

# ES\_LPC2468

Errata sheet LPC2468

Rev. 11 — 1 June 2011

Errata sheet

## Document information

Info	Content
<b>Keywords</b>	LPC2468 errata
<b>Abstract</b>	<p>This errata sheet describes both the known functional problems and any deviations from the electrical specifications known at the release date of this document.</p> <p>Each deviation is assigned a number and its history is tracked in a table.</p>



**Revision history**

Rev	Date	Description
11	20110601	<ul style="list-style-type: none"><li>Added USB.5.</li></ul>
10	20110420	<ul style="list-style-type: none"><li>Added Note.2.</li></ul>
9	20110301	<ul style="list-style-type: none"><li>Added ADC.2.</li></ul>
8	20100513	<ul style="list-style-type: none"><li>Added VBAT.2</li></ul>
7	20100401	<ul style="list-style-type: none"><li>The format of this errata sheet has been redesigned to comply with the new identity guidelines of NXP Semiconductors.</li><li>Added Ethernet.4</li></ul>
6	20090508	<ul style="list-style-type: none"><li>Added Rev D</li></ul>
5	20080602	<ul style="list-style-type: none"><li>Added Note.1</li></ul>
4	20080408	<ul style="list-style-type: none"><li>Added VBAT.1</li></ul>
3	20080212	<ul style="list-style-type: none"><li>USB.4 was added</li></ul>
2	20070921	<ul style="list-style-type: none"><li>Added Rev. B</li><li>Removed Rev A from ESD.1. ESD.1 does not appear in Rev A. It was accidentally listed in version 1.2</li></ul>
1	20070720	<ul style="list-style-type: none"><li>Added Ethernet.3</li><li>Updated Ethernet.1</li><li>Updated Flash.1</li></ul>

**Contact information**

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

## 1. Product identification

The LPC2468 devices typically have the following top-side marking:

```
LPC2468xxx
xxxxxxx
xxYYWWR[x]
```

The last/second to last letter in the third line (field 'R') will identify the device revision. This Errata Sheet covers the following revisions of the LPC2468:

**Table 1. Device revision table**

Revision identifier (R)	Revision description
'L'	Initial device revision
'A'	Second device revision
'B'	Third device revision
'D'	Fourth device revision

Field 'YY' states the year the device was manufactured. Field 'WW' states the week the device was manufactured during that year.

## 2. Errata overview

**Table 2. Functional problems table**

Functional problems	Short description	Revision identifier	Detailed description
Ethernet.4	Ethernet TxConsumeIndex register does not update correctly after the first frame is sent	'L', 'A', 'B', 'D'	<a href="#">Section 3.1 on page 5</a>
ADC.1	ADDRx read conflicts with hardware setting of DONE bit	'L'	<a href="#">Section 3.2 on page 5</a>
Ethernet.1	Setting up the Ethernet interface in RMII mode	'L'	<a href="#">Section 3.3 on page 6</a>
Ethernet.2	Ethernet SRAM disabled	'L'	<a href="#">Section 3.4 on page 7</a>
Ethernet.3	RxDescriptor number cannot be greater than 4	'L'	<a href="#">Section 3.5 on page 7</a>
I2S.1	I2S DMA can stall	'L'	<a href="#">Section 3.6 on page 7</a>
PLL.1	PLL output is limited to 290 MHz	'L'	<a href="#">Section 3.7 on page 8</a>
SRAM.1	16 kB SRAM can not be used for code execution	'L'	<a href="#">Section 3.8 on page 8</a>
USB.1	USB_NEED_CLK is always asserted	'L'	<a href="#">Section 3.9 on page 9</a>
USB.2	U1CONNECT is not functional	'L'	<a href="#">Section 3.10 on page 9</a>
USB.3	V <sub>BUS</sub> status input is not functional	'L'	<a href="#">Section 3.11 on page 9</a>
USB.4	USB_PWRDx pin(s) does not function as intended	'L'	<a href="#">Section 3.12 on page 10</a>
WDT.1	Accessing non-Watchdog APB registers in the middle of the feed sequence causes a reset	'L'	<a href="#">Section 3.13 on page 10</a>
Core.1	Incorrect update of the Abort Link register in Thumb state	'L', 'A', 'B', 'D'	<a href="#">Section 3.14 on page 10</a>
Flash.1	Operating speed out of on-chip flash is restricted	'L', 'A'	<a href="#">Section 3.15 on page 11</a>
MAM.1	Code execution failure can occur with MAM Mode 2	'L', 'A'	<a href="#">Section 3.16 on page 12</a>
CAN.1	Data overrun condition can lock the CAN controller	'L', 'A', 'B'	<a href="#">Section 3.17 on page 12</a>

