ERRATA SHEET

Date: November 8, 2004
Document Release: Version 1.4
Device Affected: LPC2124

This errata sheet describes both the functional deviations and any deviations from the electrical specifications known at the release date of this document.

Each deviation is assigned a number and its history is tracked in a table at the end of the document.

2004 November 8

Philips Semiconductors

PHILIPS

Identification:

The LPC2124 devices typically have the following top-side marking:

LPC2124xxx

xxxxxxx

xxYYWW R

The last letter in the third line (field 'R') will identify the device revision. This Errata Sheet covers the following revisions of the LPC2124:

Revision Identifier (R)	Comment
'A'	Initial device revision

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

Functional Deviations of LPC2124

IAP.1: Flash memory programming interface timing problem

Introduction: The Flash memory on the LPC2124 offers In-Application Programming (IAP) functionality. The IAP routines are part of the on-chip boot loader software, which controls the interface between the digital logic and the Flash memory. Please note that all programming methods (JTAG, ISP, IAP) use IAP calls.

Problem: Due to a timing problem in the interface between the Flash block and the digital logic the following problem may occur:

If the boot loader revision in the device is previous to V1.63 then in up to 10% of the devices the Flash memory interface, at some point during an IAP programming or erase operation, may never return from the IAP call. Please note that devices that pass the IAP programming are functional and do not suffer from any long-term reliability problems.

Devices with a date code prior to 0425 (manufactured before week 25 in 2004) are generally affected by this problem unless you receive devices with updated boot loader software from your distributor. Parts marked with date code 0425 or later are not affected by this problem. Please refer to page 2 of this document for details on how to identify the date code.

Workarounds:

1) The on-chip boot-loader software can be updated via ISP to correct this issue. The boot loader update files can be downloaded here:

http://www.semiconductors.philips.com/files/products/standard/microcontrollers/utilities/lpc2000_bl_update.zip

The boot-loader version can be read out using the Philips Flash ISP Utility which can be found here:

http://www.semiconductors.philips.com/files/products/standard/microcontrollers/utilities/lpc2000_flash_utility.zip

Both links can also be found on the product information page for this product under the Support & Tools section which can be found here:

http://www.semiconductors.philips.com/pip/LPC2124FBD64.html

2) Limiting the external clock frequency to 12 MHz AND making sure the on-chip PLL is turned OFF while programming any part of the Flash memory reduces the likelihood of the occurrence significantly. During In-System-Programming the PLL is turned off by default.

ADC.1: First two ADC conversions in burst mode from same channel

Introduction: In burst mode the A/D converter does repeated conversions at the rate selected by the CLKS field in the ADCR, scanning (if necessary) through the pins selected by 1s in the SEL field. The first conversion after the start corresponds to the least-significant 1 in the SEL field, then higher numbered 1-bits (pins) if applicable. Repeated conversions can be terminated by clearing this bit.

Problem: In burst conversion mode, the first two conversions (after setting the mode) will be of the same, lowest-numbered, of the selected channels.

work-around: Ignore the first conversion, always check the CHN bits to confirm the channel converted.

ADC.2: First conversion from channel specified by previous SEL setting

Introduction: The ADCR SFR contains bits to enable the ADC burst mode (BURST), start the conversion in software control mode (START), and to select the channel to begin converting (SEL).

Problem: In burst mode, If the BURST bit is set before or simultaneously to (using the STR instruction for example), the SEL bits, then the first channel converted will correspond to the previous SEL bit settings.

In software control mode (only when using external trigger), if the START bits are set before or simultaneously to (using the STR instruction for example) the SEL bits, then the first channel converted will correspond to the previous SEL bit settings.

work-around: Set the BURST/START bit(s) after setting the SEL bits.

ADC.3: Incorrect scan pattern

Introduction: In hardware scan mode multiple ADC channels may be selected as part of the scan by selecting the appropriate bits in the SEL field in the ADCR register.

Problem: Certain hardware scanning patterns for the A/D Converter do not operate properly. Selecting channel 2 only leads to alternate sampling of channels 2 and 3. Selecting channels 1 and 2 leads to sampling channel 1 for the first conversion, then sampling channel 2 on every subsequent conversion.

work-around: None. Do not use the sampling patterns "channel 2 only" or "channels 1 and 2". This problem has no effect on software conversion, nor on any other patterns other than the two noted above.

