19-5458; Rev 4; 11/11

ERRATA SHEET
MAXQ622

Revision B1 Errata

The errata listed below describe situations where MAXQ622 revision B1 components perform differently than expected or differently than described in the data sheet. Maxim Integrated Products, Inc., intends to correct these errata when the opportunity to redesign the product presents itself.

*This errata sheet only applies to MAXQ622 revision B1 components. Revision B1 components are branded on the topside of the package with a six-digit code in the form yywwB1, where yy and ww are two-digit numbers representing the year and work week of manufacture, respectively. To obtain an errata sheet on another MAXQ622 die revision, visit our website at **www.maxim-ic.com/errata**.*

1) WATCHDOG TIMER RESET LOCKS THE IC IN RESET

Description:
When the internal oscillator (PLL) is used in divide-by-4 mode as system clock, the watchdog timer reset locks the IC in reset. This requires a power cycle to recover.

Workaround:
Prevent watchdog timer resets by regular monitoring of the watchdog or by using the watchdog interrupt to switch the system clock away from internal oscillator (PLL).

2) FLASH PROGRAM/ERASE OPERATIONS FAIL IN CLOCK-DIVIDED MODE

Description:
Flash word programming and page erase operations do not complete successfully when invoked with clock divide-by-2, divide-by-4, or divide-by-8 modes active.

Workaround:
When using the flash controller directly, write and erase functions exercised through the FCNTL and FDATA registers must be performed with a nondivided system mode setting (see the CKCN register description). In addition, the following utility ROM function calls access the FCNTL and FDATA registers and have the same requirement prior to being called: UROM_flashWrite, UROM_flashErasePage, and UROM_flashEraseAll. The system clock must be set to divide-by-1 mode before calling any of these routines.

3) SYSTEM CODE IS INADVERTENTLY ERASED

Description:
When user loader and user application code is loaded in upper flash memory (with address greater than 0x7FFF) with memory protection enabled, the system code is inadvertently erased.

Workaround:
Avoid use of memory protection (or) make sure the code with the lowest memory privilege starts below address 0x8000.

4) MEMORY PROTECTION FAULT ASSERTED

Description:
When entering debug mode in upper memory (IP > 0x7FFF, SC_UPA = 1) the debug engine incorrectly drops the privilege level to the lowest state. If the area of memory being debugged requires a higher privilege in order to be accessed, a memory fault is generated upon return from the debug code.

Workaround:
Avoid use of memory protection (or) avoid debug in the upper memory (or) by allowing only UAPP code in the upper memory.

5) V_{DDIO} POWER SUPPLY OSCILLATES UPON A USB DISCONNECT

Description:
When the V_{DD} input power is above the V_{DDB} trip point (> 2.7V) with V_{BUS} open-circuited, the V_{DDIO} power supply oscillates.

Workaround:
By using force V_{DD} (FRCVDD) the power switching can be prevented. Switch FRCVDD on and off depending on the value of V_{DD} by monitoring the SVM to prevent power switch when V_{DD} is greater than V_{DDB}.

6) UNRELIABLE POWER-UP WHEN ONLY V_{BUS} IS APPLIED

Description:
In some instances when the chip is powered up with just V_{BUS} supply, the power-up sequence is not properly executed.

Workaround:
V_{DD} supply should be in place and stable before V_{BUS} is applied.

7) HIGH V_{BUS} STANDBY CURRENT

Description:
A high standby current is seen on V_{BUS} with USB connected under the following conditions:
FRCVDD = 1 and when $V_{DD} < V_{DDB}$ - 0.5V.

Workaround:
Setting FRCVDD to a logic-low when the above condition exists prevents a high standby current.

19-5458; Rev 4; 11/11

8) PUSH INSTRUCTION CAUSES CODE EXECUTION ERROR WHEN FOLLOWED BY READ OF CERTAIN REGISTERS, OR IF THE MOD[1:0] BITS HAVE BEEN CHANGED FROM THEIR DEFAULT VALUES

Description:
A PUSH instruction, followed immediately by an instruction reading one of the following registers, I2CBUF, UDATA, causes a code execution error. In addition, changing the MOD[1:0] bits (APC[2:0]) from their default value can cause the device to operate incorrectly any time a PUSH instruction is followed by any instruction which reads the active accumulator.

Workaround:
The workaround is dependent on whether the programming language is C or assembly.
C code:
1. IAR Embedded Workbench: Code compiled by IAR Embedded Workbench versions 2.12 and earlier is not affected by this erratum. Versions after 2.12 must add the following line at the beginning of any function that accesses the peripherals mentioned above:

asm (“NOP”)

2. Rowley CrossWorks: Code compiled by Rowley CrossWorks versions 2.0 and earlier is not affected by this erratum. Versions after 2.0 must add the following line at the beginning of any function that accesses the peripherals mentioned above:

__Insert_Opcode(0xDA3A)

Note that the line is preceded by two underscore characters.

Assembly code:
Place a NOP after every PUSH instruction that is immediately followed by a read of one of the previously mentioned registers.

In addition, the user must ensure that software never changes the MOD[1:0] bits of the AP register from their default value of 00b.

9) USE OF INTERRUPT PRIORITY FEATURE CAN CAUSE INCORRECT PROCESSING OF INTERRUPTS

Description:
An interrupt that is followed immediately by a higher priority interrupt can cause the CPU to incorrectly service the interrupts.

Workaround:
Do not use high-priority interrupts. Do not modify the IPR0 or IPR1 registers from their reset value. This sets all interrupt sources to the lowest (default) priority level.

10) I²C PERIPHERAL DOES NOT FUNCTION CORRECTLY

Description:
The I²C peripheral does not function correctly.

Workaround:
Do not use the I²C peripheral on this revision of the device..

REVISION HISTORY

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	7/10	Initial release	—
1	9/10	Reworded the workaround statement for erratum #3 to ensure clarity	2
2	11/10	Added erratum #8 (PUSH Instruction)	3
3	6/11	Added erratum #9 (Interrupt Priority)	3
4	11/11	Added erratum #10 (I^2C Peripheral)	4