# NAND Flash Support in SAM3X Microcontrollers

## 1. Scope

The purpose of this application note is to introduce the NAND Flash memory technology and describe hardware and software requirements to interface NAND Flash memory with Atmel® SAM3X ARM® Cortex-M3®-based Microcontrollers that features an Embedded NAND Flash Controller.

The SAM3X microcontroller family features an External Bus Interface (EBI) providing NAND Flash protocol support via the Static Memory Controller (SMC) and embedded NAND Flash Controller. It also contains an Error Corrected Code Controller (ECC) which performs data error identification and single bit correction.

The related source code can be found at XXXXXXX

**AT91SAM ARM-based Flash MCU**

**Application Note**

11154A–ATARM–26-Jun-12

# 2. NAND Flash Overview

## 2.1 General Overview

NAND Flash technology provides a cost-effective solution for applications requiring high-density solid-state storage. The MT29F2GxxABx is a 2Gbit NAND Flash memory device. Micron NAND Flash devices include standard NAND Flash features as well as new features designed to enhance system-level performance.

The MT29F2G08ABA is mounted on the SAM3X-EK evaluation kit, this flash device, interfaced with SAM3X microcontrollers, will be considered as a reference example throughout the rest of this document.

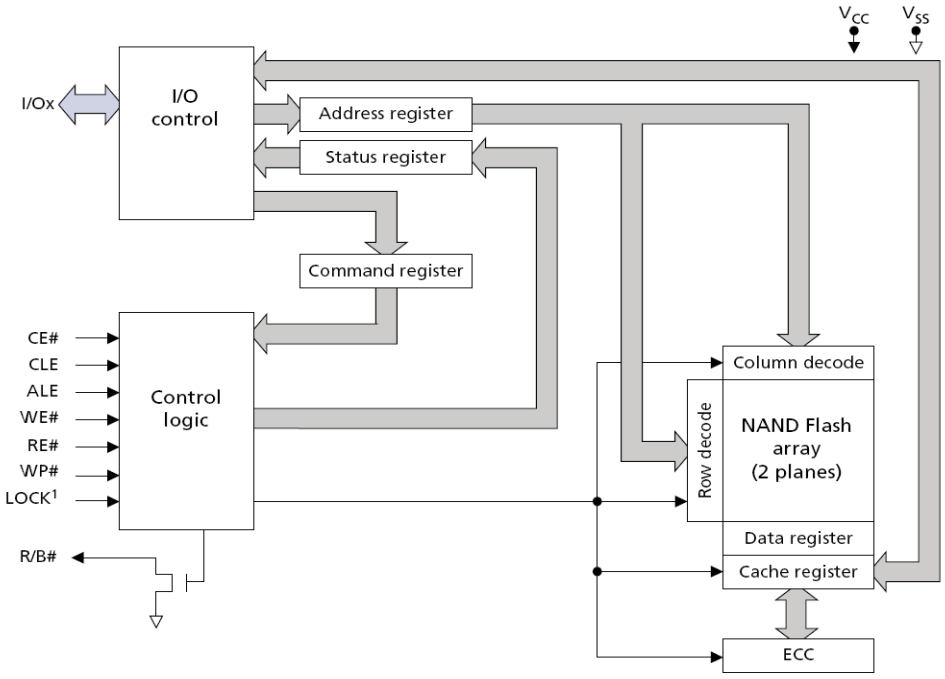
## 2.2 Internal Architecture

The MT29F2G08ABA uses NAND Flash electrical and command interface. Data, commands, and addresses are multiplexed onto the same pin and received by I/O control circuits.

The commands received at the I/O control circuits are latched by a command register and transferred to a control logic circuit; the addresses are latched by an address register and sent to a row decoder or a column decoder. The data are transferred to or from the NAND Flash memory array, byte by byte (x8) or word by word (x16), through a data register and a cache register. The NAND Flash memory array is programmed and read in page-based operations and is erased in block-based operations.

The MT29F2G08ABA also has a status register that reports the status of device operation.

**Figure 2-1.** NAND Flash Functional Block Diagram.

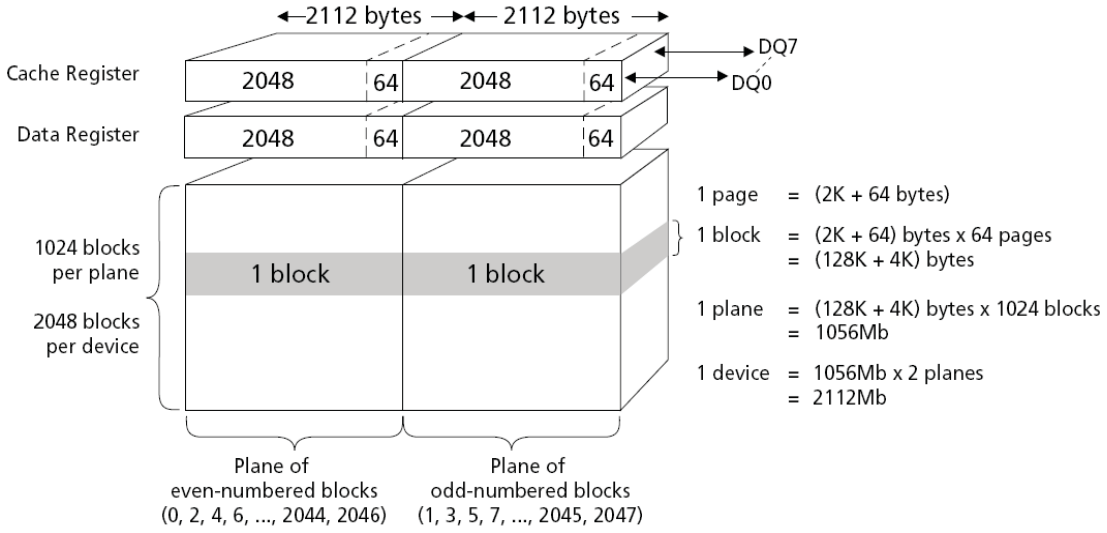



Note: 1. The LOCK pin is used on the 1.8V device.

## 2.3 Array Organization

The MT29F2G device contains 2,048 blocks. Each block is subdivided into 64 programmable pages. Each page consists of 2,112 bytes. The pages are further divided into a 2,048-byte data storage region with a separate 64-byte area. The 64-byte area is typically used for error management functions.

**Figure 2-2.** the array organization for MT29F2G08ABA(X8)



## 2.4 NAND Flash Hardware Interface

NAND Flash uses a multiplexed I/O interface and additional control signals. It is controlled by sending commands and addresses through an 8-bit or 16-bit bus to an internal command and address register. NAND Flash I/O device-type interface is composed of up to 24 pins.

