John Neil & Associates

America Online: John Neil
P.O. Box 160699
CompuServe: 70421,730
Cupertino, CA 95016-0699
internet:johnneil@netcom.com

Software FPU™

Programmer Information

John Neil
January 17, 1993

What SoftwareFPU Does

SoftwareFPU is a control panel which allows programs that expect a Motorola 68881/68882 Floating Point Unit (FPU) to work correctly without one. It is intended to be used in any 68020 or 68030 Macintosh computer without an FPU. Since both SoftwareFPU and the FPU perform IEEE Standard arithmetic, the differences between the hardware FPU and SoftwareFPU are minimal. The current known differences between SoftwareFPU and a hardware FPU are:

John Neil & Associates

America Online: John Neil
P.O. Box 160699
CompuServe: 70421,730
Cupertino, CA 95016-0699
internet:johnneil@netcom.com

- FRESTORE does not support the busy state frame.
- FMOD produces the same result as FREM.
- Mid-instruction exceptions are reported as post-instruction exceptions.
- If an exception occurs in trace mode, two instructions will execute before control returns to the debugger.
- Code which puts data below the stack pointer and then issues an FPU instruction will not work. This of course is an extreme no-no since data below the stack pointer can be clobbered by interrupt routines as well.
- Some emulated FPU instructions may produce more accurate results than on a hardware FPU.

The current list of known application incompatibilities are:

- Any program which patches the F-line exception vector itself will not work with SoftwareFPU. A development environment application would need to check for the presence of SoftwareFPU and patch a location inside the

America Online: John Neil
P.O. Box 160699
CompuServe: 70421,730
Cupertino, CA 95016-0699
internet:johnneil@netcom.com

emulator to trap F-line exceptions. That location is 2 + the value of 68000 F-line exception vector.

- Programs using recursive algorithms may have problems. SoftwareFPU can require up to 600 bytes of stack space per FPU instruction. Recursive programs will probably require more stack space than usual to work properly, if they will work at all.

America Online: John Neil
P.O. Box 160699
CompuServe: 70421,730
Cupertino, CA 95016-0699
internet:johnneil@netcom.com

- A bug in the MPW 3.1 nan() function means that any program calling nan() will not work. Because of this, the MPW functions atan2(), asin(), and acos() will not work when they try to produce QNANs. Also, fscanf will not read in NANs correctly. The bug has been fixed in MPW 3.2.
- Code which assumes that calls to SANE affect the FPU will not work. The MPW functions sinh() and cosh() may produce incorrect results in the exception status register, as the library code makes this assumption.

SoftwareFPU takes up about 16K in the system heap when installed.

Performance

SoftwareFPU performance ranges from 30% to 85% of calling SANE directly, depending on the instruction type. However, applications typically do not slow down this much

America Online: John Neil
P.O. Box 160699
CompuServe: 70421,730
Cupertino, CA 95016-0699
internet:johnneil@netcom.com

since they do not execute FPU instructions 100% of the time.

With SoftwareFPU installed, applications using direct FPU calls obtain a different performance tradeoff than using SANE. Take the performance of SANE on a non-FPU machine as a benchmark. SANE works about 10 times faster on machines with FPUs. Direct FPU calls run about 100 times faster on machines with FPUs, and 15-70% slower with SoftwareFPU on non-FPU machines.

SysEnvirons/Gestalt Patch

The emulator patches Gestalt to indicate the presence of an FPU to applications. While this maximizes the number of programs that work, it would slow down the few programs that can perform either SANE or FPU calls because they will use the software FPU instead of SANE. The known applications of this type are DataDesk, Excel, Resolve, Lotus 1-2-3, and the Apple system software. The Gestalt patch checks to see if the current application is one of these, and if so reports no FPU present.

John Neil & Associates

America Online: John Neil
P.O. Box 160699
CompuServe: 70421,730
Cupertino, CA 95016-0699
internet:johnneil@netcom.com

68LC040 CPU Bug

The 68LC040 bug is that with certain instructions streams, pending CPU writes to memory never arrive when an F-Line exception occurs. Most Macintosh applications are not affected by the bug, since an F-Line exception is normally a fatal error. However, software that depends on F-Line exceptions, such as SoftwareFPU, will not work properly. There is no known work-around for the bug once an exception has occurred (the only possible work-around for SoftwareFPU). At the application level, putting a NOP in front of every F-Line instruction will eliminate the problem, but will reduce performance on machines with hardware FPUs.

I have encountered this bug personally while debugging SoftwareFPU on a Centris 610, and have confirmed it through Motorola application support. The problem prevents

John Neil & Associates

America Online: John Neil
P.O. Box 160699
CompuServe: 70421,730
Cupertino, CA 95016-0699
internet:johnneil@netcom.com

Infini-D 2.5.1 and RayDream Designer 2.0 from working properly (they work fine on a regular 68040 Macintosh), and undoubtedly others. Some programs seem to work OK with SoftwareFPU on a 68LC040 Macintosh. Their instruction streams are apparently not succeptible to the bug. The bug does not occur in the 68LC040 emulator on PowerPC systems.