NAND Flash Support in AT91SAM9 Microcontrollers

1. Scope

The purpose of this application note is to introduce the NAND Flash technology and to describe how to interface NAND Flash memory to Atmel® AT91SAM9 ARM® Thumb®-based Microcontrollers that do not feature a NAND Flash Controller. The NAND Flash logic is driven by the Static Memory Controller on the NCS3 address space.

Sample code is provided the associated zip file, *Basic NAND Source Code.zip*; the source code is based on the product libV3.

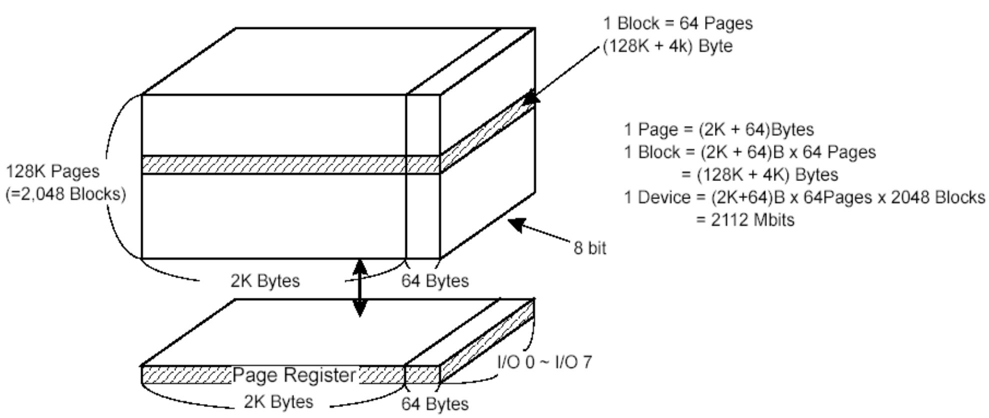
2. NAND Flash Overview

2.1 General Overview

NAND Flash provides a cost effective alternative to hard drives, especially for portable and handheld systems. The performance, pricing, and memory size options make it optimal for storage applications (pictures, audio files, etc.).

The NAND Flash used to illustrate this interface is the K9F2G08U0M, manufactured by Samsung® Electronics. Figure 2-1 shows the memory organization of this device.

Figure 2-1. K9F2G08U0M Organization



AT91 ARM Thumb Microcontrollers

Application Note

6255B–ATARM–26-Jun-09

2.2 NAND Flash vs. NOR Flash

The most important item for memories is the cost per bit which depends on memory cell area per bit. The cell area of NAND Flash is smaller than that of NOR Flash, making the NAND Flash more cost effective than NOR Flash.

The first significant difference between NAND and NOR Flash is the hardware interface.

2.2.1 Hardware Interface

NOR Flash has a fully memory-mapped random access interface similar to a RAM, with dedicated address lines and data lines making it "bootable".

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.

NOR Flash random-access interface typically composed of 41 pins:

- CE# - chip enable
- WE# - write enable
- OE# - output enable
- D[15:0] - data bus
- A[20:0] - address bus
- WP# - write protect

NAND Flash I/O device-type interface composed of up to 24 pins:

- CE# - chip enable
- WE# - write enable
- RE# - read enable
- CLE - command latch enable
- ALE - address latch enable
- I/O[7:0] or I/O[15:0] - data bus
- WP# - write protect
- R/B# - ready / busy
- RE - Read enable

2.2.2 Array Architecture

NOR Flash is divided into blocks which typically contain many 16-bit wide words. Random access to stored data words is achieved by placing the selected word address on the address bus and then reading that data off the data bus. Erase operations are managed at the block level and words can be programmed after a block has been erased.

NAND Flash is also divided into blocks which contain many pages instead of words (2K +64 bytes). Read and program operations take place on a per-page basis whereas erase operations takes place on a block basis.

To read or write from NAND Flash, a command sequence is issued to select a block and a page. After this selection, the entire page can be read or written.

NAND Flash typically contains blocks that contain errors and cannot be used. A check must be done by software to list and maintain a table of bad blocks. Data integrity is achieved by using hardware or software techniques, such as ECC, that check and correct bad data.

2.2.3 Performances

Further differences between NOR and NAND Flash can be found in read/write performances. Table 2-1 shows random access time for NOR Flash specified at 0.09 µs, whereas NAND random access is significantly slower — 25 µs — for the first byte. Once the initial access has been made, however, the remaining 2,111 bytes are shifted out of NAND at only 30 ns per byte. This results in a bandwidth of more than 23 Mb/s for 8-bit I/Os or 37 Mb/s for 16-bit I/Os.

The real benefits for NAND Flash can be found in the faster program and erase times, since NAND provides over five megabytes per second of sustained write performance. The block erase times are an impressive 2 ms for NAND versus 200 ms for NOR.

Table 2-1. Differences in Performance

Characteristics	NAND Flash K9F2G08U0M	NOR Flash AT49BV16x4-90
Random access read	25 µs (first byte) 30 ns each for remaining 2111 bytes	0.09 µs
Sustained read speed (sector basis)	37 Mbytes/s	11 Mbytes/s
Random write speed	300 µs/2,112 bytes	20 µs / bytes
Sustained write speed (sector basis)	5 Mbytes/s	0.05 Mbytes/s
Erase block size	128 Kbytes	64 Kbytes
Erase cycles	100,000 to 1,000,000	10,000 to 100,000
Erase time per block	2 ms	200 ms

2.2.4 Conclusion

Table 2-2 summarizes NAND/NOR advantages and disadvantages.

Table 2-2. NAND/NOR Comparison

	NAND	NOR
Advantages	Fast writes Fast erases Lower bit cost Higher density	Random access Byte writes possible
Disadvantages	Slow random access Byte writes difficult Bad blocks management and ECC required	Slow writes Slow erase
Applications	File (disk) applications Voice, data, video recorder Any large sequential data	Execute directly from non volatile memory

Clearly, NAND Flash has several significant positive attributes. The one negative attribute is that it is not well-suited for direct random access.