**Table 2-1. NAND Flash Device Typical Hardware Interface**

| Pin Symbol | Pin function | Pin Description |
|---|---|---|
| CE# | Chip Enable | CE# is active when asserted LOW to enable or select the device. CE# pin must remain LOW during busy periods in order to prevent the device from entering standby mode and stopping the read operation in a mid cycle.<br><br>A subset of NAND Flash devices supports the CE# “Don’t Care” option which allows deselecting the device without terminating the operation in progress. Other devices on the same memory bus can then be accessed while the NAND Flash is busy with internal operations. |
| WE# | Write Enable | The WE# input controls writes to the I/O port. Commands, address and data are latched on the rising edge of the WE# pulse. |
| RE# | Read Enable | RE# enables the output data buffers. |
| CLE | Command Latch Enable | When CLE is HIGH, commands are latched into the NAND Flash command register on the rising edge of the WE signal. |
| ALE | Address Latch Enable | When ALE is HIGH, addresses are latched into the NAND Flash address register on the rising edge of the WE signal. |
| I/O[7:0] or I/O[15:0] | Input/Output Bus | The I/O pins are used for input commands, address and data, and to output data during read operations. The I/O pins float to high-z when the chip is deselected or when the outputs are disabled.I/O8 - I/O15 are used only in an X16 organization device. Since command input and address input are X8 operations, I/O8 - I/O15 are not used to input command and address. I/O8 - I/O15 are used only for data input and output. |
| WP# | Write Protect | The WP# pin provides inadvertent write/erase protection during power transitions. The internal high voltage generator is reset when the WP# pin is active low. |
| R/B# | Ready/Busy | If the NAND Flash device is busy with an ERASE, PROGRAM or READ operation, the R/B# signal is asserted LOW. The R/B# signal is an open drain output and requires a pull-up resistor to be correctly read. |
| LOCK | Block Lock | When LOCK is HIGH during power-up, the BLOCK LOCK function is enabled. To disable the BLOCK LOCK, connect LOCK to VSS during power-up, or leave it disconnected (internal pulldown). |

## 2.5 NAND Flash Timing Characteristics

Table 2-2 lists the Micron MT29F2G08ABA timing parameters.

**Table 2-2. Micron MT29F2G08ABA Timings (3.3V)**

| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| ALE to data start | $^{t}$ADL | 70 | - | ns |
| ALE hold time | $^{t}$ALH | 5 | - | ns |
| ALE to setup time | $^{t}$ALS | 10 | - | ns |
| CE# hold time | $^{t}$CH | 5 | - | ns |
| CLE hold time | $^{t}$CLH | 5 | - | ns |
| CLE setup time | $^{t}$CLS | 10 | - | ns |
| CE# setup time | $^{t}$CS | 15 | - | ns |
| DATA hold time | $^{t}$DH | 5 | - | ns |
| DATA setup time | $^{t}$DS | 7 | - | ns |
| WRITE cycle time | $^{t}$WC | 20 | - | ns |
| WE# pulse width HIGH | $^{t}$WH | 7 | - | ns |
| WE# pulse width | $^{t}$WP | 10 | - | ns |
| WP# transition to WE# LOW | $^{t}$WW | 100 | - | ns |
| ALE to RE# delay | $^{t}$AR | 10 | - | ns |
| CE# access time | $^{t}$CEA | - | 25 | ns |
| CE# HIGH to output High-Z | $^{t}$CHZ | - | 50 | ns |
| CLE to RE# delay | $^{t}$CLR | 10 | - | ns |
| CE# HIGH to output hold | $^{t}$COH | 15 | - | ns |
| Output High-Z to RE# LOW | $^{t}$IR | 0 | - | ns |
| READ cycle time | $^{t}$RC | 20 | - | ns |
| RE# access time | $^{t}$REA | - | 16 | ns |
| RE# HIGH hold time | $^{t}$REH | 7 | - | ns |
| RE# HIGH to output hold | $^{t}$RHOH | 15 | - | ns |
| RE# HIGH to WE# LOW | $^{t}$RHW | 100 | - | ns |
| RE# HIGH to output High-Z | $^{t}$RHZ | - | 100 | ns |
| RE# LOW to output hold | $^{t}$RLOH | 5 | - | ns |
| RE# pulse width | $^{t}$RP | 10 | - | ns |
| Ready to RE# LOW | $^{t}$RR | 20 | - | ns |
| Reset time (READ/PROGRAM/ERASE) | $^{t}$RST | - | 5/10/500 | us |
| WE# HIGH to busy | $^{t}$WB | - | 100 | ns |
| WE# HIGH to RE# LOW | $^{t}$WHR | 60 | - | ns |

Figure 2-3 and Figure 2-4 illustrates respectively, Command Latch and Address Latch Cycle write sequences.

**Figure 2-3.** Command Latch Cycle

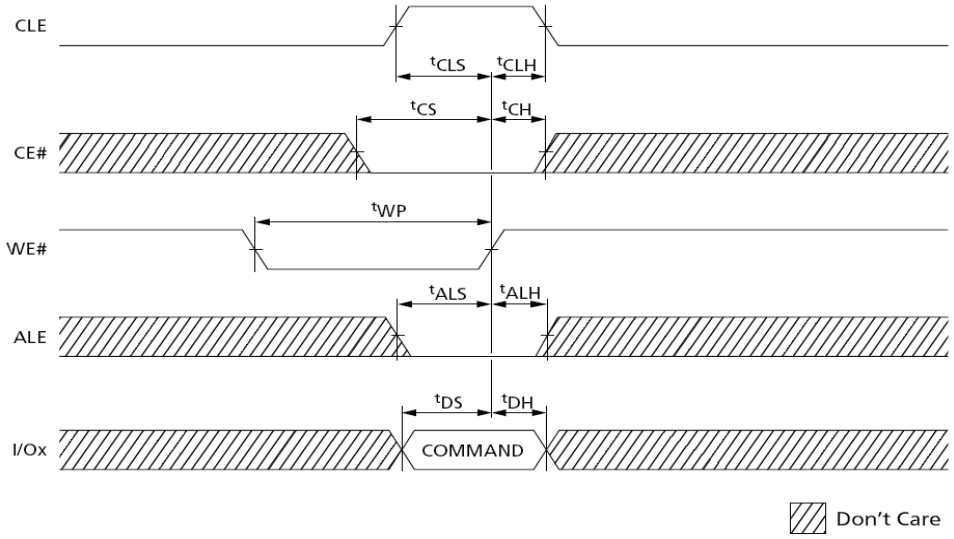



**Figure 2-4.** Address Latch Cycle

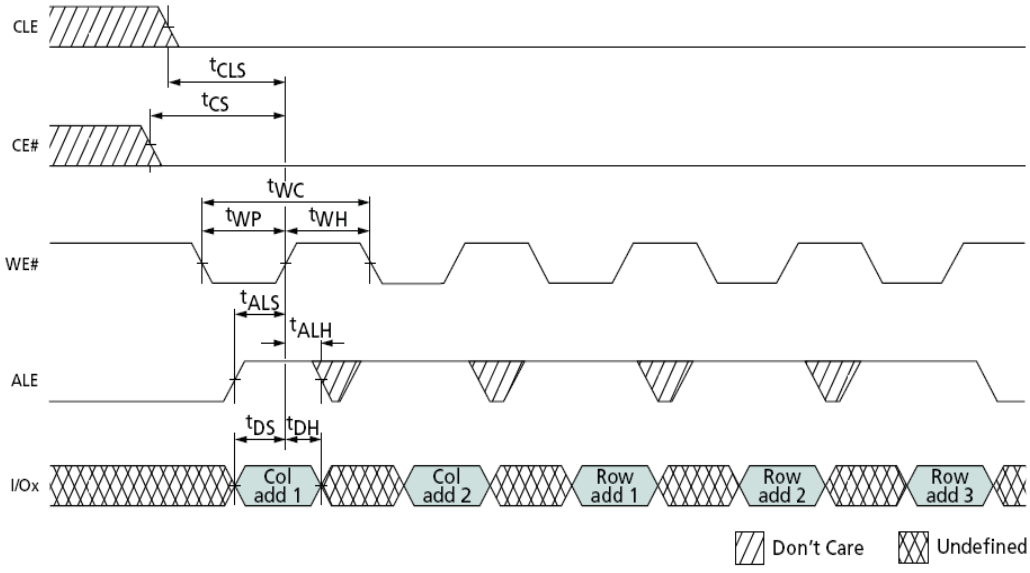


Figure 2-5 and Figure 2-6 illustrates respectively, Data Input Cycle and Data Output Cycle

