



Micromega Corporation

Release Notes

uM-FPU64 Firmware

Release 408

Changes in Release 408

New Features and Changes

- SELECTMA, SELECTMB, SELECTMC: fixed problem with using memory pointers to select matrices
- FTOA: fixed problem that could cause the FPU to hang when converting certain floating point numbers that required rounding of trailing nines
- XOP: new extended opcode instruction added
- READVAR: added additional entries
 - 21 - number of functions programmed in Flash
 - 22 - number of XOP instructions programmed in Flash
 - 23 - error status word
- added error status word to report internal error (see description below)
- when NaN is returned by an instruction the NaN bit is now set in the status byte
- added table of contents to uM-FPU64 Datasheet

XOP *Execute extended opcode instruction stored in Flash memory*

Syntax: **XOP, xop_number, arg1, arg2, arg2**

Description: Executes the extended opcode instruction specified by *xop_number*, using arguments *arg1*, *arg2*, *arg3*. The value of the arguments depends on the XOP instruction. Separate documentation for XOP library files specify the arguments required for each XOP. All XOP instructions have three argument bytes. If an XOP requires fewer than three arguments, zero bytes are used as fillers.

The uM-FPU64 IDE software is used to load the code for XOP instructions into Flash memory.

Opcode: **FE**

Byte 2: **xop_number**
An 8-bit value specifying the XOP number which is used to determine the location of the XOP ode programmed in Flash memory.

Byte 3: **arg1**
An 8-bit value used as the first argument for the XOP instruction.

Byte 4: **arg2**
An 8-bit value used as the second argument for the XOP instruction.

Byte 5: **arg3**
An 8-bit value used as the third argument for the XOP instruction.

Examples:

ASM: XOP, 01, 10, 20, 30 ; call XOP 1 with arguments 10, 20, and 30

Compiler: qa[4] equ F%
 qb[4] equ F%
 qc[4] equ F%
 q_add(qa, qb, qc) ; call XOP q_add with quaternions qa, qb, qc

READVAR *Read internal variable*

Syntax: **READVAR, item**

Description: Sets register 0 to the internal value selected by *item*.

...	
21	Number of Functions loaded in Flash
22	Number of XOP Instructions loaded in Flash
23	Error Word

Error Status Word

Bit	7	6	5	4	3	2	1	0
	-	Paren Level	Function Level	Address	XOP Call	Function Call	Under run	Over run

Bit 6 Parenthesis Level Error

The maximum temporary register level (8) was exceeded by a **LEFT** instruction. The temporary register level is reset to zero. **RIGHT** instructions will return NaN.

Bit 5 Function Level Error

The maximum function call level (16) was exceeded. The **FCALL** is aborted and NaN is returned in register 0 (32-bit) or register 128 (64-bit).

Bit 4 Address Error

An address error occurred within an **XOP** instruction.

Bit 3 XOP Call Error

An **XOP** instruction was executed that specified an extended instruction that was not programmed into Flash memory.

Bit 2 Function Call Error

An **FCALL** instruction was executed that specified a function that was not programmed into Flash memory.

Bit 1 Underrun Error

An instruction required additional bytes that were not received before a 5 msec timeout. A data byte of zero was returned.

Bit 0 Overrun Error

The maximum size of the instruction buffer (256 bytes) was exceeded. One or more data bytes were lost.