NAND is available in large capacities and is the lowest cost Flash memory available today. NAND is used in virtually all removable cards for cost/density reasons: USB Cards, Memory Stick, MMC Multimedia Card, SD Secure Digital, CF Compact Flash.

3. Bad Block Management and Error Corrected Code (ECC)

3.1 Definition of “Bad Block”

By default, NAND devices contain invalid blocks which have one or more invalid bits.

Furthermore, since the first memory block (physical block address 00h) in NAND devices is guaranteed to be free of defects (up to 1,000 PROGRAM/ERASE cycles), the first 8 Kb of Flash memory can safely be used for system bootstrapping functions.

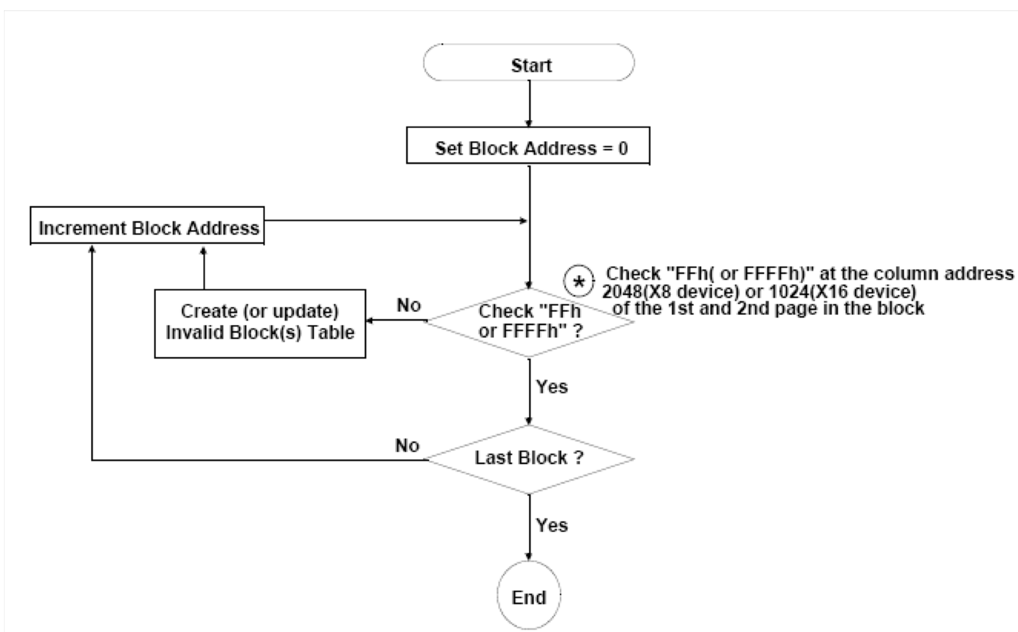
3.2 Software Considerations

To avoid writing to and reading from bad memory blocks, system software must create a map of invalid memory blocks. If the application code executes from RAM rather than Flash memory, system software bad-block mapping is only necessary at boot time and during Flash storage updates.

All device locations are erased (FFh for X8, FFFFh for X16) except locations where the invalid block information is written prior to shipping. The invalid block status is defined by the 1st byte (X8 device) or 1st word (X16 device) in the spare area.

The 1st or 2nd page of every invalid block has non-FFh(X8) or non-FFFFh(X16) data at the column address of 2048 (X8 device) or 1024 (X16 device). Since the invalid block information is also erasable in most cases, it is impossible to recover the information once it has been erased. Therefore, the system must be able to recognize the invalid block(s) based on the original invalid block information and create the invalid block table via the flow chart in Figure 3-1.

Figure 3-1. Bad Block Recognition Flow Chart



Important Note: Any intentional erasure of the original invalid block information is prohibited.

3.3 ECC

NAND devices are subject to data failures that occur during device operation. To ensure data read/write integrity, system error-checking and correction (ECC) algorithms must be implemented. Depending on the AT91 product, the ECC algorithm must be calculated by software or can be generated by the embedded hardware ECC controller. The ECC controller is capable of single bit error correction and 2-bit random detection. When NAND has more than 2 bits of errors, the data cannot be corrected. This controller allows ECC management without CPU intervention and thus improves the total bandwidth of the system.

4. NAND Flash Signals

4.1 Bus Operation

The bus on NAND Flash devices is internally multiplexed. Data I/O, addresses, and commands all share the same pins. I/O pins. I/O[15:8] are used only for data in the x16 configuration. Addresses and commands are always supplied on I/O[7:0].

The command sequence normally consists of a command latch cycle, an ADDRESS LATCH cycle, and a DATA cycle — either READ or WRITE.

4.2 Control Signals

The signals CE#, WE#, RE#, CLE and ALE control Flash device READ and WRITE operations.

CE# is used to enable the device. When CE# is low and the device is not in the busy state, the Flash memory accepts command, data, and address information.

When the device is not performing an operation, the CE# pin is typically driven HIGH and the device enters standby mode. The memory enters standby if CE# goes HIGH while data is being transferred and the device is not busy.

A subset of NAND Flash supports the CE# “Don’t Care” operation allowing the NAND Flash to reside on the same asynchronous memory bus as other Flash or SRAM devices. Other devices on the memory bus can then be accessed while the NAND Flash is busy with internal operations. This capability is important for designs that require multiple memory devices on the same bus.

4.3 Commands (ALE = 0, CLE = 1)

All the NAND operations (except READ STATUS and RESET commands) consist of a command write cycle followed by address write cycle. The READ STATUS command does not have an address write cycle. The command is transferred into the NAND command register followed by the start address, for the read or program operation, latched into the address register.

Commands are written to the command register on the rising edge of WE# when:

- CE# and ALE are low
- CLE is high

Commands are input on I/O[7:0] only. For devices with a x16 interface, I/O[15:8] must be written with zeros when issuing a command.

4.4 Address (ALE = 1, CLE = 0)

Addresses are written to the address register on the rising edge of WE# when:

- CE# and CLE are low
- ALE is high

Addresses are input on I/O[7:0] only. For devices with a x16 interface, I/O[15:8] must be written with zeros when issuing an address. Generally all five ADDRESS cycles are written to the device.