**Table 2. Functional problems table ...continued**

Functional problems	Short description	Revision identifier	Detailed description
VBAT.1	Increased power consumption on VBAT when VBAT is powered before the 3.3 V supply used by rest of device	'-', 'A', 'B'	<a href="#">Section 3.18 on page 13</a>
VBAT.2	The VBAT pin cannot be left floating	'-', 'A', 'B'	<a href="#">Section 3.19 on page 13</a>
ADC.2	External sync inputs not operational	'-', 'A', 'B', 'D'	<a href="#">Section 3.20 on page 14</a>
USB.5	USB host controller hangs on a dribble bit	'-', 'A', 'B', 'D'	<a href="#">Section 3.21 on page 15</a>

**Table 3. AC/DC deviations table**

AC/DC deviations	Short description	Product version(s)	Detailed description
ESD.1	2 kV ESD requirements are not met on the RTCX1 pin	'-'	<a href="#">Section 4.1 on page 16</a>

**Table 4. Errata notes table**

Errata notes	Short description	Revision identifier	Detailed description
Note.1	When the input voltage is $V_i \geq V_{DD} I/O + 0.5$ V on each of the following port pins P0.23, P0.24, P0.25, P0.26, P1.30, P1.31, P0.12, and P0.13 (configured as general purpose input pin (s)), current must be limited to less than 4 mA by using a series limiting resistor.	'-', 'A', 'B', 'D'	<a href="#">Section 5.1 on page 16</a>
Note.2	On the LPC2468 Rev D, design changes to the Memory Accelerator Module were made to enhance timing and general performance.	'D'	<a href="#">Section 5.2 on page 16</a>

### 3. Functional problems detail

---

#### 3.1 Ethernet.4: Ethernet TxConsumeIndex register does not update correctly after the first frame is sent

**Introduction:**

The transmit consume index register defines the descriptor that is going to be transmitted next by the hardware transmit process. After a frame has been transmitted hardware increments the index, wrapping the value to 0 once the value of TxDescriptorNumber has been reached. If the TxConsumeIndex equals TxProduceIndex the descriptor array is empty and the transmit channel will stop transmitting until software produces new descriptors.

**Problem:**

The TxConsumeIndex register is not updated correctly (from 0 to 1) after the first frame is sent. After the next frame sent, the TxConsumeIndex register is updated by two (from 0 to 2). This only happens the very first time, so subsequent updates are correct (even those from 0 to 1, after wrapping the value to 0 once the value of TxDescriptorNumber has been reached)

**Work-around:**

Software can correct this situation in many ways; for example, sending a dummy frame after initialization.

#### 3.2 ADC.1: ADDRx read conflicts with hardware setting of the DONE bit

**Introduction:**

The LPC2468 has a 10-bit ADC, which can be used to measure analog signals and convert the signals into a 10-bit digital result. There are eight A/D channels and each channel has its own individual A/D Data Register (ADDR0 to ADDR7). The A/D Data Register holds the result when an A/D conversion is complete, and also includes the flags that indicate when a conversion has been completed (DONE bit) and when a conversion overrun has occurred. The DONE bit is cleared when the respective A/D Data Register is read.

**Problem:**

If a software read of ADDRx conflicts with the hardware setting of the DONE bit in the same register (once a conversion is completed) then the DONE bit gets cleared automatically, thereby clearing the indication that a conversion was completed.

