device (MTD) support  --->
            [*]   Command line partition table parsing
            <*>   Direct char device access to MTD devices
            <*>   Caching block device access to MTD devices
            <*>   NAND Device Support  --->
                        <*>    NAND Flash device on OMAP2 and OMAP3
            <*>   Enable UBI - Unsorted block images  --->
```

Module Build

Module build for the NAND driver is supported. To do this, make the changes as below

```
Device Drivers  --->
  <*> Memory Technology Device (MTD) support  --->
            [*]   Command line partition table parsing
            <*>   Direct char device access to MTD devices
            <*>   Caching block device access to MTD devices
            <M>   NAND Device Support  --->
```

```
                      <M>    NAND Flash device on OMAP2 and OMAP3
            <*>   Enable UBI - Unsorted block images  --->
```

For loading modules follow the steps below

```
       target$ insmod drivers/mtd/nand/nand_ecc.ko
       target$ insmod drivers/mtd/nand/nand_ids.ko
       target$ insmod drivers/mtd/nand/nand.ko
       target$ insmod drivers/mtd/devices/omap2_elm.ko
       target$ insmod drivers/mtd/nand/omap2.ko
```

Device Interface

You always need a pair mtd**x**/mtdblock**x** of device nodes to access your flash file system. The mtd**x** is used to access the raw flash device, the mtdblock**x** is used to access the disk/block established in the raw flash.

Use MTD-Utils user space tools to access NAND partition from Linux console

Below are some of the examples for accessing NAND partitions

- Erase the nand partition (assume partition 3)

```
     target$ flash_eraseall /dev/mtd3
```

- write a file to nand partition (assume partition 4)

```
     target$ nandwrite -p /dev/mtd4 u-boot.img
```

- Mount nand partition from U-Boot as Root file system partition for Linux

```
     u-boot> setenv bootargs 'console=ttyO0,115200n8 root=ubi0:rootfs rw ubi.mtd=<partion_id>,2048 noinitrd rootfstype=ubifs mem=256M rootwait=1'
```

The value of PARTITION_ID depends on MTD device which holds the root filesystem. The below example assumes UBIFS file system is flashed on MTD partition 7 and respective device is enabled on the board.

```
     u-boot> setenv bootargs 'console=ttyO0,115200n8 root=ubi0:rootfs rw ubi.mtd=7,2048 noinitrd rootfstype=ubifs mem=256M rootwait=1'
```

Proc Interface

The `/proc/mtd` kernel interface is a status interface. A lot of useful information about the nand system can be found in the `/proc/mtd` file.

- Use /proc/mtd to get information on how many partitions are currently configured by the kernels flash driver.

```
     target$ cat /proc/mtd
```

You should see output similar to:

```
     target$ cat /proc/mtd
    dev:    size   erasesize  name
    mtd0: 00020000 00020000 "SPL"
    mtd1: 00020000 00020000 "SPL.backup1"
    mtd2: 00020000 00020000 "SPL.backup2"
    mtd3: 00020000 00020000 "SPL.backup3"
    mtd4: 001e0000 00020000 "U-Boot"
    mtd5: 00020000 00020000 "U-Boot Env"
    mtd6: 00500000 00020000 "Kernel"
    mtd7: 0f880000 00020000 "File System"
```

AM335x JFFS2 support

On AM335x, JFFS2 support has been super-seeded by UBIFS. Hence, Support for JFFS2 is disabled in default config.

To enable JFFS2, Refer AM335x JFFS2 Support Guide

*For technical support please post your questions at http://e2e.ti.com. Please post only comments about the article **AM335x NAND Driver's Guide** here.*

Links

ARM Microcontroller MCU | ARM Processor | Digital Media Processor | Digital Signal Processing | Microcontroller MCU | Multi Core Processor

Ultra Low Power DSP | 8 bit Microcontroller MCU | 16 bit Microcontroller MCU | 32 bit Microcontroller MCU

Categories: AM335x | Linux | PSP

Leave a Comment

This page was last modified on 15 March 2012, at 06:50.

This page has been accessed 2,614 times.

Privacy policy About Texas Instruments Embedded Processors Wiki Disclaimers