4.5 Data (ALE = 0, CLE = 0)

Data is written to the data register on the rising edge of WE# when CE#, CLE, and ALE are low. Data is input on I/O[7:0] for x8 devices, and I/O[15:0] on x16 devices.

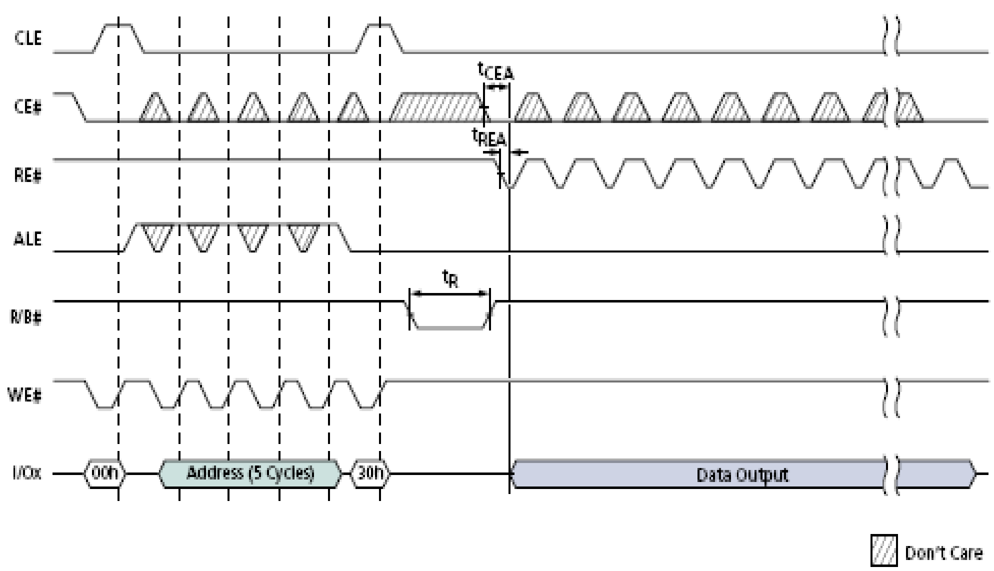
4.6 Ready / Busy

The R/B# output provides a hardware method of indicating the completion of a PROGRAM/ERASE/READ operation. The signal is typically high, and transitions to low after the appropriate command is written to the device. A dedicated PIO should be assigned to this signal with a pull-up resistor for proper operation. Alternatively, the READ STATUS command can be used by the software.

4.7 Example

The following waveforms shows the successive accesses: COMMAND Latch, ADDRESS Latch and DATA Output with a “CE don’t Care” NAND. Notice that no command can be sent to the NAND Flash during tR, because it is busy.

Figure 4-1. Page READ Operation



5. AT91 EBI NandFlash Logic

The NAND Flash logic is driven by the Static Memory Controller (SMC) on the NCS3 address space. Programming the CS3A field in the EBI_CSA Register in the Bus Matrix User Interface to the appropriate value enables the NAND Flash logic. For details on this register, refer to the Bus Matrix User Interface section in the product datasheet. Access to an external NAND Flash device is then made by accessing the address space reserved to NCS3 (i.e., between 0x40000000 and 0x4FFF FFFF).

The NAND Flash Logic drives the read and write command signals of the SMC on the NANDOE and NANDWE signals when the NCS3 signal is active.(refer to the Static Memory Controller section in the product datasheet).

The address latch enable and command latch enable signals on the NAND Flash device are driven by address bits A22 and A21 of the EBI address bus.The command, address or data words on the data bus of the NAND Flash device are distinguished by using their address within the NCS3 address space. The chip enable (CE) signal of the device and the ready/busy (R/B) signals are connected to PIO lines.

Two NAND Flash types exists, those who are “CE don’t care” and those who are not.

For “CE don’t care” NAND, the chip enable state is don’t care during the busy period preceding the data read cycle. Thus allowing this flash to be connected to active memory buses such as the AT91 memory bus.

For standard NAND, the CE signal remains asserted even when NCS3 is not selected, preventing the device from returning to standby mode. In this case, a PIO line should be dedicated to drive the Chip Enable signal.

Unlike the AT91SAM9261, where A21 and A22 are not specifically dedicated to NAND flash ALE and CLE signals, on the AT91SAM9260 A21/ALE and A22/CLE are the signals mandatory to drive the NAND flash and ECC controller. Another combination of addresses prevent using the ECC controller with the NAND Flash.

Table 5-1. EBI Signals Example for AT91SAM9261

Name	Function	Type	Active Level
NANDCS	NAND Flash Chip Select Line	Output	Low
NANDOE	NAND Flash Output Enable	Output	Low
NANDWE	NAND Flash Write Enable	Output	Low
CLE(A21) (3)	Command Latch Enable	Output	High
ALE(A22) (4)	Address Latch Enable	Output	High
PIOx/CE	Chip Enable (1)(2)	Output	Low
PIOy/RDY/BSY	Ready/Busy (1)	Input	Low

Notes:
1. Any free PIO can be used for this purpose
2. For standard NAND
3. For AT91SAM9261 the address bit A21 is arbitrarily dedicated to CLE
4. For AT91SAM9261 the address bit A22 is arbitrarily dedicated to ALE