**Work-around:**

For software controlled mode or burst mode with only one channel selected, the DONE bit in the A/D Global Data Register (located at 0xE003 4004) can be used instead of the individual ADDRx result register with no impact on performance.

For burst mode with multiple channels selected, the DONE bit together with the CHN field in the A/D Global Data Register can be used with some impact on throughput.

### 3.3 Ethernet.1: Setting up the Ethernet interface in RMII mode

#### Introduction:

The LPC2468 has an Ethernet interface, which can be interfaced with an off-chip PHY using the RMII interface.

#### Problem:

The default configuration of the device does not enable the RMII interface.

#### Work-around:

To use the Ethernet interface in RMII mode write a 1 to bit 12 (P1.16) in PINSEL2 register (located at 0xE002 C008). This workaround only applies for Rev '-' devices and does not apply for Rev 'A' and newer devices. In order to have both Rev '-' and other revisions coexist in the same piece of software, the MAC module ID can be used to identify the part and determine if port pin P1.6 needs to be set or not.

Here are the steps (along with some sample code) to initialize the MAC based on the module ID:

1. In master header file ILPC24xx.h, make sure Module ID is defined (Please note, this ID register is not documented in the User's Manual).

```
#define MAC_BASE_ADDR          0xFFE00000
#define MAC_MODULEID (*(volatile unsigned long*)(MAC_BASE_ADDR + 0xFFC)) /*
Module ID reg (R0) */
```

2. In the beginning of the MAC initialization file, add below definition:

```
#define OLD_EMAC_MODULE_ID      0x3902 << 16) | 0x2000)
```

3. In MAC initialization routine, right after setting the EMAC clock in the PCONP register, add a few lines as below:

```
/* Turn on the ethernet MAC clock in PCONP, bit 30 */
regVal = PCONP;
regVal |= PCONP_EMAC_CLOCK;
PCONP = regVal;
/*-----
* Write to PINSEL2/3 to select the PHY functions on P1[17:0]
* P1.6, ENET-TX_CLK, has to be set for Rev '-' devices and it
* must not be set for Rev 'A' and newer devices
*-----*/
regVal = MAC_MODULEID;
if ( regVal == OLD_EMAC_MODULE_ID )
{
    /* On Rev. '-', MAC_MODULEID should be equal to
    OLD_EMAC_MODULE_ID, P1.6 should be set. */
    PINSEL2 = 0x50151105;
    /* selects P1[0,1,4,6,8,9,10,14,15] */
}
else
{
    /* on rev. 'A', MAC_MODULEID should not equal to
```

```
        OLD_EMAC_MODULE_ID, P1.6 should not be set. */
        PINSEL2 = 0x50150105;
        /* selects P1[0,1,4,8,9,10,14,15] */
    }
    PINSEL3 = 0x00000005;    /* selects P1[17:16] */
```

### 3.4 Ethernet.2: Ethernet SRAM disabled

#### Introduction:

The LPC2468 has an Ethernet interface, which has a dedicated 16 kB SRAM.

#### Problem:

When the Ethernet block is disabled (in the PCONP register located at 0xE01F C0C4), the Ethernet SRAM is also disabled.

#### Work-around:

Enable the Ethernet block by setting the PCENET bit (bit no. 30) in the PCONP register. The Ethernet SRAM is now enabled.

### 3.5 Ethernet.3: Receive Status registers will not function correctly if RxDescriptor number is greater than 4

#### Introduction:

The Receive number of Descriptors register (RxDescriptor-0xFFE0 0110) defines the number of descriptors in the Descriptor array. Each receive descriptor element in the Descriptor array has an associated status field which consists of the HashCRC word and Status Information word.

#### Problem:

The status words are updated incorrectly if the number of Descriptors set in the Receive number of Descriptors register is greater than or equal to 5.

#### Work-around:

Define 4 or less in the Receive number of Descriptors register.