**Figure 2-5.** Data Input Cycle

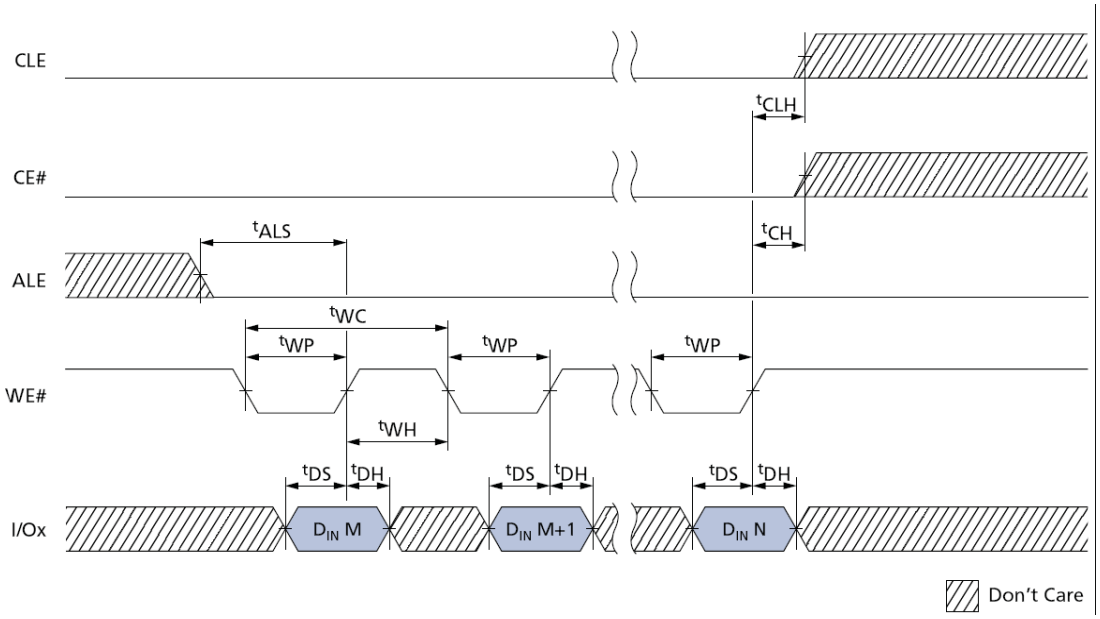



**Figure 2-6.** Data Output Cycle

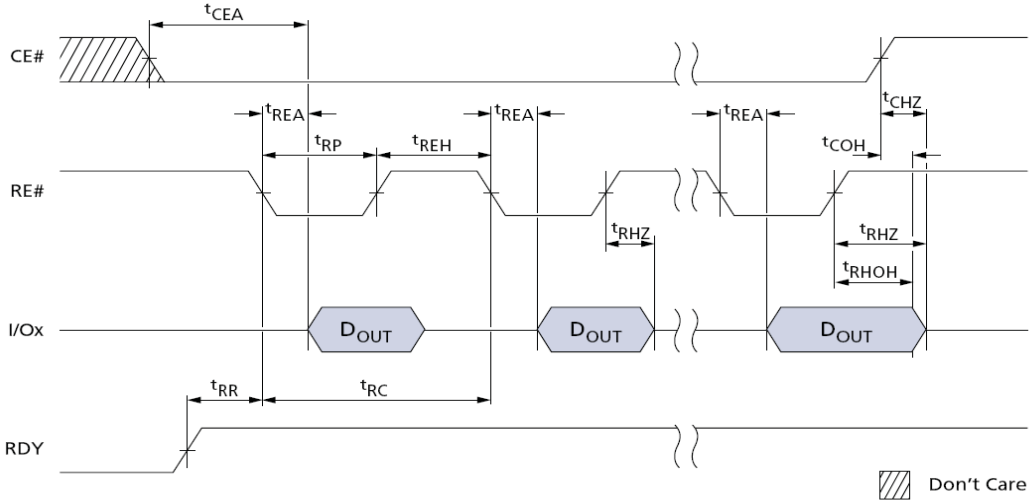
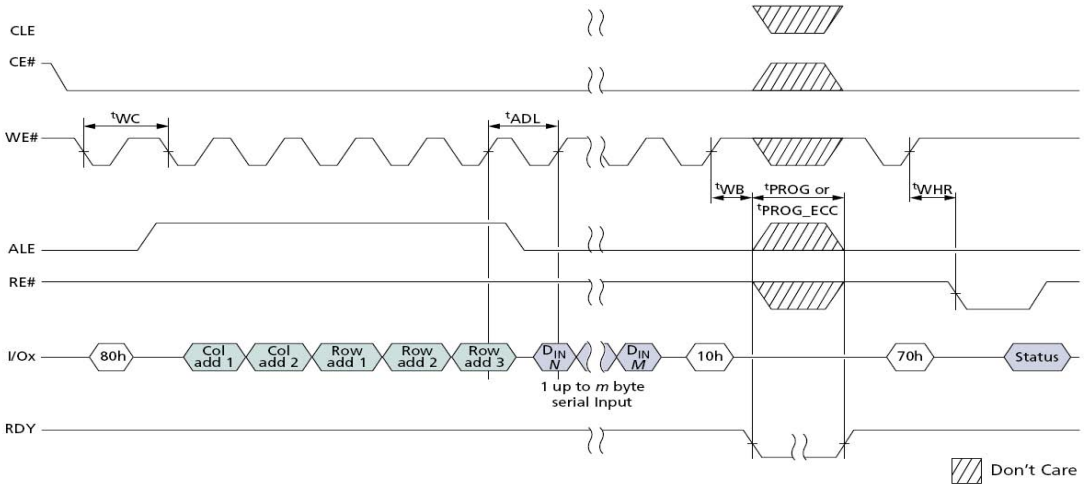


Figure 2-7 illustrate the Page Program sequence

**Figure 2-7.** Program Page Operation



# 3. SAM3X NAND Flash Support

## 3.1 NAND Flash Controller Overview

The SMC embeds a NAND Flash Controller (NFC). The NFC can handle automatic transfers, sending the commands and address cycles to the NAND Flash and transferring the contents of the page (for read and write) to the NFC SRAM; It minimizes the CPU overhead.

### 3.1.1 NFC Controller Register

NAND Flash Read and NAND Flash Program operations can be performed through the NFC

Command Registers. In order to minimize CPU intervention and latency, commands are posted in a command buffer. This buffer provides zero wait state latency. The detailed description of the command encoding scheme is explained below.

The NFC handles automatic transferss between the external NAND Flash and the chip via the NFC SRAM; It is done via NFC Command Registers.

The NFC Command Registers are very efficient to use. When writing to these registers:

- the address of the register (NFCADDR_CMD) contains the command used
- the data of the register (NFCDATA_ADDT) contains the address to be sent to the NAND Flash

That implies that in one single access the command is sent and immediately executed by the NFC. Even two commands can be programmed within a single access (CMD1, CMD2) depending on the VCMD2 value. The NFC can send up to 5 Address cycles.

The NFC Command Registers can be found at address 0x68000000 - 0x6FFFFFFF. Reading the NFC command register (to any address) will give the status of the NFC. Especially useful to know if the NFC is busy, for example.

### 3.1.2 Building NFC Address Command Example

The base address is made of address 0x60000000 + NFCCMD bit set = 0x68000000.