Table 5-2. EBI Signals Example for AT91SAM9260

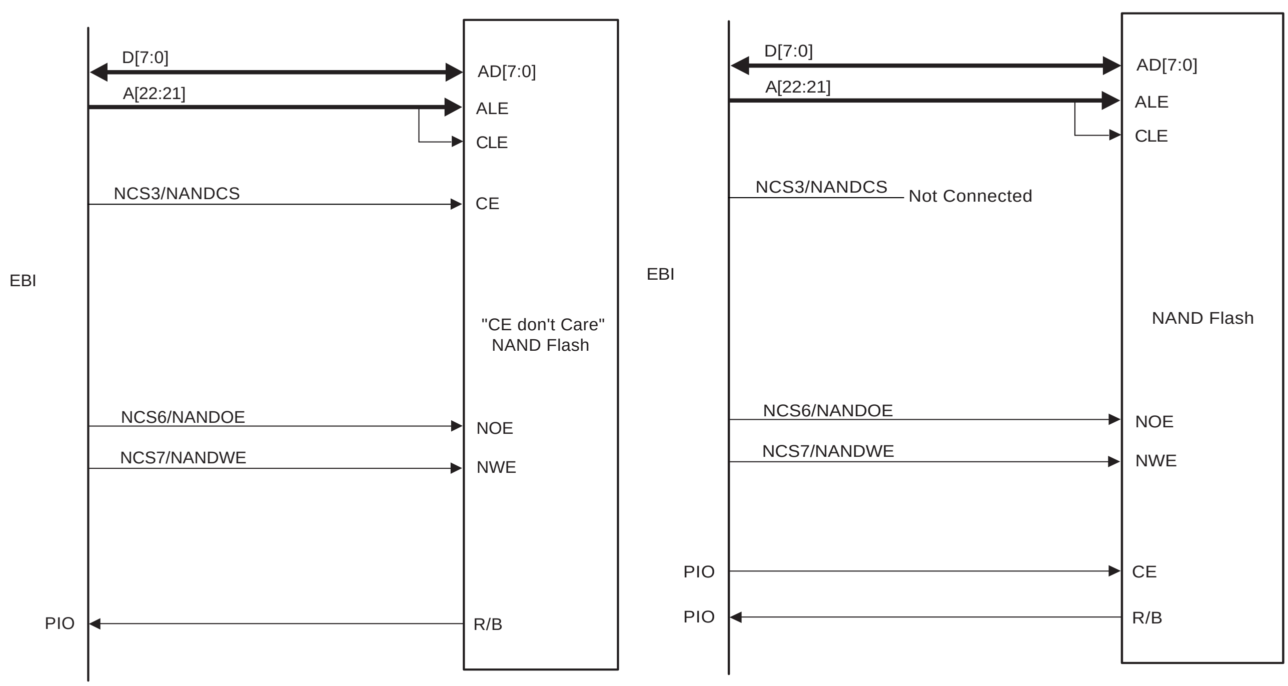
Name	Function	Type	Active Level
NANDCS	NAND Flash Chip Select Line	Output	Low
NANDOE	NAND Flash Output Enable	Output	Low
NANDWE	NAND Flash Write Enable	Output	Low
A22/CLE	Command Latch Enable	Output	High
A21/ALE	Address Latch Enable	Output	High
PIOx/CE	Chip Enable (1)(2)	Output	Low
PIOy/RDY/BSY	Ready/Busy (1)	Input	Low

Notes: 1. Any free PIO can be used for this purpose
2. For standard NAND

Table 5-3. ALE/CLE Management

ALE	CLE	AT91SAM9261 Memory Address Offset	AT91SAM9260 Memory Address Offset	NAND Register Selected
0	0	0x000000	0x000000	DATA register
0	1	0x200000	0x400000	COMMAND register
1	0	0x400000	0x200000	ADDRESS register
1	1	0x600000	0x600000	Undefined (Don't use)

Figure 5-1. "CE don't care" and Standard NAND Flash Application Example



6. AT91 System Initialization for a K9F2G08U0M Device

6.1 Clocks

The system is running at full speed, this means 198 MHz for the processor and 99 MHz for the Bus. The EBI NCS3 is to be assigned for NAND Flash usage.

Table 6-1. System Configuration

Description	Settings	Register/field	Value
System			
PLL frequency	**198 MHz**	**PMC_PLLAR**	**0x20603F09**
Processor / Bus Clock	**198 / 99 MHz**	**PMC_MCKR**	**0x00000102**
EBI Chip Select Assignment	**NAND**	**EBI_CSA**	
		EBI_CS3A	**0x8**

6.2 PIOs

6.2.1 Standard NAND

Two PIO lines are needed for CE and RDY/BSY. NANDOE (PC0) and NANDWE (PC1) are to be configured for NAND Flash usage.

Table 6-2. Peripheral Configuration for Standard NAND on AT91SAM9261

Description	Settings	AT91 libV3 Function
NANDOE and NANDWE are respectively PC0 and PC1	**Output**	`AT91F_PIO_CfgPeriph(AT91C_BASE_PIOC, (AT91C_PC0_SMOE | AT91C_PC1_SMWE), 0);`
PC14 is CE (1)	**Output**	`AT91F_PIO_CfgOutput(AT91C_BASE_PIOC, AT91C_PIO_PC14);`
PC15 is RDY/BSY (1)	**Input**	`AT91F_PIO_CfgInput(AT91C_BASE_PIOC, AT91C_PIO_PC15);`
Enable PIO clock (1)		`AT91F_PIOC_CfgPMC ();`

Note: 1. Any free PIO can be used for this purpose

6.2.2 “CE don’t care” NAND

One PIO line only is needed, for RDY/BSY. NCS3 has the NANDCS function. On the AT91SAM9260 the NCS3 is multiplexed with a PIO line allowing standard NAND Flash and “CE don’t care” device support without any hardware modification.

Table 6-3. Peripheral Configuration for “CE don’t Care” NAND on AT91SAM9260

Description	Settings	AT91 libV3 function
NCS3 is CE	**Output**	`AT91F_PIO_CfgPeriph(AT91C_BASE_PIOC, AT91C_PC14_NCS3_NANDCS,0);`
PC13 is RDY/BSY (1)	**Input**	`AT91F_PIO_CfgInput(AT91C_BASE_PIOC, AT91C_PIO_PC13);`
Enable PIO clock (1)		`AT91F_PIOC_CfgPMC ();`

Note: 1. Any free PIO can be used for this purpose

6.3 SMC Timings

The K9F2G08U0M is a 256 MB device connected with an 8-bit data bus width.

An accurate one-to-one comparison is necessary between NandFlash and SMC waveforms for a complete SMC configuration. Figure 6-1 and Figure 6-2 show two cases that highlight all the required timings.