### 3.6 I2S.1: I<sup>2</sup>S DMA interface is non-operational

#### Introduction:

The LPC2468 has an I<sup>2</sup>S interface, which can be used for audio devices. The I<sup>2</sup>S interface was initially designed to operate with the general purpose DMA controller.

#### Problem:

The DMA controller cannot access the I<sup>2</sup>S interface.

#### Work-around:

No known workaround.

### 3.7 PLL.1: PLL output ( $F_{CCO}$ ) is limited to 290 MHz

#### Introduction:

The PLL input, in the range of 32 KHz to 50 MHz, may initially be divided down by a value “N”, which may be in the range of 1 to 256. Following the PLL input divider is the PLL multiplier. This can multiply the input divider output through the use of a Current Controlled Oscillator (CCO) by a value “M”, in the range of 1 through 32768. The resulting frequency,  $F_{CCO}$  must be in the range of 275 MHz to 550 MHz. This frequency can be divided down (using the Clock Divider registers) to get the desired clock frequencies for the core and peripherals.

#### Problem:

The maximum output of the CCO within the PLL block is limited to 290 MHz.

#### Work-around:

Care should be taken while programming the PLL so that  $F_{CCO}$  resides in the desired range. The suggested setting is to use a 12 MHz external crystal. Use a PLLdivider (N) of 1 and PLL multiplier (M) of 12. Putting the values in the equation:

$$F_{CCO} = (2 \times M \times F_{IN}) / N$$

$$F_{CCO} = 288 \text{ MHz}$$

The CPU Clock Configuration register (located at 0xE01F C104) can then be used to divide this frequency by 4 to produce the maximum CPU speed of 72 MHz (except on Rev ‘-’ and Rev ‘A’, see Flash.1).

### 3.8 SRAM.1: 16 kB SRAM cannot be used for code execution

#### Introduction:

The LPC2468 has 16 kB of SRAM on the AHB2 bus, which would generally be used by the Ethernet block.

#### Problem:

The 16 kB of SRAM can only be used as data RAM. Code can not be executed from this memory.

#### Work-around:

No known workaround.

### 3.9 USB.1: USB\_NEED\_CLK is always asserted

**Introduction:**

The USB\_NEED\_CLK signal is used to facilitate going into and waking up from chip Power Down mode. USB\_NEED\_CLK is asserted if any of the bits of the USBClkSt register are asserted.

**Problem:**

The USB\_NEED\_CLK bit of the USBIntSt register (located at 0xE01F C1C0) is always asserted, preventing the chip from entering Power Down mode when the USBWAKE bit is set in the INTWAKE register (located at 0xE01F C144).

**Work-around:**

After setting the PCUSB bit in PCONP (located at 0xE01F C0C4), write 0x1 to address 0xFFE0C008. The USB\_NEED\_CLK signal will now function correctly. Writing to address 0xFFE0C008 only needs to be done once after each chip reset.

### 3.10 USB.2: U1CONNECT signal is not functional

**Introduction:**

U1CONNECT Signal (alternate function of P2.9) is part of the SoftConnect USB feature, which is used to switch an external 1.5 KW resistor under the software control.

**Problem:**

The USB U1CONNECT alternate function does not work as expected.

**Work-around:**

Configure P2.9 as a GPIO pin, and use it to enable the pull-up resistor on the U1D+ pin.

### 3.11 USB.3: V<sub>BUS</sub> status input is not functional

**Introduction:**

The V<sub>BUS</sub> signal indicates the presence of USB bus power.

**Problem:**

The V<sub>BUS</sub> status input is not functional.

**Work-around:**

Configure P1.30 as a GPIO pin, and poll it to determine when V<sub>BUS</sub> goes to 0, signalling a disconnect event.

### 3.12 USB.4: USB\_PWRDx pin(s) does not function as intended

**Introduction:**

The device has a USB\_PWRD1 signal for USB port 1 and a USB\_PWRD2 signal for USB port 2. Both signals monitor the status of  $V_{BUS}$  (USB bus power) and are active high signals.