Page read operation example:

```
// Build the Address Command (NFCADDR_CMD)
AddressCommand = (0x60000000   |
                  NFCCMD = 1   | // NFC Command Enable
                  NFCWR = 0    | // NFC Read Data from NAND Flash
                  NFCEN = 1    | // NFC Enable.
                  CSID = 1     | // Chip Select ID = 1
                  ACYCLE = 5   | // Number of address cycle.
                  VCMD2 = 1    | // CMD2 is sent after Address Cycles
                  CMD2 = 0x30  | // CMD2 = 30h
                  CMD1=0x0)      // CMD1 = Read Command = 00h
// Set the Address for Cycle 0
SMC_ADDR = Col. Add1
// Write command with the Address Command built above
*AddressCommand = (Col. Add2 |   // ADDR_CYCLE1
                   Row Add1  |   // ADDR_CYCLE2
                   Row Add2  |   // ADDR_CYCLE3
                   Row Add3)     // ADDR_CYCLE4
```

## 3.2 SMC or NFC I/O Lines Description of SAM3X for NAND Flash

**Table 3-1.** I/O line description for NAND Flash interface of SAM3X-EK

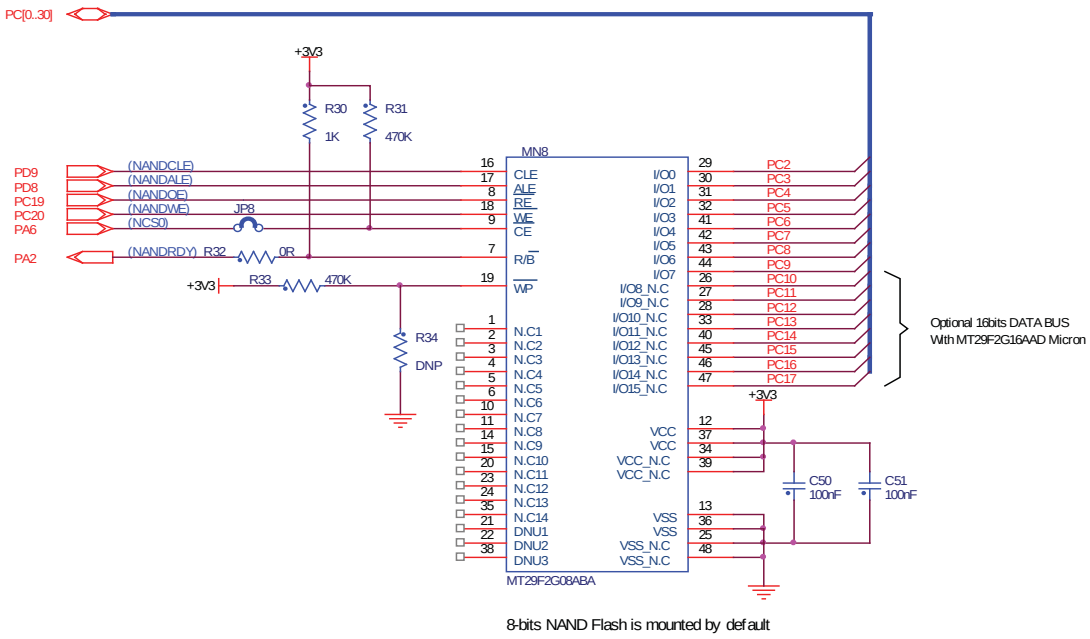
| Name | Description | Type | Active Level | PIO Line Peripheral Multiplex |
|---|---|---|---|---|
| NCS0 | Chip Select Lines | Output | Low | PA6 |
| NANDOE | NAND Flash Output Enable | Output | Low | PC19 |
| NANDWE | NAND Flash Write Enable | Output | Low | PC20 |
| NANDCLE | NAND Flash Command Line Enable | Output | Low | PD9 (A22) |
| NANDALE | NAND Flash Address Line Enable | Output | Low | PD8 (A21) |
| NANDRDY | NAND Flash Ready/Busy | Input | Low | PA2 |
| D[0:15] | NAND Flash Data line | BI | | PC[2:17] |

## 3.3 NAND Flash Connection to SAM3X-EK

The SAM3X features an External Bus Interface (EBI) to offer interface to a wide range of external memories and to any parallel peripheral. The SAM3X-EK board is equipped with EBI with NAND Flash MT29F2G08ABA. The chip select NCS0 for NAND Flash chip select.

The NAND Flash controller drives the read and writes command signals through the SMC or NFC on the NANDOE and NANDWE signals when the NCS0 signal is active.

**Figure 3-1.** SAM3X-EK connection to NAND Flash



## 3.4 SAM3X SMC Controller Read and Write

In order to access the NAND FLASH correctly with the SAM3X SMC Controller, the timing sequence of read and write operations have to be considered carefully.

### 3.4.1 Standard Read and Write waveform

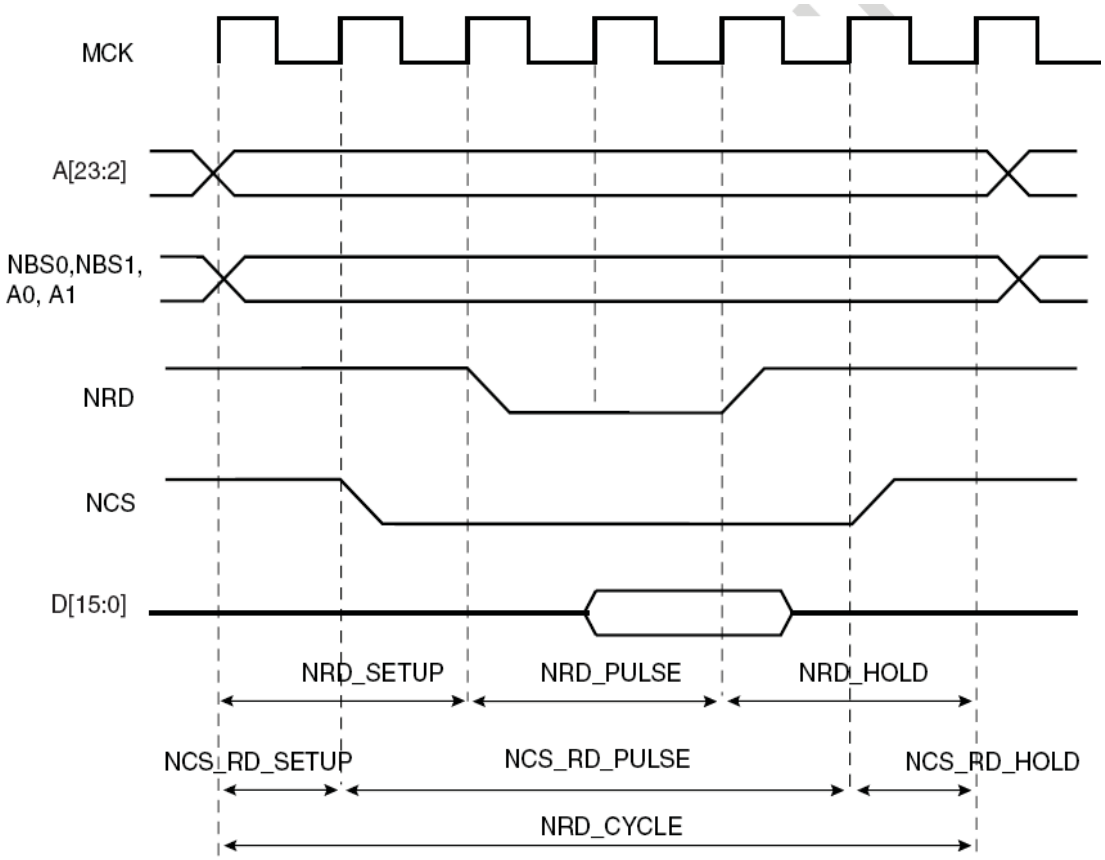
SAM3X SMC controller provides the NRD, NWE, and NCS signals timing parameters register to generate the read and write waveform.