Figure 6-1. COMMAND LATCH and ADDRESS LATCH Cycle

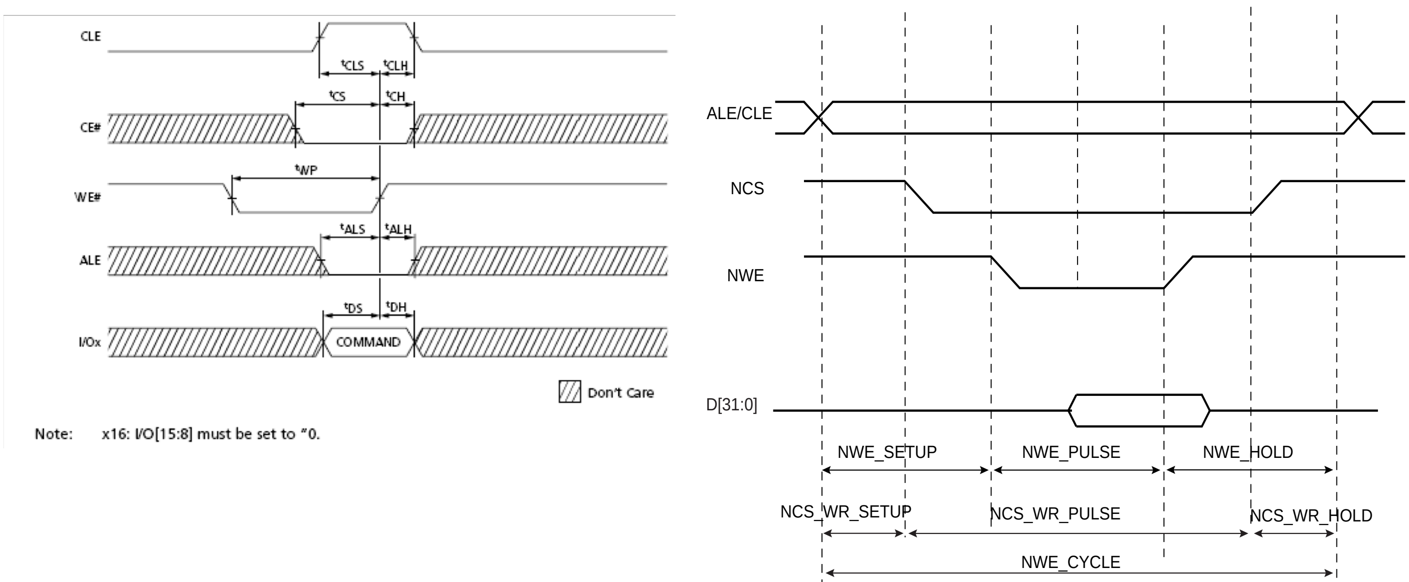
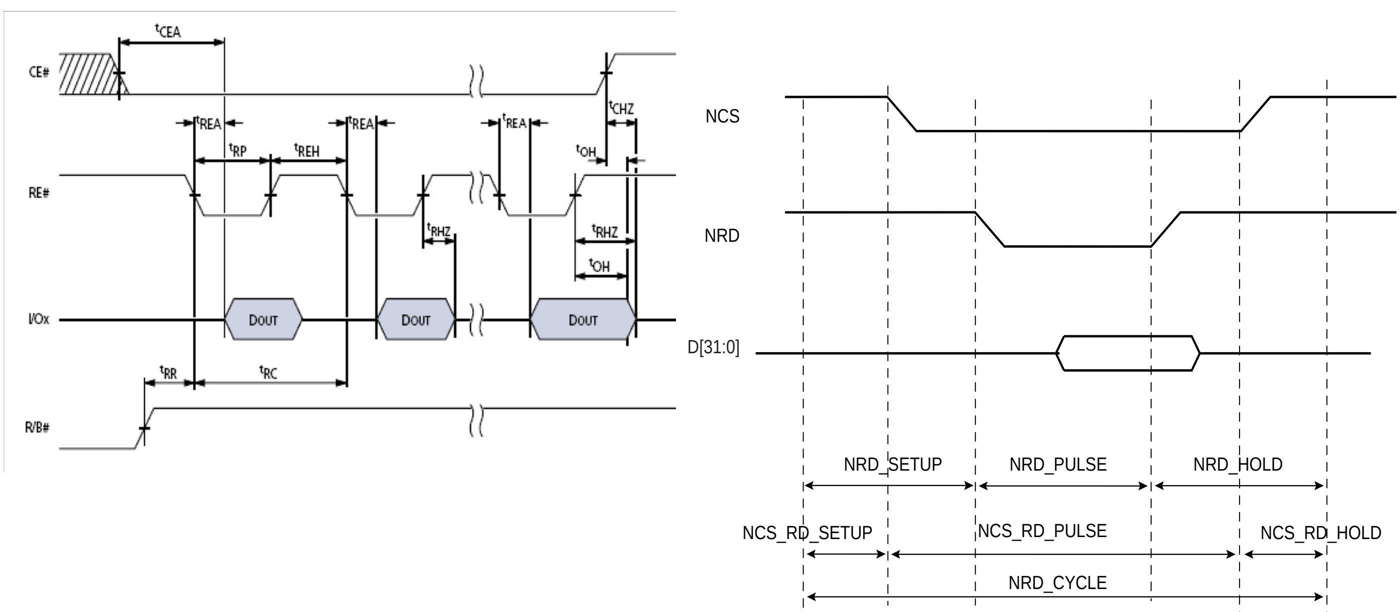