**Problem:**

On the Rev '-', the USB\_PWRDx signals are implemented as active low signals and as a result, they are unable to monitor the status of  $V_{BUS}$  without external inverter.

**Work-around:**

An external inverter is needed on the USB\_PWRDx pin(s) to be able to monitor the  $V_{BUS}$  status.

### 3.13 WDT.1: Accessing non-Watchdog APB registers in the middle of the feed sequence causes a reset

**Introduction:**

The Watchdog timer can reset the microcontroller within a reasonable amount of time if it enters an erroneous state.

**Problem:**

After writing 0xAA to WDFEED, any APB register access other than writing 0x55 to WDFEED may cause an immediate reset.

**Work-around:**

Avoid APB accesses in the middle of the feed sequence. This implies that interrupts and the GPDMA should be disabled while feeding the Watchdog.

### 3.14 Core.1: Incorrect update of the Abort Link register in Thumb state

**Introduction:**

If the processor is in Thumb state and executing the code sequence STR, STMIA or PUSH followed by a PC relative load, and the STR, STMIA or PUSH is aborted, the PC is saved to the abort link register.

**Problem:**

In this situation the PC is saved to the abort link register in word resolution, instead of half-word resolution.

**Conditions:**

The processor must be in Thumb state, and the following sequence must occur:

```
<any instruction>  
<STR, STMIA, PUSH> <---- data abort on this instruction  
LDR rn, [pc,#offset]
```

In this case the PC is saved to the link register R14\_abt in only word resolution, not half-word resolution. The effect is that the link register holds an address that could be #2 less than it should be, so any abort handler could return to one instruction earlier than intended.

**Work-around:**

In a system that does not use Thumb state, there will be no problem.

In a system that uses Thumb state but does not use data aborts, or does not try to use data aborts in a recoverable manner, there will be no problem.

Otherwise the workaround is to ensure that a STR, STMIA or PUSH cannot precede a PC-relative load. One method for this is to add a NOP before any PC-relative load instruction. However this would have to be done manually.

### 3.15 Flash.1: Operating speed out of on-chip flash is restricted

**Introduction:**

The operating speed of this device out of internal flash/SRAM is specified at 72 MHz.

**Problem:**

Code execution from internal flash is restricted depending upon the device revision:

1. Rev 'A' devices: Code execution from internal flash is restricted to a maximum of 60 MHz.

For example, use a PLL output frequency of  $F_{CCO} = 360$  MHz and divide it by 6 (CCLKSEL = 5) to generate 60 MHz CPU clock (Do not use even values for CCLKSEL).

2. Rev 'C' devices: Code execution from internal flash is restricted to a maximum of 60 MHz also.

However, this device revision has one more restriction in terms of the PLL output frequency ( $F_{CCO}$  - Please refer to PLL.1 above).  $F_{CCO}$  is limited to 290 MHz.

Considering the same example in PLL.1 (Input crystal - 12 MHz, N = 1, M = 12):  
 $F_{CCO} = 288$  MHz

The CPU Clock Configuration register (located at 0xE01F C104) can then be used to divide this frequency by 6 (CCLKSEL = 5) to achieve a maximum of 48 MHz. Since this register only accepts odd values for CCLKSEL, a division by 5 (CCLKSEL = 4) is not a valid option.

In both the above revisions, code can still execute out of SRAM at up to 72 MHz.

**Work-around:**

None.

### 3.16 MAM.1: Under certain conditions in MAM Mode 2 code execution out of internal flash can fail

#### Introduction:

The MAM block maximizes the performance of the ARM processor when it is running code in flash memory. It includes three 128-bit buffers called the Prefetch Buffer, the Branch Trail Buffer and the data buffer. It can operate in 3 modes; Mode 0 (MAM off), Mode 1 (MAM partially enabled) and Mode 2 (MAM fully enabled).

#### Problem:

Under certain conditions when the MAM is fully enabled (Mode 2) code execution from internal flash can fail. The conditions under which the problem can occur is dependent on the code itself along with its positioning within the flash memory.

