

Floating Point Instructions and the IBM 486DX2 Microprocessors



Application Note

Author: Scott Pheasant

Introduction

This paper explains the instruction sequence necessary to initialize floating point instructions on the IBM 486DX2 processors and the instructions necessary for saving the state of the coprocessor during SMM.

Many complex mathematical calculations require that numbers which include decimal places be used to insure accurate final results. Originally, computer systems that incorporated a floating point instruction execution unit (more commonly called a math coprocessor), did so by means of a chip that was external to the microprocessor. As technology improved and system performance became more important, the math coprocessor was placed onboard the microprocessor itself. This was the beginning of the 486DX class of processors. By having the coprocessor packaged with the microprocessor, there were no dependencies on the bus speed for data transfer between the two units. Therefore, the results from the floating point unit's calculations could be quickly transferred to the microprocessor.

Because the math coprocessor is still technically a silicon chip separate from the microprocessor silicon chip, it must be treated as such. Before executing any floating point instruction or during System Management Mode, there are special instruction code sequences that must be followed to insure that proper data handling and instruction executions occur.

Floating Point Instructions, RESET, and the IBM 486DX2

In general, whenever the IBM 486DX2 processor comes out of a RESET, the programmer must first initialize the floating point unit before executing any floating point instruction. To perform the FPU initialization, the FINIT instruction must be placed prior to any other floating point instruction in a program stream.

The above description holds true for IBM 486DX2 processor revisions 4.1 and below. However, beginning with revision 4.2 of the IBM 486DX2, whenever a programmer wishes to use the FRSTOR instruction (after a RESET), the FINIT instruction does not need to be executed before the FRSTOR instruction.

Excluding the FRSTOR instruction, all other numeric processing instructions require the FINIT instruction to be executed first after a RESET.

Floating Point Instructions, SMM, and the IBM 486DX2

To insure that data integrity, with respect to Floating Point Instructions, is maintained during SMM, there is a special instruction procedure which must be followed.

To save the state of the FPU during SMM, the FNSAVE instruction must be used. This is further explained in the "Maintaining FPU State" in the SMM section of the IBM 486DX2 Data Book.

The FPU state can be restored to the processor by executing the FRSTOR instruction. As stated earlier, if the FRSTOR instruction is executed following a RESET, revisions 4.1 and below require the FINIT instruction to be executed first.

IBM Corporation 1995. All rights reserved.

IBM and the IBM logo are registered trademarks of International Business Machines Corporation. IBM Microelectronics is a trademark of the IBM Corp.

All other product and company names are trademarks/registered trademarks of their respective holders. 1995 IBM Corp.

This document may contain preliminary information and is subject to change by IBM without notice. IBM assumes no responsibility of liability for any use of the information contained herein. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties.

The products described in this document are not intended for use in implantation or other direct life support applications where malfunction may result in physical harm or injury to persons.

NO WARRANTIES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE OFFERED IN THIS DOCUMENT.