

UTMC Errata Sheet

UT80CXX196KD (JD02X) NORML Instruction Operation

Anomaly:

There exists an anomaly with the “A,” “B,” and “C” revisions of the JD02 die (UT80CXX196KD) that the normalize (NORML -- Opcode 0Fh) instruction does not function properly with negative numbers. Under correct operation, the NORML instruction shifts a 32-bit operand to the left until the most significant bit is a “1”. When the NORML instruction completes the shift, stores a count for the number of shifts that have occurred into a destination register. When the 32-bit operand is negative, the most significant bit will already be set and the NORML instruction should not perform a shift and loads a “0” into the destination operand. For the UT80CXX196KD revisions “A”, “B”, and “C” the NORML instruction performs a single shift before checking for the most significant bit being set, and therefore loads a “1” into the destination operand.

Background:

The NORML instruction is frequently used in software that emulates floating point functionality. If your software application is written in C/C++ and performs floating point operations (e.g. $\sin(x)$, $\cos(x)$, etc.), then there exists a strong possibility that you will execute software that includes the NORML instruction. UTMC has identified a software work-around for this anomaly and is in the process of making another revision to the UT80CXX196KD. The software solution can be performed manually through ROM based debug software, or through post-processing software provided by UTMC.

Anomaly Solution:

The software solution to the NORML instruction anomaly requires the user to replace all instances of the NORML instruction with the LCALL instruction consisting of the same number of bytes as the NORML instruction and manually implement a normalize emulation subroutine in machine hex code. An assembly example of the software work-around is shown below:

NORML Instruction Correction Example:

```
LCALL    replace_norml    ;replaces the NORML instruction

replace_normal:          ;start of the NORML instruction emulation subroutine for example operands
                        40h, 22h

JBC     43h, 7, do_norml ;if the operand is not negative, then perform the NORML instruction

OR      42h, zero_reg    ;clear the Carry, Zero, and Overflow flags

ST      00h, 22h        ;return the number of shifts

                        ;NOTE: The NORML instruction does not affect the overflow flag. This
                        ;      subroutine clears the overflow flag if the operand is negative.
                        ;      Therefore, ensure that no JV or JVT instructions following the
                        ;      NORML instruction are dependent on an Overflow flag set prior
                        ;      to the NORML instruction.

RET                                           ;return from subroutine

do_norml:

NORML   40h, 22h        ;Execute the intended NORML instruction

RET                                           ;return from subroutine
```

UTMC Errata Sheet

The above subroutine facilitates proper operation of the floating point instructions available in the C/C++ language. UTMC understands this is manually intensive to perform the above software solution and has developed a post processing application to automatically perform the software fix for the user. The post processor reads in the .hex output file generated by the C/C++ compiler/assembler/linker, and replaces all occurrences of the NORML instruction with a LCALL to a subroutine that will also be added to the user's .hex file. This post processor allows users the ability to continue with software development while UTMC completes the migration of the UT80CXX196KD from revision "C" to revision "D". The "D" revision of the UT80CXX196KD will be available for production orders in June 2000. If you would like a copy of the post processing software, please visit the UTMC website (<http://www.utmc.com>) where this errata resides and download the software.