#### Work-around:

If the above problem is encountered then Mode 2 should not be used. Instead, partially enable the MAM using Mode 1.

### 3.17 CAN.1: Data Overrun condition can lock the CAN controller

#### Introduction:

Each CAN controller provides a double Receive Buffer (RBX) per CAN channel to store incoming messages until they are processed by the CPU. Software task should read and save received data as soon as a message reception is signaled.

In cases where both receive buffers are filled and the contents are not read before the third message comes in, a CAN Data Overrun situation is signaled. This condition is signaled via the Status register and the Data Overrun Interrupt (if enabled).

#### Problem:

In a Data Overrun condition, the CAN controller is locked from further message reception.

#### Work-around:

1. Recovering from this situation is only possible with a soft reset to the CAN controller.
2. If software cannot read all messages in time before a third message comes in, it is recommend to change the acceptance filtering by adding further acceptance filter group(s) for messages, which are normally rejected. With this approach, the third incoming message is accepted and the Data Overrun condition is avoided. These additional messages are received with the corresponding group index number can be easily identified and rejected by software.

### 3.18 VBAT.1: Increased power consumption on VBAT when VBAT is powered before the 3.3 V supply used by rest of the device

#### Introduction:

The device has a VBAT pin which provides power only to the RTC and Battery RAM. VBAT can be connected to a battery or the same 3.3 V supply used by rest of the device ( $V_{DD(3V3)}$  pin,  $V_{DD(DCDC)(3V3)}$  pin).

#### Problem:

If VBAT is powered before the 3.3 V supply, VBAT is unable to source the start-up current required for the Battery RAM. Therefore, power consumption on the VBAT pin will be high and will remain high until 3.3 V supply is powered up. Once 3.3 V supply is powered up, power consumption on the VBAT pin will reduce to normal and subsequent power cycle on the 3.3 V supply will not cause an increased power consumption on the VBAT pin.

#### Work-around:

Provide 3.3 V supply used by rest of the device first and then provide VBAT voltage.

### 3.19 VBAT.2: The VBAT pin cannot be left floating

#### Introduction:

The device has a VBAT pin which provides power only to the Real Time Clock (RTC) and Battery RAM. VBAT can be connected to a battery or the same supply used by rest of the device ( $V_{DD(3V3)}$  pin,  $V_{DD(DCDC)(3V3)}$  pin). The input voltage range on the VBAT pin is 2.0 V minimum to 3.6 V maximum for temperature  $-40\text{ }^{\circ}\text{C}$  to  $+85\text{ }^{\circ}\text{C}$ . Normally, if the RTC and the Battery RAM are not used, the VBAT pin can be left floating.

#### Problem:

If the VBAT pin is left floating, the internal reset signal within the RTC domain may get corrupted and as a result, prevents the device from starting-up.

#### Work-around:

The VBAT should be connected to a battery or the same supply used by rest of the device ( $V_{DD(3V3)}$  pin,  $V_{DD(DCDC)(3V3)}$  pin).

### 3.20 ADC.2: External sync inputs not operational

#### Introduction:

In software-controlled mode (BURST bit is 0), the 10-bit ADC can start conversion by using the following options in the A/D Control Register:

26:24	START	When the BURST bit is 0, these bits control whether and when an A/D conversion is started:	0
000		No start (this value should be used when clearing PDN to 0).	
001		Start conversion now.	
010		Start conversion when the edge selected by bit 27 occurs on P2.10/EINT0.	
011		Start conversion when the edge selected by bit 27 occurs on P1.27/CAP0.1.	
100		Start conversion when the edge selected by bit 27 occurs on MAT0.1.	
101		Start conversion when the edge selected by bit 27 occurs on MAT0.3.	
110		Start conversion when the edge selected by bit 27 occurs on MAT1.0.	
111		Start conversion when the edge selected by bit 27 occurs on MAT1.1.	

