

UTMC Errata Sheet

UT80CXX196KD MicroController HSI Interrupt and I/O Status Bits Anomalies

The following anomalies are related to the UT80CXX196KD MicroController designs identified with either JD02A and JD02B on the lid marking (SMD #5962-98583).

Anomaly #1:

When INT02 is used as the HSI Data Available interrupt source, INT02 interrupts will not occur if any events remain in the HSI FIFO or holding register at the time a new HSI event occurs.

Details:

The HSI_RDY status bit of IOS1.7 *must be* in a low state in order for a new HSI event to set the INT_PEND bit and generate INT02. The pending interrupt register receives a single input pulse signal to set a bit. For the HSI Data Available interrupt, this pulse occurs only when the HSI_RDY (IOS1.7) is first set high (when data is loaded into the first FIFO location (Holding Register)). Therefore, the HSI_RDY status *must be cleared* before another input event will cause an interrupt.

Work around:

1. Do not use the HSI Data Available source for INT02, instead use FIFO Full as the INT02 interrupt source.
2. Use the HSI Data Available interrupt source only with a standard interrupt routine (not PTS) and in the routine perform a loop on reads of the HSI_STATUS and HSI_TIME until the IOS1.7 status bit is cleared (HSI_RDY). Perform this loop of reads when switching from another HSI interrupt mode to the HSI Data Available INT02 mode.

Anomaly #2:

The IOS1 and IOS2 status bits, which are cleared during a read, may miss reporting an event. This anomaly occurs when the read occurs at precisely the same time as an incoming status event. This "collision" gives priority to the clearing of the bit which is requested by the read cycle.

Details:

The affected bits are the Software Timer Flags and Timer Overflow Flags in IOS1 and the HSO Event Flags and T2RST Flag of IOS2. The corresponding interrupts will operate properly.

Work around:

The recommended work around is to develop code which does not depend on these status bits. A secondary work around is to carefully read these registers at a point in time where an event is not defined to occur.

1. Example of setups to avoid using the status bits:

- Software Timer Flags:

1. If only one of the four software timers is used, reading of the software timer flags is not required. If the interrupt will not be enabled, the INT_PEND register can report the overflow status (in this case a read-modify-write operation (ANDB) is required to clear the pending bit when it is found to be set).
2. For use of multiple software timers, instead of reading IOS1, read the associated timer value to determine which software timer has expired when the interrupt occurs.

- Timer Overflow Flags:

Use INT00 for Timer 1 only and use INT12 for Timer 2. This allows the INT_PEND and INT_PEND1 registers to report the overflow status of each Timer if it is not desired to enable the interrupts (in this case a read-modify-write operation (ANDB) operation is required to clear the pending interrupt bit when it is found to be set).

UTMC Errata Sheet

- HSO Event Flags:

1. If only one HSO event has the interrupt enabled, reading of the HSO Event Flags is not required. If the interrupt will not be enabled, the INT_PEND register reports the HSO Event status (in this case a read-modify-write operation (ANDB) operation is required to clear the pending bit when it is found to be set).
2. For use of multiple HSO events triggering an interrupt, instead of reading IOS1, read the associated timer value to determine which HSO event occurred when the interrupt occurs.

- T2RST Flag:

1. If software timers are not enabled, reading of the T2RST flag is not required. Otherwise, the INT_PEND register can report the T2RST status if it is not desired to enable the interrupt (in this case a read-modify-write operation (ANDB) operation is required to clear the pending bit when it is found to be set).
2. For use of software timers, as well as an interruptible Timer2 Reset via HSO, read the associated timer value to determine which event occurred when the interrupt occurs.

2. Example of setup to read the IOS1 & IOS2 registers at a point in time where an event is not defined to occur:

- Define all HSO events (for which the application requires an accurate IOS status) to occur when the timer count does not end in XXXC or XXXD (or 2 other sequential counts). When it is desired to read the IOS status, disable interrupts, then wait until the associated timer reaches the first "reserved" count (i.e., XXXC), then perform the IOS read.

Anomaly #3:

The SFR register IOS2 is defined as a status register with write capability. An anomaly exists causing IOS2 bit 6 (T2RST_EVENT) to not accept a state change when written to by the user. (Bits 0 through 5 are writable as specified, and all bits 0 through 6 are cleared as specified when IOS2 is read. Also, the HSO event correctly sets bit 6 when required.)

Work around:

None--Do not expect writes into IOS2.6 to be latched.