The read cycle is shown on Figure 3-2

The timing of NRD is used for NANDOE signal which control read timing sequence with NAND Flash, the timing of NCS is used for NANDCS signal when access the NAND Flash.

```
The Read Cycle = NRD_SETUP + NRD_PULSE + NRD_HOLD
               = NCS_RD_SETUP +NCS_RD_PULSE + NCS+RD_HOLD
```

**Figure 3-2.** Standard Read Cycle

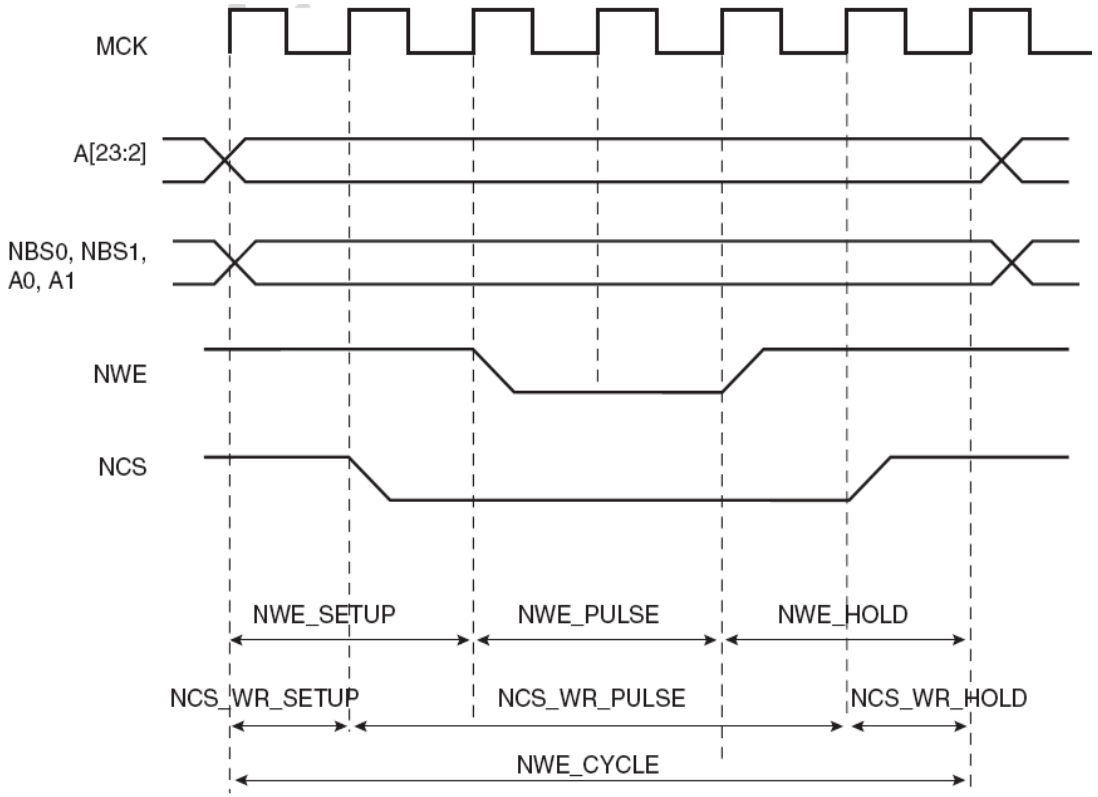


The write cycle is shown on Figure 3-3

The timing of NWE is used for NANDWE signal which control write timing sequence with NAND Flash, the timing of NCS is used for NANDCS signal when access the NAND Flash.

```
The Write Cycle = NWE_SETUP + NWE_PULSE + NWE_HOLD
                = NCS_WR_SETUP +NCS_WR_PULSE + NCS+WR_HOLD
```

**Figure 3-3.** Write Cycle




### 3.4.2 Standard Read and Write Mode

SAM3X supports two kinds of read mode and two kinds of write mode. When READ_MODE is one, the read operation is controlled by NRD signal; otherwise, the read operation is controlled by NCS signal. When WRITE_MODE is one, the write operation is controlled by NEW signal; on the other hand, the write operation is controlled by NCS signal.

## 3.5 SAM3X SMC Controller Register Parameters

To secure transactions with the NAND Flash devices, the SMC controller must be programmed with the appropriate values in accordance with the NAND Flash device timings.

Please note that the following timing parameters calculation only concerns how to decide the SMC timing parameters base on NAND Flash timing specification, it does not optimize the timing to improve the transfer rate, Section 5 gives a description how to optimize the timing sequence.

The first step on order to achieve the SMC settings is to determine what's kind of read and writes mode has to be used. Upon Figure 2-5 and Figure 2-7, NAND flash device use OE and WE to complete read and write operation, it is deduced that the SMC control has to use both READ_MODE = 1 and WRITE_MODE = 1.

The data bus width (DBW) is decided by the specific NAND Flash device type, the MT29F2G08ABA has an 8-bit data width.

It is assumed that the master clock frequency (MCK) of SAM3X system is running at 84MHz, hence, the one MCK cycle is about 12nS. Note: MCK runs on 84MHz and VDDIO is 3.3V

From Figure 2-3, Figure 2-4, Figure 2-5, Figure 2-6, Figure 2-7 and Figure 3-2, Figure 3-3, They can be deduced the timing relationship between MT29F2G08ABA device and SAM3X SMC which is shown on Table 3-2.

**Table 3-2.** MT29F2G08ABA Timing versus SMC Programmable Parameters

| Micron MT29F2G08ABA Timings | Symbol | SMC Related Parameter |
|---|---|---|
| **SMC command and address latch timings** | | |
| CLE Setup Time | $^{t}$CLS | NWE Setup + NWE Pulse |
| ALE Setup Time | $^{t}$ALS | NWE Setup + NWE Pulse |
| CLE Hold Time | $^{t}$CLH | NWE Hold |
| ALE Hold Time | $^{t}$ALH | NWE Hold |
| SMC Read timings - NRD Controlled (READ_MODE = 1) | | |
| Data Setup time (read) | $^{t}$DS | 18 (Data Setup Before NRD) |
| Data Hold time (read) | $^{t}$DH | 0 |
| $\overline{CE}$ Setup Time | $^{t}$CS | NRD setup + NRD pulse - NCS rd setup |
| $\overline{CE}$ Access Time | $^{t}$CEA | $\overline{CE}$ Setup Time - Data Setup time (read) |
| $\overline{RE}$ Access time | $^{t}$REA | MAX 20nS defined in NAND Flash datasheet |
| $\overline{RE}$ High to Output Hi-Z | $^{t}$RHZ | NRD hold + TDF |
| $\overline{CE}$ High to Output Hi-Z | $^{t}$CHZ | NRD hold + TDF |
| $\overline{RE}$ High Hold time | $^{t}$REH | NRD Hold + NRD Setup |
| $\overline{RE}$ Pulse | $^{t}$RP | NRD pulse |

**Table 3-2.** MT29F2G08ABA Timing versus SMC Programmable Parameters

| | | |
|---|---|---|
| Read Cycle Time | $^{t}$RC | NRD Pulse + NRD Setup + NRD Hold |
| SMC Write timings - NWE Controlled (WRITE MODE = 1) | | |
| Data Setup time (write) | $^{t}$DS | Data Out Valid Before NWE High (NWE pulse time - 7) |
| Data Hold time (write) | $^{t}$DH | NWE Hold |
| $\overline{CE}$ Setup Time | $^{t}$CS | NWE setup + NWE pulse - NCS WE setup |
| $\overline{CE}$ Hold Time | $^{t}$CH | NWE Hold - NCS WE Hold |
| $\overline{WE}$ Pulse width | $^{t}$WP | NWE Pulse |
| Write Cycle Time | $^{t}$WC | NEW Pulse + NWE Setup + NWR Hold |

For Read access, the “Data Setup before NRD high” which is given in the SAM3X product data-sheet (refer to the SMC signals in Electrical characteristics section) are respectively equal to

18nS, and from Figure 2-6, in accordance to meet RE Access time requirement (MAX 16nS), therefore the NRD pulse width at least is equal to tDS(SAM3X) + tREA(Nand), if the NRD pulse width is 36nS which meets the MT29F2G08ABA timing requirement.