Fig 1. A/D control register options

#### Problem:

The external start conversion feature, AD0CR:START = 0x2 or 0x3, may not work reliably and ADC external trigger edges on P2.10 or P1.27 may be missed. The occurrence of this problem is peripheral clock (pclk) dependent. The probability of error (missing a ADC trigger from GPIO) is estimated as follows:

- For PCLK\_ADC = 72 MHz, probability error = 12 %
- For PCLK\_ADC = 50 MHz, probability error = 6 %
- For PCLK\_ADC = 12 MHz, probability error = 1.5 %

The probability of error is not affected by the frequency of ADC start conversion edges.

#### Work-around:

In software-controlled mode (BURST bit is 0), the START conversion options (bits 26:24 set to 0x1 or 0x4 or 0x5 or 0x6 or 0x7) can be used. The user can also start a conversion by connecting an external trigger signal to a capture input pin (CAPx) from a Timer peripheral to generate an interrupt. The timer interrupt routine can then start the ADC conversion by setting the START bits (26:24) to 0x1. The trigger can also be generated from a timer match register.

### 3.21 USB.5: USB host controller hangs on a dribble bit

**Introduction:**

Full-/low-speed signaling uses bit stuffing throughout the packet without exception. If the receiver sees seven consecutive ones anywhere in the packet, then a bit stuffing error has occurred and the packet should be ignored.

The time interval just before an EOP is a special case. The last data bit before the EOP can become stretched by hub switching skews. This is known as dribble and can lead to a situation where dribble introduces a sixth bit that does not require a bit stuff. Therefore, the receiver must accept a packet for which there are up to six full bit times at the port with no transitions prior to the EOP.

**Problem:**

The USB host controller will hang indefinitely if it sees a dribble bit on the USB bus. It will hang the first time a dribble bit is seen. Once it is in this state there is no recovery other than a hard chip reset. This problem has no effect on the USB device controller.

**Work-around:**

None.

## 4. AC/DC deviations detail

---

### 4.1 ESD.1: The LPC2468 does not meet the 2 kV ESD requirements on the RTCX1 pin

**Introduction:**

The LPC2468 is rated for 2 kV ESD. The RTCX1 pin is the input pin for the RTC oscillator circuit.

**Problem:**

The LPC2468 does not meet the required 2 kV ESD specified.

**Work-around:**

Observe proper ESD handling precautions for the RTCX1 pin.

## 5. Errata notes detail

---

### 5.1 Note.1

On each of the following port pins P0.23, P0.24, P0.25, P0.26, P1.30, P1.31, P0.12, and P0.13 (when configured as general purpose input pin (s)), leakage current increases when the input voltage is  $V_i \geq V_{DD} I/O + 0.5$  V. Care must be taken to limit the current to less than 4 mA by using a series limiting resistor.

### 5.2 Note.2

On the LPC2468 Rev D, design changes to the Memory Accelerator Module were made to enhance timing and general performance. Design changes are intended to enhance performance in general and will result in minor differences in the code execution timing between the previous device revisions and rev D. Actual performance impact is code dependent, some code sequences may speed up while other code sequences may slow down between the previous device revisions and rev D. This might be observed when using software delays and in such cases, a hardware timer should be used to generate a delay instead of a software delay.

## 6. Legal information

### 6.1 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.

### 6.2 Disclaimers

**Limited warranty and liability** — 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.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

**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 life support, life-critical or safety-critical systems or equipment, nor in applications where failure or

malfunction of an 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 at 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.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