Figure 6-2. SERIAL ACCESS Cycle after READ

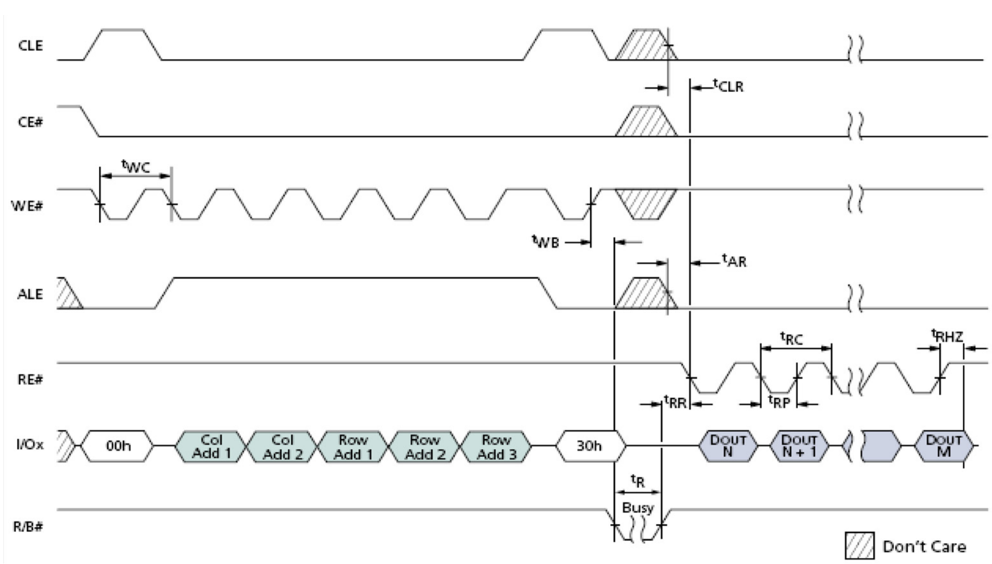


These timings are summarized in Table 6-4.

Table 6-4. NAND Flash Timings vs. SMC Configuration

Timing	Name	Value (ns) Min	Max	SMC Description for "CE don't Care" NAND	SMC Description for Standard NAND	Value @ 100 MHz (cycles)
Setup	tCLS	25		NWE Setup + NWE Pulse	N/A - CE is a PIO line	3
	tALS	25		NWE Setup + NWE Pulse	N/A - CE is a PIO line	3
	tCS	35		NWE Setup + NWE Pulse	N/A - CE is a PIO line	4
	tDS	20		NWE Setup + NWE Pulse	NWE Setup + NWE Pulse	2
	tCEA		45	Not programmable	Not programmable	
	tREA		30	Not programmable	Not programmable	
	tRR	20		managed by software	managed by software	
Hold	tCLH	10		NWE Hold	NWE Hold	1
	tALH	10		NWE Hold	NWE Hold	1
	tCH	10		NWE Hold	N/A - CE is a PIO line	1
	tDH	10		Data Float Time	Data Float Time	1
	tOH	15		Data Float Time	Data Float Time	2
	tREH	15		NRD Cycle - NRD Pulse	NRD Cycle - NRD Pulse	2
Pulse	tWP	25		NWE pulse length	NWE pulse length	3
	tRP	25		NRD pulse length	NRD pulse length	3
	tWC	45		NWE cycle	NWE cycle	5
	tRC	50		NRD cycle	NRD cycle	5

As CLE and ALE are address lines (A21, A22), an additional setup timing is required to respect tAR (10 ns) and tCLR (10 ns) on STATUS or RANDOM DATA READ Cycle.



To fit these requirements the values to program in SMC are:

- NRD_SETUP = NWE_SETUP = 1
- NRD_CYCLE = NWE_CYCLE = 5
- NRD_PULSE = NWE_PULSE = 3
- Data Float Time = 2

Therefore all the timings are realized:

- NWE_HOLD = NRD_CYCLE - NRD_PULSE - NRD_SETUP = 1
- NRD_CYCLE - NRD_PULSE = 2
- NWE_SETUP + NWE_PULSE = 4

Table 6-5 gives SMC register configurations, other fields keep the reset values.

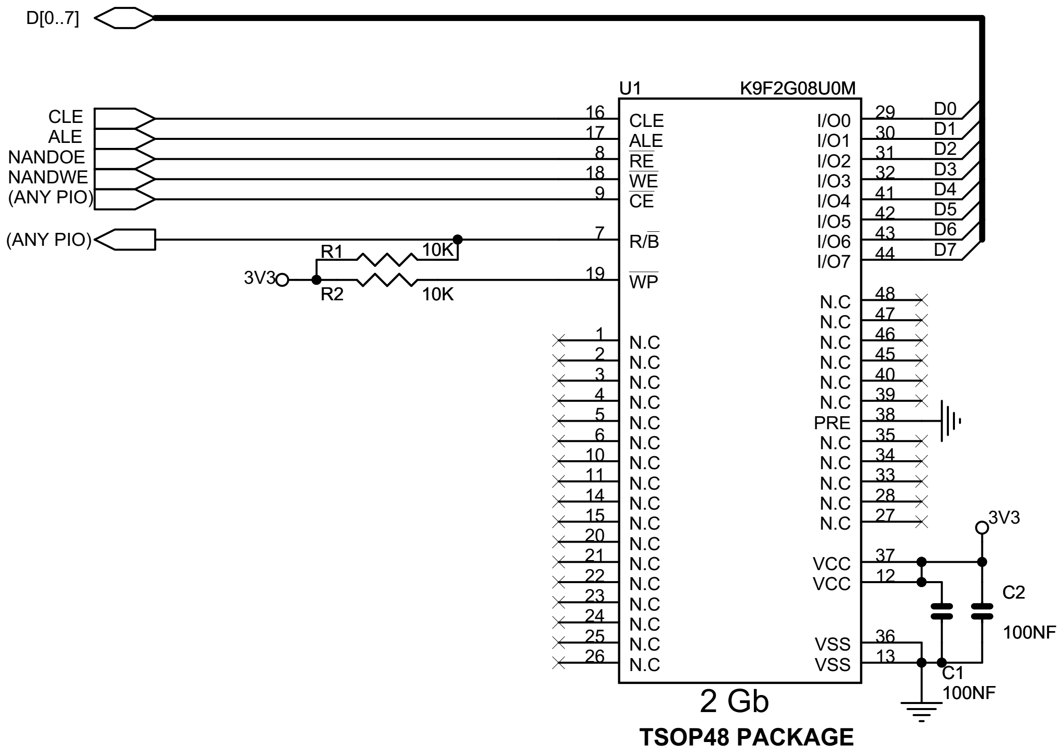
Table 6-5. SMC NCS3 Configuration

Description	**Settings**	**Register/field**	**Value**
NWE and NRD Setup	**1 cycle**	**SMC_SETUP[3]**	**0x01010101**
NRD and NWE Pulse length	**3 cycles**	**SMC_PULSE[3]**	**0x03030303**
NRD and NWE Cycle length	**5 cycles**	**SMC_CYCLE[3]**	**0x05050505**
Chip Select Control		**SMC_CTRL[3]**	**0x00020003**
Read Mode	**NRD**	**READ_MODE**	**0x1**
Write Mode	**NWE**	**WRITE_MODE**	**0x2**
NWAIT action	**disable**	**EXNW_MODE**	**0**
Databus Width	**8 (bits)**	**DBW**	**0**
Data Float Time	**2 (cycles)**	**TDF_CYCLES**	**2**