It is assumed that NRD setup and NRD hold time is minimum one MCK cycle (12nS), and NCS has the same timing with NRD; in Figure 2-6 and Table 3-2 $^{t}$CEA = 36 – 18 = 18nS, the tREH = 24nS; and base on tRHZ and tRHOH in Figure 2-6, the TDF cycle is equal to 24nS.

For Write access, the $^{t}$DS minimal is 7nS, the “Data Out Valid Before NWE High” is given in the SAM3X product datasheet (refer to the SMC signals in Electrical characteristics section) are respectively equal to NWE pulse – 7nS, therefore, NWE pulse need 14nS at least which deduced 2 MCK cycle about 24nS in real setting. It is assumed that NWE setup is minimal one MCK cycle about 12nS, Upon Figure 2-3, Figure 2-4 and Table 3-2, it is can be seen that the $^{t}$CLS and $^{t}$ALS is equal to 36ns. It is assumed that NCS WE setup is minimal one MCK cycle about 12nS, the tCS is 24nS, and $^{t}$CH is 5ns minimal which is decided by NCS WE Hold - NWE Hold, therefore, NCS WR hold time is two MCK cycle minimal and NWE hold time 1 one MCK cycle. The $^{t}$CLH and the $^{t}$ALH is equal to NWE hold time is 24nS.

By far, most of SMC timing parameters have to be considered and meet the MT29F2G08ABA timing specification, beside the above NRD, NCS, NWE timing parameters, The $^{t}$CLR, the $^{t}$ADL, the $^{t}$AR, the tRR and the $^{t}$WB also needs to be configured with an appropriate value given in the MT29F2G08ABA datasheet.

The Table 3-3 summarizes the SMC timing parameter setting.

**Table 3-3.** SMC Programmable timing setting.

| SMC Related Parameter | MCK cycle | nS |
|---|---|---|
| NRD setup | 1 | 12 |
| NRD pulse | 3 | 36 |
| NRD hold | 1 | 12 |
| NCS RD setup | 1 | 12 |
| NCS RD pulse | 3 | 36 |
| NCS RD hold | 1 | 12 |
| NWE setup | 1 | 1 |
| NWE pulse | 2 | 24 |
| NWE hold | 2 | 24 |
| NCS WR setup | 1 | 1 |
| NCS WR pulse | 3 | 36 |
| NCS WR hold | 1 | 12 |
| $^{t}$CLR | 1 | 12 |
| $^{t}$ADL | 7 | 84 |
| $^{t}$AR | 1 | 12 |
| $^{t}$RR | 2 | 24 |
| $^{t}$WB | 5 | 60 |

Figure 3-4 and Figure 3-5 are shown the read and write waveform base on SMC timing settings.

MCK = 84MHz $T_{mck}$ = 1/MCK = 12nS

**Figure 3-4.** Read access waveform

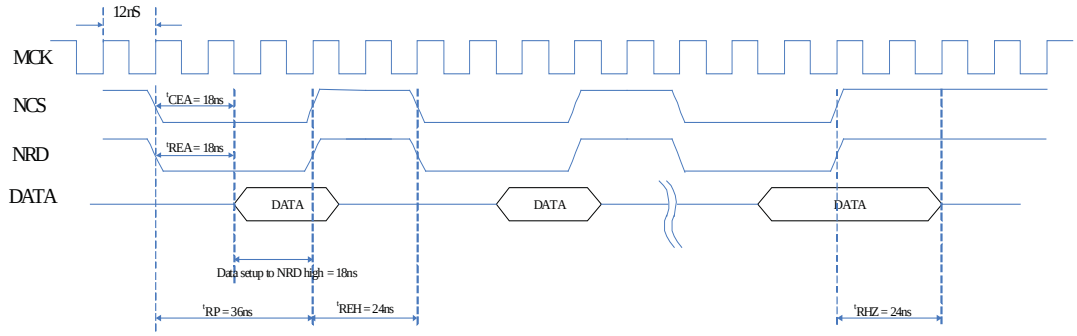
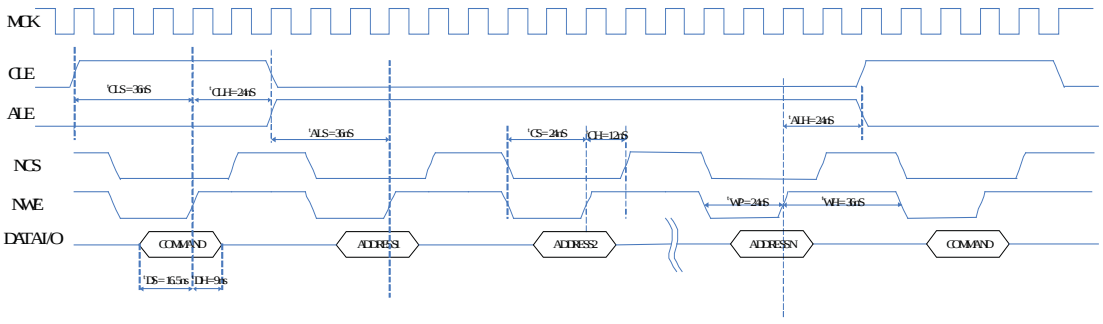



**Figure 3-5.** Write access waveform



# 4. Invalid Block Management and Error Correcting Code (ECC)

## 4.1 Error Management on NAND FLASH device

The NAND Flash device is specified to have the minimum number of valid blocks (NVB) of the total available blocks. This means that the devices may have blocks that are invalid when shipped from the factory.

An invalid block is one that contains at least one page that has more bad bits than can be corrected by the minimum required ECC. Additional bad blocks may develop during use. However, the total number of available blocks will not fall below $N_{VB}$ during the life endurance of the product.

The minimum number of Valid blocks (NVB) is 2008 blocks of 2048 total available blocks for MT29F2G08ABA.

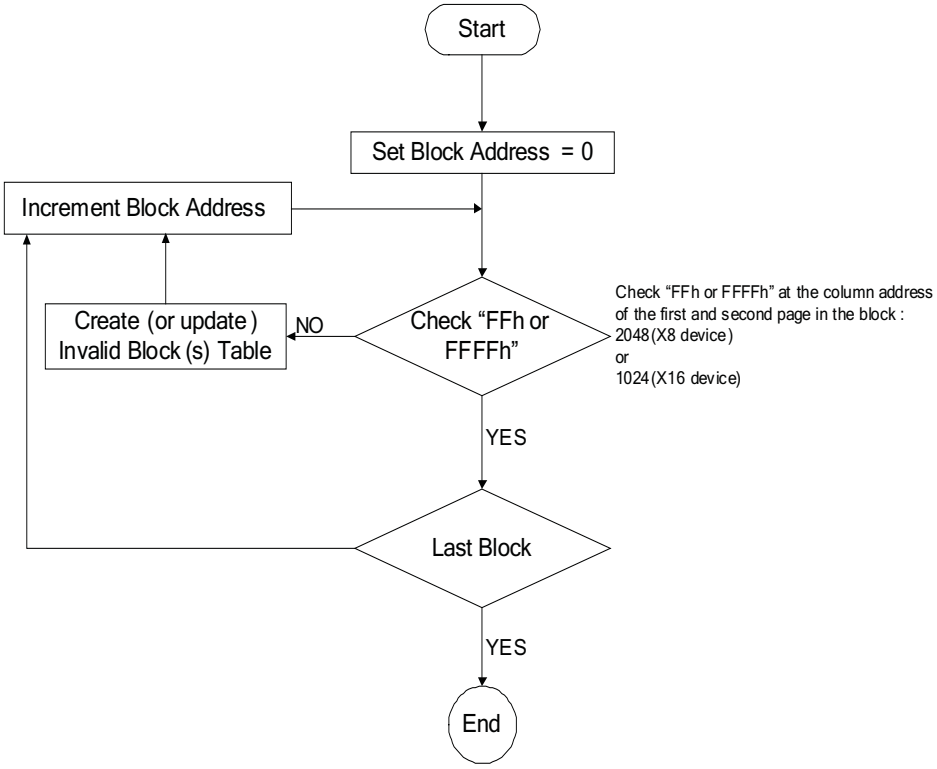
## 4.2 Invalid Block identification

Before shipping, every NAND flash device is tested with specific test patterns under different voltage and temperature conditions in order to identify memory locations containing errors. When errors are detected, the block to which the invalid memory location belongs is marked as an “Invalid Block”.