ADC.4 Global powerdown does not power down the ADC

Introduction: Setting the PD bit (bit 1) in PCON stops all clocks and powers down the peripherals. The ADC is powered down by clearing the PDN bit (bit 21) in the ADCR register, setting the bit powers up (enables) the ADC.

Problem: If the PDN in ADCR is set, setting the PD bit in PCON will not power down the ADC.

Work-around: Clear the PDN bit in the ADCR SFR to turn off the ADC prior to setting the PD bit in PCON.

ADC.5 Edge triggered ADC conversion start error

Introduction: When the START field of the ADCR register contains 010-111 the EDGE bit in ADCR will determine whether a conversion is started on a rising or falling edge of the selected CAP/MAT signal. EDGE=0 selects rising edge detection, EDGE=1 selects falling edge detection. (On CAP/MAT pin)

Problem: If the state of the selected CAP/MAT signal is 1 and EDGE is set to detect rising edges (EDGE = 0) or, if detection of falling edges is selected (EDGE = 1) and the state of the selected CAP/MAT signal is 0, an ADC conversion will immediately be initiated when the START bits are written to. So the first conversion behaves as a level triggered event rather than edge triggered.

Work-around: Clear the selected CAP/MAT signal for EDGE = 0 or set the selected CAP/MAT signal for EDGE = 1 before writing 010-111 to START field. Alternatively, discard the first conversion after writing to the start bits.

ADC.6 Writing to ADCR while conversion in progress

Introduction: Writing to ADCR while a conversion is in progress should set the DONE bit and start a new conversion.

Problem: In actuality, if the ADCR is written to within 2.5 ADC_clock cycles, a new conversion is started but the DONE bit is not set. If the ADCR is written to after 2.5 ADC_clocks, but within a conversion time, the DONE bit is set within one ADC_clock and a new conversion is started.

work-around: Do not write to ADCR until the conversion is complete.

SPI.1 Unintentional clearing of SPI interrupt flag

Introduction: The SPI interrupt flag is set by the SPI interface to generate an interrupt. It is cleared by writing a 1 to this bit.

Problem: A write to any register associated with the SPI peripheral will clear the SPI interrupt register.

work-around: Avoid writing to SPI registers while transmissions are in progress or while SPI interrupts are pending.

EXTINT.1 Corruption of VPBDIV via EXTPOLAR or EXTMODE

Introduction: The VPBDIV register controls the rate of the VPB clock in relation to the processor clock. EXTPOLAR and EXTMODE determine the operating parameters of the external interrupts.

Problem: A write to either the external interrupt polarity register (EXTPOLAR) or the external interrupt mode register (EXTMODE) will corrupt the VPBDIV register. A read of either EXTPOLAR or EXTMODE will be corrupted BY the VPBDIV register. If VPBDIV is “1” or “2” prior to any write to EXTPOLAR or EXTMODE, the CPU will hang up on the write to EXTPOLAR or EXTMODE.

work-around: If VPBDIV is non-zero, write all zeroes to VPBDIV before reading or writing EXTMODE or EXTPOLAR, then write the proper value back to VPBDIV. In most applications this is a known and fixed value, but if there is a possibility of dynamic changes in VPBDIV, software will need to read VPBDIV, write zero to VPBDIV, read or write EXTMODE and/or EXTPOLAR, and then rewrite the value previously read from VPBDIV.

CAP.1 Problem when selecting P0.21 as a capture 1.3 input (timer1)

Introduction: P0.21 and P0.19 may be configured as capture inputs via the PINSEL register.

Problem: When PINSEL(11:10) is set to "11" P0.21 is not internally connected as capture 1.3

Work-around: To use P0.21 as capture 1.3, PINSEL(7:6) must also be set to "11" which means that P0.19 must be selected as capture input 1.2.

VPBDIV.1 Incorrect read of VPBDIV

Introduction: The Peripheral Bus Divider (VPBDIV) divides the processor clock (CCLK) by one, two, or four. This is the clock that is provided to the peripheral bus.

Problem: Reading the VPBDIV register may return an incorrect value.

work-around: Performing two consecutive reads of the VPBDIV assures that the correct value is returned.

Timer.1 Missed Interrupt Potential

Introduction: The Timers may be configured so that events such as Match, Capture, and PWM, cause interrupts. Bits in the Interrupt Register (IR) indicate the source of the interrupt, whether from Capture or Match.

