AN10600

Connecting NXP ARM-based microcontroller LPC2200 to small page NAND flash

Rev. 01 — 5 March 2007

Application note

Document information

Info	Content
Keywords	NAND flash, LPC2000, Interface, Memory, ECC
Abstract	This application note describes the hardware and firmware design to drive a small page NAND flash memory using an NXP ARM-based microcontroller LPC2000.

Revision history

Rev	Date	Description
01	20070305	Initial version

Contact information

For additional information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

For many applications like POS, communications devices, consumer electronics products or OS based systems, NAND flash is needed for code, mass data storage and lower the cost of material of bill. The LPC2200 family microcontrollers provides an External Memory Controller (EMC) module, which can be easily interfaced between the system bus and external (off-chip) memory. However the on-chip flash controller is a NOR flash controller. To connect to NAND flash, extra control signals are required.

In this application, we will demonstrate how to connect a small page NAND flash to an LPC2200 microcontroller. Section 2 will describe the Hardware Design; Section 3 will describe the firmware driver and the assembler ECC code.

2. Hardware design

The interface hardware consumes LPC2200 with: 2 GPIOs, 1 EMI bank.

The small page NAND flash devices in this application note have an x8 bus width. The memory array is organized in blocks where each block contains 32 pages. The array is split into two areas - the main area and the spare area. The main area of the array is used to store data whereas the spare area is typically used to store Error Correction Codes (ECC), software flags or bad block identification. So each page is split into a main area with two half pages of 256 B each and a spare area of 16 B.

Table 1 describes the interface signal of the design. For detailed schematic, please refer to Appendix A.

Table 1. Interface signal description

LPC2200 signal	**NAND signal**[1]	**Description**
MCU_CS0	-	LPC2200 Low-active Chip Select 0 Output signal, together with the OR gate IC 74HC32 controlling the Read/Write operation of NAND flash. In this way, users can add multiple bus devices on Bank0 of LPC2200.
MCU_WE	NAND_W#	Low-active NAND flash Write Enable Signal.
MCU_OE	NAND_R#	Low-active NAND flash Read Enable Signal.
MCU_P018	R/B#	Ready/Busy (open-drain output). We use LPC2200 GPIO P018 to detect whether NAND flash is busy.
MCU_P019	E#	NAND flash Chip Enable. We use LPC2200 GPIO P019 to enable NAND flash operation.
	NAND_WP#	Write Protect. Enable manual by Jumper J101
MCU_A3	CL	NAND flash Command Latch Enable.
MCU_A2	AL	NAND flash Address Latch Enable
MCU_D0-D7	I/O0-I/O7	Data Input/Outputs, Address Inputs, or Command Inputs for x8 NAND flash

[1] Symbol “#” stands for LOW-active

3. Firmware design

In this application note, we separate the firmware for NAND flash Low-Level Drivers (LLD) into hardware related layer and hardware independent layer. The two layer structure may help customers in easily porting the drivers to a different NXP ARM-based microcontroller with External Memory Interface (EMI) e.g., LPC2378.

3.1 Low level drivers

The below tables summarize the firmware functions. Table 2 shows the hardware related layer functions. Table 3 gives the hardware independent layer functions. For more details, please refer to the corresponding source code found at http://www.nxp.com/standardics/support/documents/?type=software (under heading “LPC22xx NAND source code (AN10600)”).

Table 2. Hardware related functions

Return value	Function name	Argument	Argument description	Function description
Void	nandOpen	-	-	This function triggers to open the Chip Enable signal of NAND flash. This function must be called at the start of any operation.
Void	nandClose	-	-	This function triggers to close the Chip Enable signal of NAND flash. This function must be called at the end of any operation
unsigned int	nandRd_ReadyBusy	-	-	Return the status of NAND flash operation.
Void	nandWr_Cmd	nand_cmd	command to issue to NAND	implements a Command Latch Cycle
Void	nandWr_Addr	nand_addr	address to issue to NAND	implements an Address Latch Cycle
Void	nandWr_Data	nand_data	data to write to NAND	implements a Data Input Latch Cycle
unsigned char	nandRd_Data	-	-	Return the data read from the NAND flash
Void	nandFlashInit	-	-	Initialization of the LPC2200