NAND Flash devices are shipped from the factory erased. The factory identifies invalid blocks before shipping by attempting to program the bad-block mark into every location in the first page of each invalid block. The first spare area location in each bad block is guaranteed to contain the bad-block mark. It may not be possible to recover the bad-block marking if the block is erased. Therefore, the system software should initially check the first spare area location for non-FFH data on the first page of each block prior to performing any program or erase operations on the NAND Flash device.

Figure 4-1 describes how to manage the invalid block by software.

**Figure 4-1.** Bad Block Identification Flow Chart



## 4.3 Error Correcting Code (ECC) in SAM3X SMC controller

Over time, some memory locations may fail to program or erase properly. In order to ensure that data is stored properly over the life of the NAND Flash device, NAND Flash providers recommend using an Error Correcting Code to guarantee the data integrity.

The SAM3X supports single bit error correction and two bit error detection per 256 byte, per 512 byte of date or per full page (528/1056/2112/4224).

24-bit ECC is generated in order to perform one bit correction per 256 or 512 bytes for pages of 512/2048/4096 8-bit words.

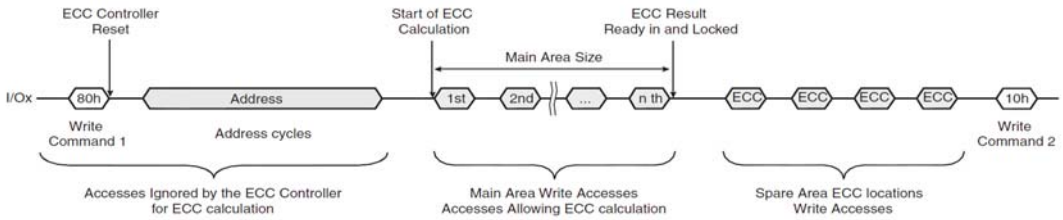
32-bit ECC is generated in order to perform one bit correction per 512/1024/2048/4096 8- or 16-bit words

There is a limitation that only 1 ECC for all pages is possible when using 16-bit NAND flash.

### 4.3.1 Page Write Sequence

- The ECC is automatically reset as soon as the first write command (80h) is issued to the NAND Flash
- The ECC calculation starts only once the target address is issued to NAND Flash
- The ECC is refreshed at each write access of the page until last byte or half word of the main area is written
- Once the whole main area has been written, the final ECC result is available in the ECC Parity (ECC_PR) Register and ECC NParity Register until a new write or read occurs
- The software application has to write the Parity ECC and NParity ECC in the appropriate locations of the device spare area

**Figure 4-2.** ECC Calculation During Page Write Sequence.




### 4.3.2 Page Read Sequence

- The ECC controller is automatically reset as soon as the first read command (00h) is performed to the NAND Flash
- The ECC calculation starts only once the required address cycles and the second read command (30h) is performed to the NAND Flash
- The ECC is refreshed at each read access of the page until the last byte or half word of the main area is read

- Since Parity ECC and NParity have been previously stored in locations of the spare area which are not contiguous to the main area, it is useful to perform a random read command sequence before performing the ECC data accesses

**Figure 4-3.** ECC Error Detection during Page Read with Random Read Spare Area

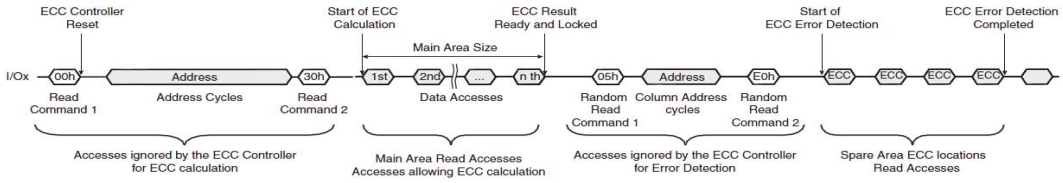
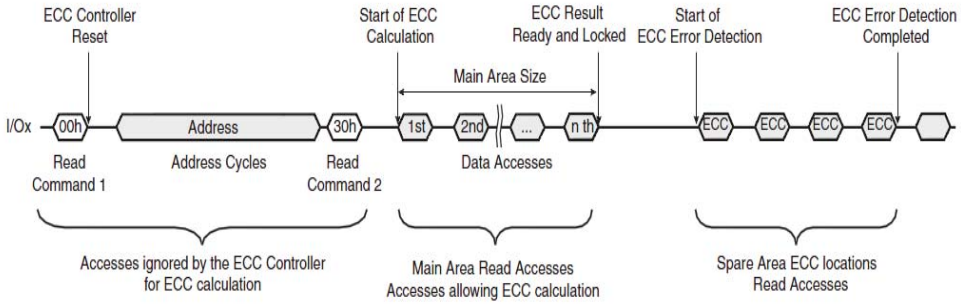



**Figure 4-4.** ECC Error Detection during Page Read without Random Read Spare Area




After reading the whole data in the main area, the application must perform read accesses to the extra area where ECC code has been previously stored. Error detection is automatically performed by the ECC controller.

The application can check the ECC Status Registers for any detected errors. It is up to the application to correct any detected error. ECC computation can detect four different circumstances.

- No error: XOR between the ECC computation and the ECC code stored at the end of the NAND Flash is equal to 0
- Recoverable error: Only the RECERR flags in the ECC Status registers are set. The corrupted word offset in the read page is defined by the WORDADDR field in the ECC Parity Registers. The corrupted bit position in the concerned word is defined in the BITADDR field in the ECC Parity Registers.
- ECC error: The ECCERR flag in the ECC Status Registers is set. An error has been detected in the ECC code stored in the Flash memory. The position of the corrupted bit can be found by the application performing an XOR between the Parity and the NParity contained in the ECC code stored in the Flash memory
- Non correctable error: The MULERR flag in the ECC Status Registers is set. Several unrecoverable errors have been detected in the Flash memory page

Please note that it is mandatory to read consecutively the entire main area and the locations where Parity and NParity values have been previously stored to let the ECC controller perform error detection.

# 5. Optimize NAND FLASH Device Transfer rate

## 5.1 How to optimize the timing parameters

To maximize the NAND Flash device transfer rate, the Read and Write timing parameters have to be optimized to reduce read and write cycles.

### 5.1.1 The Read timing parameters optimize

Referring to Figure 3-4, in order to reduce read cycle, $^{t}$RP and $^{t}$REH have to be reduced by improving timing parameter $^{t}$RP = NRD pulse = $^{t}$REA + Data Setup time. Data Setup time is given by 18nS on SAM3X datasheet, therefore, $^{t}$REA NRD pulse could be 24nS (2 MCK cycle), $^{t}$REA is reduce 6nS, and it also meet the MT29F2G08ABA specification.

$^{t}$REH minimal is 7nS, In Figure 3-4, $^{t}$REH is 24nS with NRD hold time plus NRD Setup time, it can be improved by set NRD hold is equal to 0, only using next NRD setup time as $^{t}$REH.