Problem: If more than one interrupt for multiple Match events using the same Timer are enabled, it is possible that one of the match interrupts may not be recognized. If this occurs no more interrupts from that specific match register will be recognized. This could happen in a scenario where the match events are very close to each other. This issue also affects the Capture functionality.

Specific details:

Suppose that two match events are very close to each other (Say Match0 and Match1). Also assume that the Match0 event occurs first. When the Match0 interrupt occurs the 0th bit of the Interrupt Register will be set. To exit the Interrupt Service Routine of Match0, this bit has to be cleared in the Interrupt Register. The clearing of this bit might be done by using the following statement:

T0_IR = 0x1;

It is possible that software will be writing a 1 to bit 0 of the Interrupt Register while a Match1 event occurs, meaning that hardware needs to set the bit 1 of the Interrupt Register. In this case, since hardware is accessing the register at the same time as software, bit 1 for Match1 never gets set, causing the interrupt to be missed.

In summary, while software is writing to the Interrupt Register, any Match or Capture event (which are configured to interrupt the core) occurring at the same time may result in the subsequent interrupt not being recognized.

Similarly for the Capture event, if a capture event occurs while a Match event is being is serviced then the Capture event might be missed if the software and hardware accesses coincide.

Affected features:

1. Interrupt on Match for Timer0/1.
2. Interrupt on Capture for Timer0/1.
3. These same features will be affected when using PWM.

Workaround: There is no clear workaround for this problem but some of the below mentioned solutions could work with some applications.

Possible workarounds for Match functionality:

1. If the application only needs two Match registers then distribute them between Timer 0 and Timer 1 to avoid this problem
2. Stop the timer before accessing the Interrupt register for clearing the interrupt and then start timer again after the access is completed.
3. Polling for interrupt: Supposing that there are two Match events (Match X and Match Y). At the end of the Interrupt Service Routine (ISR) for Match X, compare the Timer Counter value with the Match Register Y value. If the Timer Counter value is more than the Match Register Y value then it is possible that this event might have been missed. In this case jump to the ISR directly and service Match event Y.

Possible workarounds for Capture functionality:

1. Try to spread the capture events between both timers if there are two capture events. If the application also has a match event then one of the capture events may suffer
2. Polling for Capture: At the end of a Match interrupt ISR or Capture event ISR compare the previous Capture value with the current Capture value. If the Capture value has changed then the Capture event might have been missed. In this case, jump to the ISR directly and service the Capture event.

The same above workarounds will also be applicable for the Match and Compare functionality of the PWM block.

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 PCrelative 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 is would have to be done manually.

Electrical and Timing Specification Deviations of the LPC2124

V_{IH}.1 Incompatibility of actual V_{IH} levels as compared to those specified

Introduction: The specified, minimum, value for V_{IH} is 2.0V.

Problem: Any pin associated with either an external interrupt input or an analog to digital converter (ADC) input has a V_{IH} of 2.4V, not 2.0V. The pins that are affected are the ones that can be configured as either an ADC input or and external interrupt input, not just the ones that are configured as such.

work-around: Make sure that high logic levels are at least 2.4V at these pins.

Errata History - Functional Problems

Functional Problem	Short Description	errata occurs in device revision
IAP.1	No return from IAP erase/program call	A
ADC.1	First two ADC conversions in burst mode from same channel	A
ADC.2	First conversion from channel specified by previous SEL setting	A
ADC.3	Incorrect scan pattern	A
ADC.4	Global powerdown does not power down the ADC	A
ADC.5	Edge triggered ADC conversion start error	A
ADC.6	Writing to ADCR while conversion in progress	A
SPI.1	Unintentional clearing of SPI interrupt flag	A
EXTINT.1	Corruption of VPBDIV via EXTPOLAR or EXTMODE	A
CAP.1	Problem when selecting P0.21 as a capture 1.3 input (timer1)	A
VPBDIV.1	Incorrect read of VPBDIV	A
CORE.1	Incorrect load of the link register	A
Timer.1	Missed Interrupt Potential	A

Errata History - Electrical and Timing Specification Deviations

AC/DC Deviations	Short Description	errata occurs in device revision
V_{IH}.1	Incompatibility of actual V_{IH} levels as compared to those specified	A