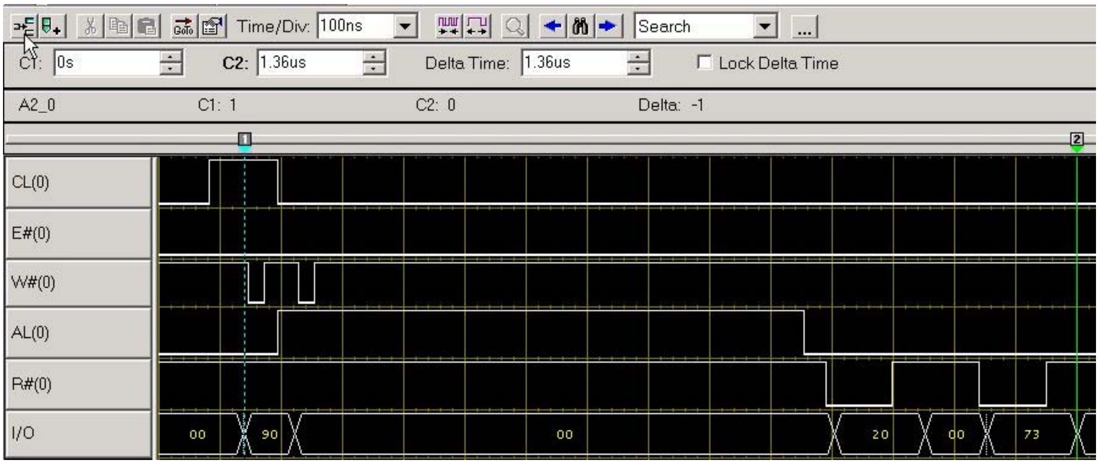
Table 3. Hardware independent functions

Return value	Function name	Argument	Argument description	Function description
unsigned int	nandFlashReadID	-	-	Return the ID number of the NAND flash
unsigned char	nandFlashBlockErase	NandAddr	Block address to be erased	Erase the block and return the erase status
unsigned char	nandFlashPageWrite	NandAddr	NAND page addr.	Write one page of data from RAM to NAND flash and return the write status
		*Buffer	Pointer to data in RAM	

Return value	Function name	Argument	Argument description	Function description
Void	nandFlashPageRead	NandAddr	NAND page addr.	Read one page of data from NAND flash to RAM
		*Buffer	Pointer to space in RAM	
unsigned char	nandFlashSpareWrite	NandAddr	NAND spare addr.	Write spare data from RAM to NAND flash and return the write status
		*Buffer	Pointer to spare data in RAM	
Void	nandFlashSpareRead	NandAddr	NAND spare addr.	Read spare data from NAND flash to RAM
		*Buffer	Pointer to space in RAM	
unsigned char	nandFlashStatusRegRead	-	-	Return the status register value of NAND flash
unsigned char	nandFlashReset	-	-	Reset NAND flash and return NAND flash status
unsigned char	nandFlashCopyBack	SrcAddr	NAND source page addr.	Implement a copy back operation and return the NAND flash status
		DestAddr	NAND destination page addr.	

Note: The small page NAND flash devices in this application notes have 128 MB memory size. If the size of the small page NAND flash is equal or larger than 512 MB, then four address cycles instead of three address cycles are required.

Fig 1 and Fig 2 show the output waveforms of the driver by Tektronix TLA5202 Logic Analyzer.