### 6.3 Trademarks

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

## 7. Contents

<b>1</b>	<b>Product identification</b>	<b>3</b>			
<b>2</b>	<b>Errata overview</b>	<b>3</b>			
<b>3</b>	<b>Functional problems detail</b>	<b>5</b>			
3.1	Ethernet.4: Ethernet TxConsumeIndex register does not update correctly after the first frame is sent.	5			
	Introduction:	5			
	Problem:	5			
	Work-around:	5			
3.2	ADC.1: ADDRx read conflicts with hardware setting of the DONE bit.	5			
	Introduction:	5			
	Problem:	5			
	Work-around:	5			
3.3	Ethernet.1: Setting up the Ethernet interface in RMII mode	6			
	Introduction:	6			
	Problem:	6			
	Work-around:	6			
3.4	Ethernet.2: Ethernet SRAM disabled	7			
	Introduction:	7			
	Problem:	7			
	Work-around:	7			
3.5	Ethernet.3: Receive Status registers will not function correctly if RxDescriptor number is greater than 4	7			
	Introduction:	7			
	Problem:	7			
	Work-around:	7			
3.6	I2S.1: I2S DMA interface is non-operational	7			
	Introduction:	7			
	Problem:	7			
	Work-around:	7			
3.7	PLL.1: PLL output (F <sub>CCO</sub> ) is limited to 290 MHz	8			
	Introduction:	8			
	Problem:	8			
	Work-around:	8			
3.8	SRAM.1: 16 kB SRAM cannot be used for code execution	8			
	Introduction:	8			
	Problem:	8			
	Work-around:	8			
3.9	USB.1: USB_NEED_CLK is always asserted.	9			
	Introduction:	9			
	Problem:	9			
	Work-around:	9			
3.10	USB.2: U1CONNECT signal is not functional.	9			
	Introduction:	9			
	Problem:	9			
	Work-around:	9			
3.11	USB.3: V <sub>BUS</sub> status input is not functional	9			
	Introduction:	9			
	Problem:	9			
	Work-around:	9			
3.12	USB.4: USB_PWRDx pin(s) does not function as intended	10			
	Introduction:	10			
	Problem:	10			
	Work-around:	10			
3.13	WDT.1: Accessing non-Watchdog APB registers in the middle of the feed sequence causes a reset.	10			
	Introduction:	10			
	Problem:	10			
	Work-around:	10			
3.14	Core.1: Incorrect update of the Abort Link register in Thumb state	10			
	Introduction:	10			
	Problem:	10			
	Conditions:	10			
	Work-around:	11			
3.15	Flash.1: Operating speed out of on-chip flash is restricted	11			
	Introduction:	11			
	Problem:	11			
	Work-around:	11			
3.16	MAM.1: Under certain conditions in MAM Mode 2 code execution out of internal flash can fail	12			
	Introduction:	12			
	Problem:	12			
	Work-around:	12			
3.17	CAN.1: Data Overrun condition can lock the CAN controller	12			
	Introduction:	12			
	Problem:	12			
	Work-around:	12			
3.18	VBAT.1: Increased power consumption on VBAT when VBAT is powered before the 3.3 V supply used by rest of the device	13			
	Introduction:	13			
	Problem:	13			
	Work-around:	13			
3.19	VBAT.2: The VBAT pin cannot be left floating	13			
	Introduction:	13			
	Problem:	13			
	Work-around:	13			

continued >>

3.20	ADC.2: External sync inputs not operational .	14
	Introduction: . . . . .	14
	Problem: . . . . .	14
	Work-around: . . . . .	14
3.21	USB.5: USB host controller hangs on a dribble bit . . . . .	15
	Introduction: . . . . .	15
	Problem: . . . . .	15
	Work-around: . . . . .	15
<b>4</b>	<b>AC/DC deviations detail . . . . .</b>	<b>16</b>
4.1	ESD.1: The LPC2468 does not meet the 2 kV ESD requirements on the RTCX1 pin. . . . .	16
	Introduction: . . . . .	16
	Problem: . . . . .	16
	Work-around: . . . . .	16
<b>5</b>	<b>Errata notes detail . . . . .</b>	<b>16</b>
5.1	Note.1 . . . . .	16
5.2	Note.2 . . . . .	16
<b>6</b>	<b>Legal information. . . . .</b>	<b>17</b>
6.1	Definitions . . . . .	17
6.2	Disclaimers . . . . .	17
6.3	Trademarks. . . . .	17
<b>7</b>	<b>Contents . . . . .</b>	<b>18</b>

---

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

---

© NXP B.V. 2011.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 1 June 2011

Document identifier: ES\_LPC2468