Now, the optimized timing parameters is given by above discussion, NRD setup is equal to12nS (1MCK cycle), NRD pulse time is equal to 24nS (2 MCK cycle) and NRD hold time is equal to 0.

The total read cycle is reduce to 36nS.

### 5.1.2 The Write timing parameters optimize

Referring to Figure 3-5, in order to reduce read cycle, $^{t}$WP and $^{t}$WH have to be reduced by improving the following timing parameters; the $^{t}$DS minimal is7nS, the NWE pulse time minimal is equal to $^{t}DS_{min}$ + 7nS as 14nS, so NWE pulse time need 24nS (2MCK), $^{t}$WH minimal is 7nS; In Figure 3-5, $^{t}$WH is 36nS with NWE hold time plus NWE Setup time, it can be improved by setting NWE hold equal to 0, only using next NWE setup time as $^{t}$WH.

Now, the optimized timing parameters are given by the above directions, NWE setup is equal to12nS (1MCK cycle), NWE pulse time is equal to 24nS (2 MCK cycle) and NWE hold time is equal to 0. The total read cycle is reduced to 36nS.

## 5.2 The maximum transfer rate

Figure 5-1 show the maximum transfer rate between SAM3X and NAND flash MT29F2G08ABA, the transfer period is defined from issuing a read or write command to get the full page data in NFC SRAM.

The test code using SAM3X soft package “examples_storage smc_nandflash” compiled with IAR5.5

**Table 5-1.** NAND Flash Maximum transfer rate in SAM3X-EK

| Test Mode | Transfer Rate | Test Condition | |
|---|---|---|---|
| Read | 11.64MBytes/S | NRD pulse = 24ns, NRD cycle = 36ns | tWB=60nS<br>tRR=24nS<br>tAR=12nS<br>tADL=72nS<br>tCLR = 12nS |
| Write | 6.1MBytes/s | NWE pulse = 24ns, NWE cycle = 36ns | |

# 6. How to initialize the transfers between SAM3X and NAND Flash

## 6.1 NAND Flash controller Configuration steps

In order to access the external NAND Flash device, the PMC, PIO and SMC have to be configured with the right settings prior to actual NAND flash operations.

For detailed source code, please refer to the SAM3X software package “examples_storage smc_nandflash”

Shown below are the configuring steps being followed in the SAM3X software package.

- PMC configuration
  - Configure PLL output frequency
  - Configure Processor (PCK) / Master Clock (MCK)
  - Enable SMC peripheral clock
  - Enable PIO peripheral clock

- SMC controller configuration
  - Configure EBI chip select signal
  - Configure Byte Access Type
  - Configure Data bus Width
  - Configure Read and Write Mode
  - Set Read, Write and Chip select Setup time, and Hold time and Clock cycle
  - Configure SMC timing parameters for NAND flash device
  - Enable embedded NFC controller

- PIO controller configuration
  - Configure PIO line function of SMC interface

The SMC configuration code of SAM3X soft package is listed below.

```
/* Enable peripheral clock */
PMC_EnablePeripheral( ID_SMC ) ;

/* Configure the SMC timing register*/
SMC->SMC_CS_NUMBER[0].SMC_SETUP = SMC_SETUP_NWE_SETUP(0)
                                | SMC_SETUP_NCS_WR_SETUP(0)
                                | SMC_SETUP_NRD_SETUP(0)
                                | SMC_SETUP_NCS_RD_SETUP(0);

SMC->SMC_CS_NUMBER[0].SMC_PULSE = SMC_PULSE_NWE_PULSE(2)
                                | SMC_PULSE_NCS_WR_PULSE(3)
                                | SMC_PULSE_NRD_PULSE(2)
                                | SMC_PULSE_NCS_RD_PULSE(3);

SMC->SMC_CS_NUMBER[0].SMC_CYCLE = SMC_CYCLE_NWE_CYCLE(3)
                                | SMC_CYCLE_NRD_CYCLE(3);

SMC->SMC_CS_NUMBER[0].SMC_TIMINGS = SMC_TIMINGS_TCLR(1)
                                  | SMC_TIMINGS_TADL(6)
                                  | SMC_TIMINGS_TAR(4)
                                  | SMC_TIMINGS_TRR(2)
                                  | SMC_TIMINGS_TWB(9)
                                  | SMC_TIMINGS_RBNSEL(7)
                                  |(SMC_TIMINGS_NFSEL);


/*Configure SMC R/W mode and Data bus width*/
SMC->SMC_CS_NUMBER[0].SMC_MODE = SMC_MODE_READ_MODE
                               | SMC_MODE_WRITE_MODE
                               | SMC_MODE_DBW_BIT_8;
```

## 6.2 NAND Flash Initialize steps

In order to access the external NAND Flash device correctly, the NAND Flash device has to be initialized to get NAND Flash information prior to actual NAND flash operations.

Shown below are the configuring steps being followed in the SAM3X software package

- NAND Flash device initialization
  - NAND Flash Reset
  - Read NAND Flash ID
  - Get NAND Flash device information
  - Retrieve block number, page/block size, bus width from device
  - Configure PAGESIZE in NFC Configuration register
  - Re-configure Data bus width base on NAND flash device information

- Bad block table creation
  - Check first spare area location for non-FFH data on the first page of each block prior to performing any program or erase operations to create bad block table

The Nand Flash initialization code of SAM3X soft package is listed below.

```
/* Reset*/
RawNandFlash_Reset(raw);

/* If model is not provided, autodetect it*/
if (!model) {

    TRACE_DEBUG("No model provided, trying autodetection ...\n\r");
    if (NandFlashModel_Find(nandFlashModelList,
                            NandFlashModelList_SIZE,
                            RawNandFlash_ReadId(raw),
                            &(raw->model))) {

        TRACE_ERROR("RawNandFlash_Initialize:Couldnotautodetect
        chip.\n\r");

        return NandCommon_ERROR_UNKNOWNMODEL;
    }
}
else {

    /* Copy provided model*/
    raw->model = *model;
}
```

The Nand Flash Bad Block creation code of SAM3X soft package is listed below.

```
      /* Retrieve model information */
        numBlocks =
NandFlashModel_GetDeviceSizeInBlocks(MODEL(skipBlock));

      /* Initialize block statuses */
        TRACE_DEBUG(“Retrieving bad block information ...\n\r");

        /* Retrieve block status from their first page spare area */
        for (block = 0; block < numBlocks; block++) {

            /* Read spare of first page */
            error = SkipBlockNandFlash_CheckBlock(skipBlock, block);

            if (error != GOODBLOCK) {

                if (error == BADBLOCK) {

                    TRACE_DEBUG("Block #%d is bad\n\r", block);
                }
                else {

                    TRACE_ERROR(
                              "SkipBlockNandFlash_Initialize: Cannot
                              retrieve info from block #%u\n\r",
                              block);
                }
            }
        }
```

# 7. Revision History

**Table 7-1.**

| Document | Comments | Change Request Ref. |
|---|---|---|
| 11154A | Initial version. | |

## Headquarters


***Atmel Corporation***
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600


## International


***Atmel Asia Limited***
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

***Atmel Munich GmbH***
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

***Atmel Japan***
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581


## Product Contact

***Web Site***
www.atmel.com
www.atmel.com/AT91SAM

***Technical Support***
AT91SAM Support
Atmel technical support

***Sales Contacts***
www.atmel.com/contacts/

***Literature Requests***
www.atmel.com/literature


**Disclaimer:** The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

**© 2012 Atmel Corporation. All rights reserved.** Atmel®, Atmel logo and combinations thereof, QTouch®, DataFlash®, SAM-BA® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® and others, are registered trademarks or trademarks of Microsoft Corporation in the US and/or other countries. ARM®, ARM®Powered logo, Cortex®, Thumb®-2 and others are registered trademarks or trademarks of ARM Ltd. Other terms and product names may be trademarks of others.