Published: 06/05/68

Identification

Hardware Features to Avoid
J. H. Saltzer

Purpose

This section documents a number of restrictions on usage of the 645 which should be observed when writing programs to operate in the Multics environment. Some of these restrictions are enforced by the Multics supervisor; others, while not enforced, should be followed to minimize the effect of potential supervisor or hardware changes. In some cases, these restrictions are implied, without being explicitly stated, in miscellaneous other sections of the MSPM.

Use of sp-sb Base Pair

The sp-sb base pair is reserved for use as a stack pointer. Base register sb is locked; its value may only be changed by a call to the supervisor. (N.B.: call not currently provided.) This locking is assumed by interrupt handling routines. The convention that location sp|16 contains the next value to be loaded into sp on a call is assumed by the fault and interrupt handling procedures; failure to observe this convention can cause data in the stack to be overwritten unpredictably by fault or interrupt handlers. (See also BD.7.00.)

Base Register Pairing

The fields controlling pairing of the base registers can be loaded in slave mode. As a result, it is possible to write a program which unpairs or re-pairs the base registers. This ability should not be used, however, because it is anticipated that the ability to change base register control fields will be restricted to master mode procedures only. A supervisor entry may be called to change base register control field settings, if desired. (N.B.: call not currently provided.) The emphasis on retaining only standard base register pairing is inspired primarily by the desire to keep open the ability to add more pairs of base registers to the 645 processor, a change which would include permanent "wired-in" pairing of the present eight base registers. Also, if a slave mode program unintentionally re-pairs bases it is very difficult to trace back through the chaos resulting. Such chaos can follow a return macro using incorrect machine conditions. (See also BD.7.00.)

Parity Mode Bit

The 645 processor permits a slave mode procedure to set the parity mask bit, thereby inhibiting recognition of parity errors on access to core memory (and attempts to read a prototype clock incorrectly). In the interest of good programming practice, it would be unwise to set the parity mask bit, since trouble in accessing core memory should be reported, not ignored. In addition:

1. Whenever a fault or interrupt occurs, a supervisor procedure will force the setting of the parity mask bit to the off position, and will take no special care to restore a different setting upon completion of the fault or interrupt.

2. It is anticipated that the processor specification will be changed to prohibit setting the parity mask bit in slave mode.

Hard-to-Interrupt Instructions and Modifiers

The 645 processor has in its repertoire a number of instructions (e.g., RPD) and special modifiers (e.g., IT) with the property that an instruction, once begun:

1. May not be able to complete execution because of a missing page or pending interrupt, and

2. Cannot be scrapped and restarted from the beginning because a core location or register has already been modified.

Such instructions must be interruptable in mid-execution in a way that they can be continued at a later time; elaborate special-purpose hardware has been provided to snapshot the entire processor state including internal registers when an interrupt or fault occurs.

The cost of providing this interruptability is quite high in several directions:

1. The special purpose hardware is not needed for any other function.

2. Every interrupt and fault (not just those occuring during execution of a hard-to-interrupt instruction) must be started and ended with a pair of relatively long instructions requiring 11 microseconds to save the snapshot in core, and restore it to the processor, respectively. In addition, following every fault

or interrupt on which control was returned to the user, the stored machine conditions must be checked for validity by a procedure which performs about a dozen tests.

Since it is unlikely that these continually paid costs are paid back by the time saved in occasional use of these instructions, a hardware change to force a compatibility fault when they are used would allow removal of both of the above costs, and it would also permit addition of an interpreter procedure which simulates the effect desired but using more easily interruptible instructions. Thus it is inadvisable to utilize any of the instructions or modifiers in question, so as to make as simple as possible any future hardware change along this line.

The following instructions and modifiers are included in the above discussion:

XED Execute double

RPT Repeat

RPD Repeat double

RPL Repeat link

All indirect modifiers which change the indirect word, namely:

DI

AD

SD

ID

DIC

IDC

SC

SCR