7. NAND Flash Connection Example on AT91SAM9260

7.1 8-bit NAND Flash

7.1.1 Hardware Configuration



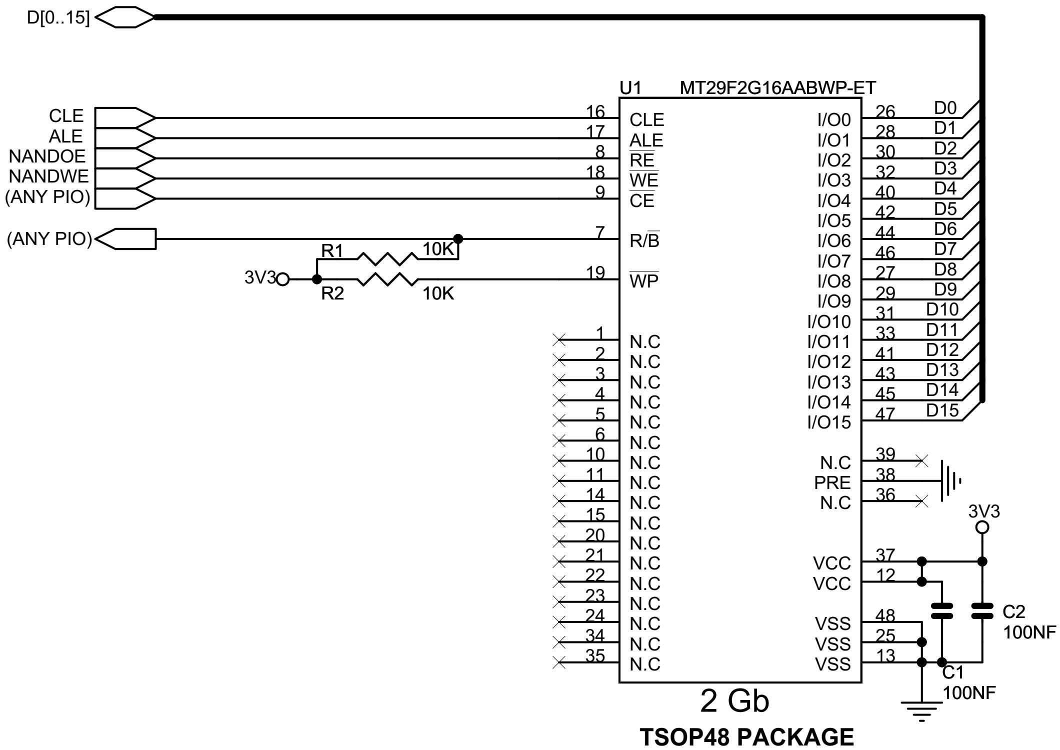
7.1.2 Software Configuration

The following configuration must be carried out:

- Assign the EBI CS3 to the NAND Flash by setting the bit EBI_CS3A in the EBI Chip Select Assignment Register located in the bus matrix memory space
- Reserve A21/A22 for CLE/ALE functions. Address and Command Latches are controlled respectively by setting to 1 the address bit A21 and A22 during accesses.
- Configure a PIO line as an input and enable the clock of this PIO to manage the Ready/Busy signal.
- Configure Static Memory Controller CS3 Setup, Pulse, Cycle and Mode depending on NAND Flash timings, the data bus width and the system bus frequency.

7.2 16-bit NAND Flash

7.2.1 Hardware Configuration



7.2.2 Software Configuration

The software configuration is the same as for 8-bit NAND Flash except the data bus width programmed in the mode register of the Static Memory Controller.

8. NAND Flash Access on AT91SAM9260

8.1 Boot on NAND Flash

Each AT91SAM9 product embeds a Boot Program that integrates different programs permitting download and/or upload into the different memories of the product.

For the NAND Flash, an initialization phase is done and the NAND Flash Boot program is then executed.

Refer to the section "AT91SAM Boot Program" of the product datasheet for more details.

8.1.1 NAND Flash Initialization

The following actions compose the initialization phase of the "CE don't Care" NAND.

- NAND Flash logic is enabled on NCS3 through EBI_CSA.
- NAND Flash timings are programmed with SMC_SETUP3, PULSE3, CYCLE3 and MODE3 SMC registers.
- PC13 PIO line is dedicated to handle NAND R/B#.

- Another PIO line must be declared to handle CE# on standard devices (not "CE don't Care").
- PC14 PIO line is configured for an NCS3_NANDCS usage.
- The corresponding PIO Clock is started.
- A NAND Flash identification is done on NCS3 memory space.
- A NAND structure is created and filled-in with Flash ID relevant data.
- Data bus width is modified depending on the NAND Flash type.

Once completed, the system is ready to access the NAND Flash with best performances.

A software package that performs this configuration is provided with this Application Note.

8.1.2 NAND Flash Boot

First, it looks for a boot.bin file in the root directory or in the FIRMWARE directory of a FAT12/16 formatted NAND Flash. If such a file is found, code is downloaded into the internal SRAM. This is followed by a remap and a jump to the first address of the SRAM.

If the NAND Flash is not formatted, the NAND Flash Boot program looks for a sequence of eight valid ARM exception vectors. If such a sequence is found, code is downloaded into the internal SRAM. This is followed by a remap and a jump to the first address of the SRAM.

