

# Hands-on: Analyzing NAND Flashes

- ▶ To Add a New NAND flash to the BSP:
  1. Use the NAND-flash-analysis spreadsheet to capture ID bytes, timings, and dimensions of the NAND flash
  2. Add these parameters into the BSP source code.
    - MTD standard table is checked first by MTD  
`drivers/mtd/nand/nand_ids.c`
    - Freescale BSP extended table is checked second  
`drivers/mtd/nand/nand_device_info.c`
    - For the NAND to be recognized, it must have an entry in both tables.

# Hands-on: Analyzing NAND Flashes

## ▶ drivers/mtd/nand/nand\_ids.c

```
struct nand_flash_dev nand_flash_ids[] = {
/*{name, id, pagesize, chipsize, erasesize,
  options}*/

{"NAND 2GiB 1,8V 8-bit",0xA5, 0, 2048, 0, LP_OPTIONS},
{"NAND 2GiB 3,3V 8-bit",0xD5, 0, 2048, 0, LP_OPTIONS},
.
.
.
}
```

# Hands-on: Analyzing NAND Flashes

```
▶ drivers/mtd/nand/nand_ids.c
static struct nand_device_info nand_device_info_table_large_mlc[]
__initdata =
{
    {
        .end_of_table           = false,
        .manufacturer_code     = 0xec,
        .device_code           = 0xd3,
        .cell_technology        = NAND_DEVICE_CELL_TECH_MLC,
        .chip_size_in_bytes     = 1LL * SZ_1G,
        .block_size_in_pages    = 128,
        .page_total_size_in_bytes = 2 * SZ_1K + 64,
        .ecc_strength_in_bits   = 4,
        .ecc_size_in_bytes      = 512,
        .data_setup_in_ns       = 20,
        .data_hold_in_ns        = 15,
        .address_setup_in_ns    = 20,
        .gpmi_sample_delay_in_ns = 6,
        .tREA_in_ns             = -1,
        .tRLOH_in_ns           = -1,
        .tRHOH_in_ns           = -1,
        "K9G8G08U0M, K9HAG08U1M",
    },
}
```