New Technical Notes
Macintosh

Developer Support

Cooperating with the Coprocessor

Hardware

Revised by: Jim Reekes October 1990
Written by: Jim Reekes June 1989

The use of the 68881 or 68882 coprocessor is usually handled by the SANE package or by a development system's libraries. Some developers may wish to use the coprocessor during special circumstances, such as at interrupt level or installing their own hardware floating point exception handlers. In these two situations, there are special requirements that must be met. These requirements will require floating–point assembly code and are discussed in this Technical Note.
Changes since June 1989: Noted that the new Sound Manager no longer uses floating-point numbers at interrupt time.

Witnessing the Problem

If you see the message "Spurious Interrupt of the Uninitialized Interrupt vector" in MacsBug or the message "Unassigned Interrupt #$00D (format 9)" from TMON™, you should suspect a floating point protocol violation. This can be caused by improper usage of floating point instructions at interrupt level or by attempting to handle hardware floating point exceptions incorrectly.

Interrupting the Coprocessor

If you attempt to use the coprocessor at interrupt level, you may be interrupting a floating point processor. You must save the coprocessor's state before executing any floating point instructions, and, of course, restore it later during the interrupt routine. This requires assembly code since there is no convenient way to do it in a high–level language.

There is a protocol that must be followed. The first floating point instruction must be an `FSAVE`. This instruction suspends the execution of any operation in progress and saves the internal state. The number of bytes required in this operation depends on the state it is in, and it can be up to 216 bytes. If any floating point registers are to be used, they also must be saved with the `FMOVE` instruction. After performing the interrupt routine, the `FRESTORE` instruction is used to restore the floating point state.

```
VBLProc    FSAVE       -(SP)           ; save the FP state
         FMOVEM.X    FPRegs,-(SP)    ; save the FP regs we use
         …
         …                           ; your interrupt code
```

...

```
VBLExit     FMOVEM.X     (SP)+,FPRegs     ; replace the FP regs we used
          FRESTORE    (SP)+            ; restore the FP state
          RTS
```

Note that the coprocessor may not be in a condition to be interrupted, and the `FSAVE` instruction will halt the main processor until such a condition can be met. To give an idea on the time required for saving the coprocessor's state, the `FSAVE` or `FRESTORE` instructions can take approximately 900 cycles to execute. This is about 50 times slower than a `MOVE` instruction. Considering the length of time it takes to perform this necessary protocol, it may not be desirable to use floating–point math at interrupt level. As an alternative, investigate the possibility of using the Toolbox routines for `Fixed` and `Frac` numbers.

Handling Floating Point Exceptions

It is possible, and sometimes desirable, for applications to install their own hardware floating–point exception handlers. The MPW '881 SANE libraries provide routines for applications to do so. If an application is going to use this mechanism to catch exceptions such as underflow, overflow, or divide by zero, then it must follow the minimal protocol as shown in the following example.

```
Handler     FSAVE       -(SP)            ; save the FP state
          MOVE.B      (SP),D0          ; first byte of the state frame
          BEQ         NULL             ; branch if NULL state
          CLR.L       D0               ; clear data register
          MOVE.B      1(SP),D0         ; load state frame size
          BSET        #3,(SP,D0)       ; set bit 27 or BIU
          …
          …                            ; your exception code
          …
Null        FRESTORE    (SP)+            ; restore the FP state
          RTE                          ; return from exception
```

Other Issues

Debugging floating–point routines with MacsBug, SADE, and TMON may cause a protocol violation. MacsBug 6.1, and earlier, do not perform the `FSAVE` and `FRESTORE` surrounding floating–point instructions at interrupt level. As of TMON 2.8.4, it has not handled floating-point instructions. Hardware floating–point exception handlers and interrupt routines using floating–point instructions require assembly coding.

You can witness a protocol violation in another situation. This is when using the Sound Manager in System Software 6.0.5 and earlier. This Sound Manager calls SANE at interrupt time. If an application is using the coprocessor **and** this Sound Manager is running, it is very likely to interrupt the coprocessor. This problem has been resolved in the new Sound Manager, which was originally released in System Software 6.0.6. The new Sound Manager no longer uses floating-point numbers at interrupt level; it replaces them with `Fixed` and `Fract` types.

Further Reference:

- Apple Numerics Manual, Second Edition
- Motorola MC68881/MC68882 User’s Manual
- MPW reference manuals
- Technical Note M.PT.MPWmc68881, Notes on MPW’s –mc68881 Option
- Technical Note M.HW.SpeedyMathCoProc, Speedy the Math Coprocessor

TMON is a trademark of ICOM Simulations, Inc.