8.1.3 NAND Flash Waveforms

Figure 8-1. CE During Two Page Read on Standard NAND

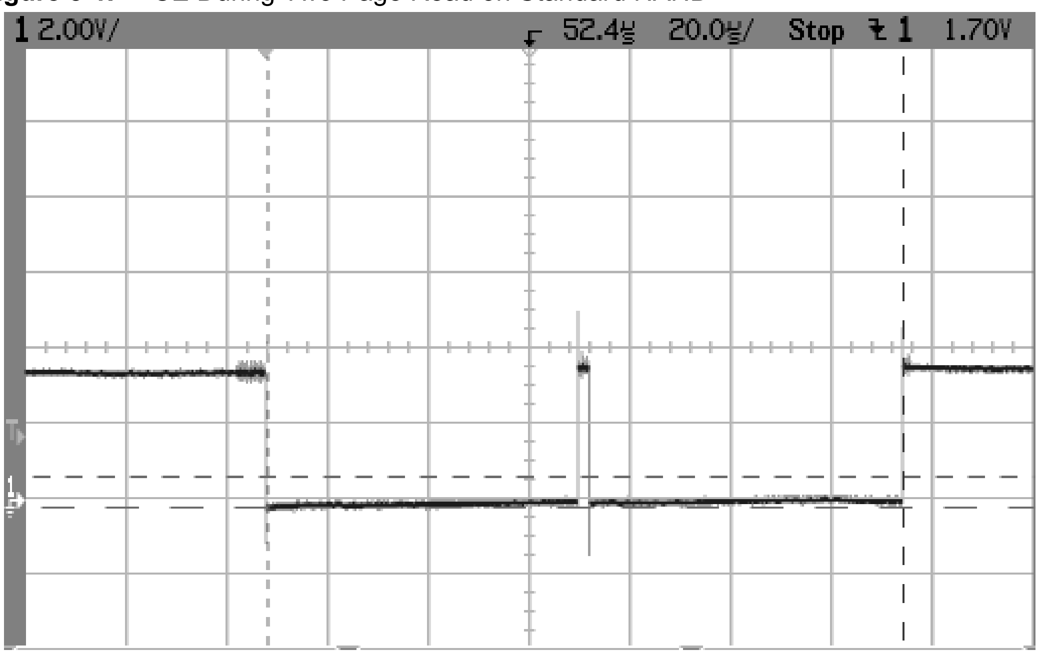
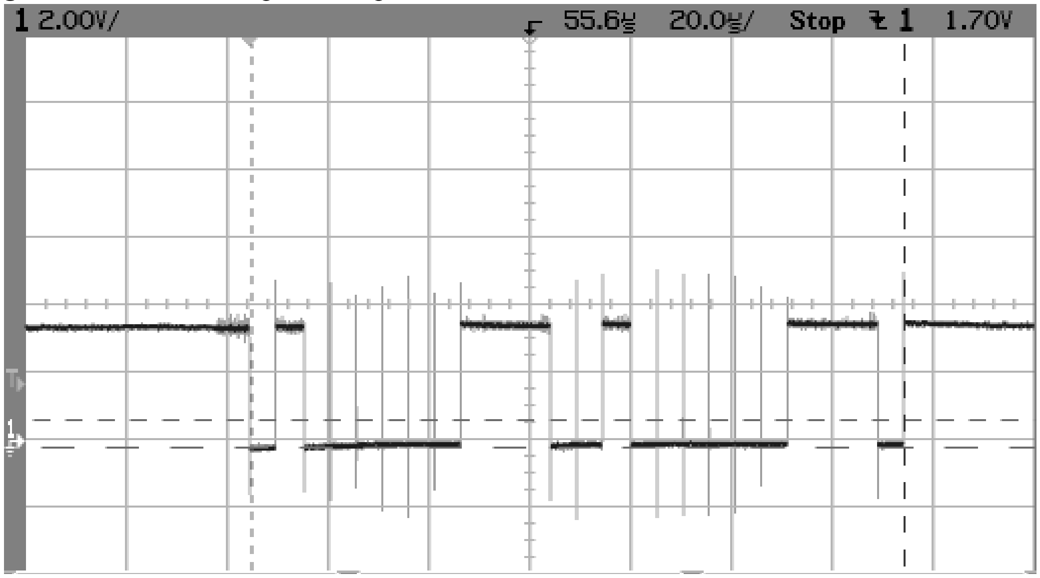


Figure 8-2. CE During Two Page Read on “CE don’t care” NAND



8.1.4 NAND Flash and SAM-BA®

SAM-BA allows the user to read, write, erase and verify NAND Flash devices through RS232 or USB interfaces. This feature is not supported on all the AT91SAM9 Microcontrollers (Refer to SAM Boot Assistant (SAM-BA) User Guide for a more detailed description, lit° 6132.

8.1.4.1 Management of “Bad Blocks”

As a NAND Flash Memory may contain invalid blocks, an Invalid Block Management algorithm is implemented. If an invalid block is detected, it is skipped and data are written in the next block.

The SAM-BA application marks a NAND Flash block as invalid by setting the “BadBlock Info” byte in the 8-byte Invalid Block Information structure to a value different from 0xFF. This is performed for the first two pages of each bad block.

The same principle has to be implemented in a user application in order to read data from a NAND Flash memory written by the SAM-BA application.

8.1.4.2 File Operations

The memory download area provides a simple way to upload and download data.

Files can be sent and received and the target’s memory content can be compared with a file on the PC.

This area also gives access to some specific scripts such as “Enable NAND” that performs the initialization phase necessary before any file operations, and “Erase NAND” that performs a full erase of the device.

Revision History

Doc. Rev	Date	Comments	Change Request Ref.
6255A	09-Oct-06	First issue	
6255B	16-Jun-09	page 1: AT91SAM -> AT91SAM9 Table 6-5 on page 12: NAND Flash support App Note: SMC Timings are not correct	3906 5582

Headquarters

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

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

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 techincal support

Sales Contacts
www.atmel.com/contacts/

Literature Requests
www.atmel.com/literature