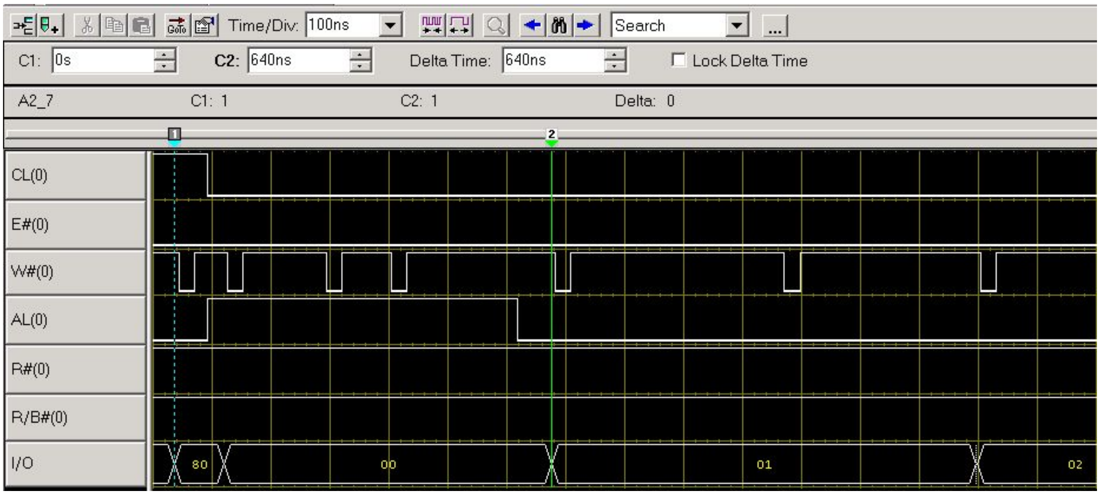


Time/Div: 100ns

C1: 0s C2: 1.36us Delta Time: 1.36us □ Lock Delta Time

A2_0 C1: 1 C2: 0 Delta: -1

Fig 1. Read ID Operation waveform (0x2073 is NAND128W3A2 ID number)



Time/Div: 100ns

C1: 0s

C2: 640ns

Delta Time: 640ns

□ Lock Delta Time

A2_7 C1: 1 C2: 1 Delta: 0

Fig 2. Page Write Operation Waveform (Write Page data to Page Address 0x00)

3.2 ECC code

The Error Check Correction (ECC) software we present in this application note is written by optimized ARM assembler language to save CPU cycles. A so-called Single Issue Multiple Data (SIMD) process is implemented. 16 B data are loaded into register R2-R5 at one time. Processing time can be largely saved by processing those 16 B data together. 512 B data needs only 32 cycles. A simulation in ARMulate shows that 512 B data only consumes 6000 core cycles to generate 3 B ECC code. The ECC functions can be used as a guide in implementing ECC code generation and correction for any length of data.

Note: Because the ECC code is optimized with SIMD process, the input data should be word aligned before feeding to the ECC code generator.

3.3 How to use the driver

To use the NAND flash driver on different platform, we need to slightly modify the driver as the following two steps:

1. Configure the EMI of LPC2200 we want to use for controlling the NAND flash.
    a. Modify below code in function *void nandFlashInit(void)* in file *lpc2200Nand.c*.

```
BCFG0 = (( 0<<28 )              //8bit width of data bus
         | ( 0x0<<11 )          //1 CCLK cycle write
         | ( 1<<10 )
         | ( 0x1<<5 )           //4 CCLK cycle read
         | ( 0x0 ));            //1 CCLK cycle idle
```

    b. Modify below code in file *lpc2200nand.h*.

```
#define   NAND_DATA          0x8000 0000
#define   NAND_ALE           0x8000 0004
#define   NAND_CLE           0x8000 0008
```

2. Configure the GPIO of LPC2200 for NAND flash interface.
   a. Modify below code in function *void nandFlashInit(void)* in file *lpc2200Nand.c*.

```
PINSEL2 = PINSEL2
              & ( ~(0x3<<26) )
              | ( 1<<25 )          //enable Addr2 as NAND flash ALE
                                   //enable Addr3 as NAND flash CLE
              | ( 1<<8 )           //enable WE
              & ( ~( 0x3<<4 ) )
              | ( 1<<4 );          //enable CS0, OE, D0...D7, D8...D15
```

   and

```
PINSEL1 &= ( ~0xf0 );
IO0DIR = IO0DIR
              | ( 1<<19 )          //P019 configured as output pin
              & ( ~( 1<<18 ) );    //P018 configured as input pin
```

   b. Modify below code in file *lpc2200nand.h*.

```
#define nandOpen()          ( IO0CLR |= ( 1<<19 ) )
#define nandClose()         ( IO0SET |= ( 1<<19 ) )
#define nandRd_ReadyBusy()  ( IO0PIN & ( 1<<18 ) )
```

4. Conclusion

In this application, we introduce a reference firmware and hardware design of connecting LPC2000 with NAND flash. The NAND flash driver can be easily migrated to different NXP ARM-based microcontroller with external memory interface (EMI) like LPC2378.

