John Neil & Associates

internet:johnneil@netcom.com P.O. Box 2156
CompuServe: 70421,730
Cupertino, CA 95015
America Online: John Neil
eWorld: John Neil

Software FPU™

Programmer Information

John Neil
July 18, 1994

What SoftwareFPU Does

SoftwareFPU is a control panel which emulates a Motorola 68881 Floating Point Unit (FPU). It is intended to be used in any 68020, 68030, or 68LC040 Macintosh computer without an FPU. It can also be used on Power Macintosh machines, since the Apple 68K emulator does not emulate FPU instructions. It works by patching out the F-Line exception vector of the machine with one that points to FPU emulation code in SoftwareFPU.

Differences Between a 68881 FPU and SoftwareFPU

The differences between the hardware FPU and SoftwareFPU are minimal. The current differences between SoftwareFPU and a hardware FPU are:

- 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 is of course a no-no since data below the stack pointer can be

internet:johnneil@netcom.comP.O. Box 2156
CompuServe: 70421,730
Cupertino, CA 95015
America Online: John Neil
eWorld: John Neil

clobbered by interrupt routines as well.
- Some emulated FPU instructions may produce slightly different results than on a hardware FPU.

In addition, on PowerPC machines in 64-bit accuracy mode the following differences exist:

- The rounding precision in the FPU Mode Control Byte is ignored. All calculations are performed as if double-precision rounding was selected in the Mode Control Byte.
- For performance reasons, the PowerPC emulator cannot emulate address and bus errors properly. If an FPU instruction causes an address or bus error, the emulator will break into MacsBug at a location inside the emulator, rather than at the offending FPU instruction.

internet:johnneil@netcom.comP.O. Box 2156
CompuServe: 70421,730
Cupertino, CA 95015
America Online: John Neil
eWorld: John Neil

The current list of known application incompatibilities are:

- Any program which replaces the F-line exception vector will not work with SoftwareFPU. A typical example is a source-level debugger in a development system like MPW or THINK C/Symantec C++. SoftwareFPU 3.0 provides a new facility so applications like this can detect whether SoftwareFPU is installed, to determine whether or not the F-Line exception vector should be replaced (see below).
- 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.
- Code that depends on 68882 or 68040 FPU stack frames or instructions will not work. SoftwareFPU currently only emulates a 68881 FPU, even on 68LC040 CPUs. Applications that depend upon a particular FPU should verify the type of FPU installed with the `gestaltFPUType` Gestalt selector.

Performance

Because of the overhead in emulating all the intricacies of the FPU, executing FPU instructions through SoftwareFPU is slower than calling SANE. SoftwareFPU is aware of existing applications that can switch between hardware or software floating-point (such as Microsoft Excel), and does not report the presence of an FPU, so they will not slow down with SoftwareFPU installed. However, if you are writing a new application with this functionality, you should check whether SoftwareFPU is installed using the new facility in SoftwareFPU 3.0 (see below) before selecting hardware floating point.

internet:johnneil@netcom.comP.O. Box 2156
CompuServe: 70421,730
Cupertino, CA 95015
America Online: John Neil
eWorld: John Neil

Detecting the Presence of SoftwareFPU

SoftwareFPU adds a new Gestalt selector when it is installed in the system. To determine whether SoftwareFPU is installed, simply make the following Gestalt call:

```
long response;

result = Gestalt('FPUE', &response);
```

If the result of the call is `noErr`, then SoftwareFPU is installed, and the 68881 FPU reported by the `gestaltFPUType` selector is an emulated FPU. The value returned in the response is private to John Neil & Associates, and should be ignored.

68LC040 CPU Chip Bug

The 68LC040 chip bug mentioned in the user documentation 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

internet:johnneil@netcom.com P.O. Box 2156
CompuServe: 70421,730
Cupertino, CA 95015
America Online: John Neil
eWorld: John Neil

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 that SoftwareFPU could use). At the application level, putting a NOP in front of every F-Line instruction will eliminate the problem, at the expense of reducing performance on machines with hardware FPUs. Despite this bug, some FPU applications still work with SoftwareFPU on 68LC040 machines, because their instruction streams do not trigger the bug. For more information on the bug, please contact Motorola Semiconductor Inc.