Appendix A

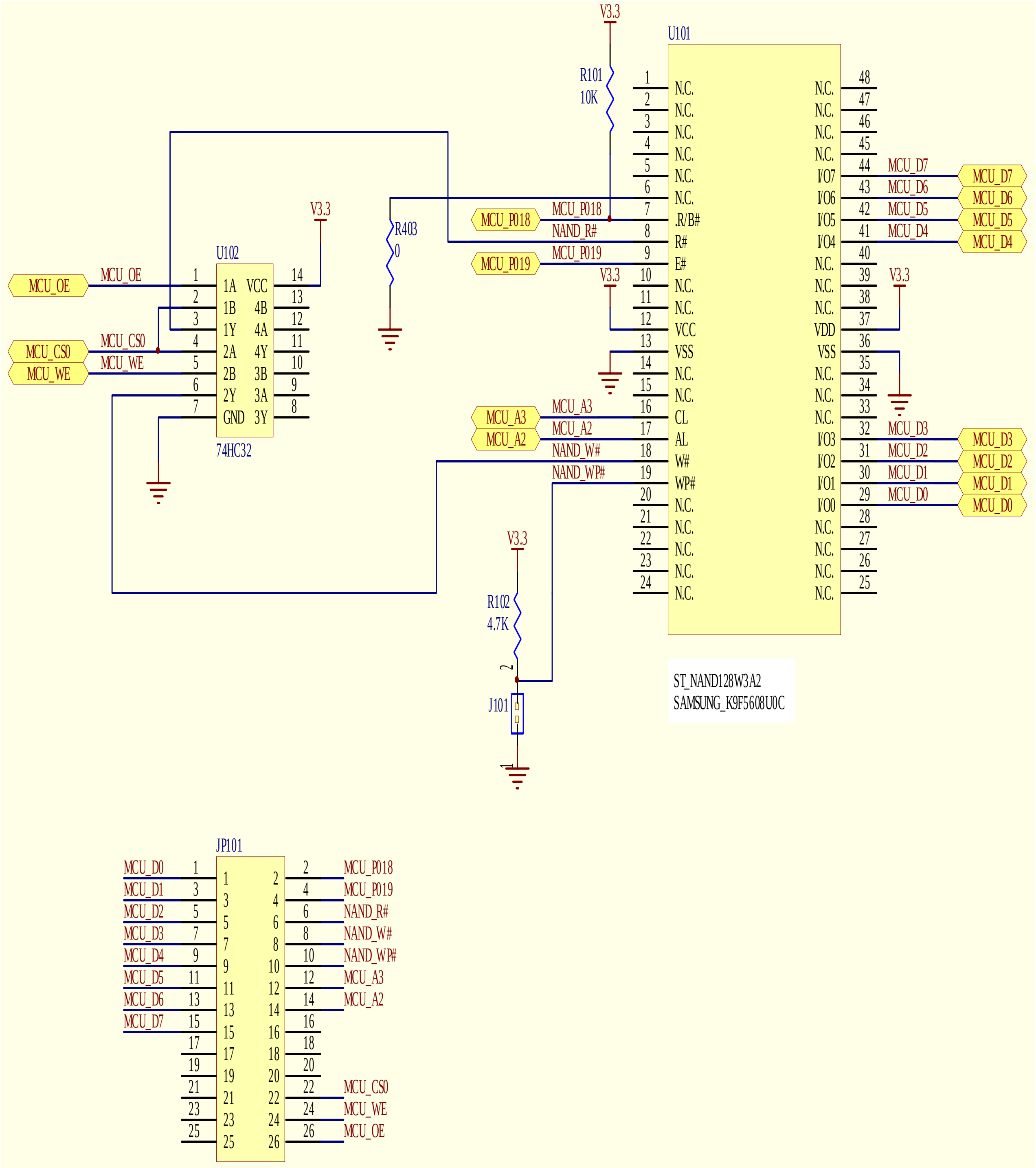


Fig 3. Schematic of LPC2200 interface to small page NAND flash

Table 4. LPC2200 interface to small page NAND flash reference design bill of material (BOM)

Item	Quantity	Designator	Part Type	Manufacturer
1	1	U001	LPC2214	NXP Semiconductor
2	1	U101	NAND128W3A2	ST Semiconductor
3	1	U102	74HC32	NXP Semiconductor

5. Legal information

5.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

5.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

6. Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, email to: salesaddresses@nxp.com

Date of release: 5 March 2007

Document identifier: AN10600_1

founded by PHILIPS