SECTION 17

Error Handling

17.1 INTRODUCTION

This section explains the signalling of errors. It explains how the microcode signals an error condition to a program or to the user.

This section is organized from the abstract to the concrete. First, the error conditions are described. Then progressively lower levels of the implementation are discussed.

17.2 MICROCODE ERROR CONDITIONS

There are a number of conditions that are signalled by the system microcode. Listed below are the condition flavors, all of which are built directly or indirectly upon ERROR. A later section covers the names of the microcode traps themselves. All of the symbols listed are in the ERROR-HANDLER (EH:) package unless another package name is present.

ARG-TYPE-ERROR: an argument to an operation is not of an acceptable type.

ARRAY-NUMBER-DIMENSIONS-ERROR: attempt to access an array with greater or fewer dimensions than the array has. Built on BAD-ARRAY-ERROR.

BAD-ARRAY-ERROR: generic array problems that lack their own condition.

CALL-TRAP-ERROR: break on entry to a function or on normal exit from a CATCH.

CANT-INITIATE-ON-THIS-DEVICE-TYPE: tried to initiate I/O on an unknown device type.

CELL-CONTENTS-ERROR: the transporter found something bad (but not fatal) in memory. Unbound variables and undefined functions signal other conditions.

DANGEROUS-ERROR: virtual-memory overflow, region-table overflow.

SYS:DIVIDE-BY-ZERO: division by zero.

EXIT-TRAP-ERROR: break on exit from a function.

FLOATING-EXPONENT-OVERFLOW-ERROR: result is too large in magnitude to be represented as a floating point number.

FLOATING-EXPONENT-UNDERFLOW-ERROR: result is too small in magnitude to be represented as a floating point number.

FUNCTION-ENTRY-ERROR: a problem was encountered in entering a function, like too many or too few arguments, or an argument of a wrong data type.

INTERNAL-MEMORY-LOCATION-OOB: out-of-bounds reference to an internal processor memory.

INVALID-FUNCTION: some non-functional object was called as a function.

MAR-BREAK: the Memory Address Register comparator caused a break on a read or write.

PDL-OVERFLOW-ERROR: stack overflow on the regular or special variable PDL.

STEP-BREAK-ERROR: signalled by breakpoints, single-step breaks, and trace breaks.

SUBSCRIPT-ERROR: a subscript for an array access is out-of-bounds, negative, or otherwise in error.

THROW-EXIT-TRAP-ERROR: break on THROW through a marked catch.

THROW-TAG-NOT-FOUND: a THROW was done to a tag for which there is no pending CATCH.

UNBOUND-VARIABLE: an access to a symbol's value cell or to a closure variable found it unbound (DTP-NULL). Built on CELL-CONTENTS-ERROR.

UNDEFINED-FUNCTION: an access to a symbol's function cell or to a method of a select-method found it unbound (DTP-NULL). Built on CELL-CONTENTS-ERROR.

UNIMPLEMENTED-HARDWARE: tried some operation on XBUS or UNIBUS, neither of which exists in the Explorer System.

USER-NUBUS-ERROR: error in user NuBus operation.

SYS:ZERO-TO-NEGATIVE-POWER: attempt was made to raise zero to a negative power.

17.2.1 Microcode Error Table.

This paragraph is a description of microcode error table entries. The microcode error table is read from SYS:UBIN;EXP1-UCODE.TBL#nnn (where nnn is the microcode version) and stored as the value of the variable MICROCODE-ERROR-TABLE. The error table provides information about error trapping locations in the microcode. Each entry in the MICROCODE ERROR TABLE is a list which associates a microcode PC to a symbolic description of the error called an Error Table Entry (ETE). The CAR of the ETE list contains information about how to construct the appropriate condition instance. This information is fully defined in SYS:EH;ERROR-CONDITION-DEFINITIONS and is described below.

Each location in the microcode where TRAP can be called is followed by an ERROR-TABLE pseudo-op. This pseudo-op appears at the PC which is going to be TRAP's return address. An example is:

```
(ERROR-TABLE ARGTYP FIXNUM M-T 0)
```

The CDR of this list is the ETE. The ETE's first element is the name of the error and the second is the first "argument" to that error's associated routines. Note that all ETEs are lists whose CARs are symbols. The symbol is defined by a DEF-UCODE-ERROR form in SYS:EH;ERROR-CONDITION-DEFINITIONS. DEF-UCODE-ERROR tells what flavor of condition instance to build and what to put in it for every microcode trap, keyed by the name of the error.

Error-table entries are listed below alphabetically by type, with the args as a lambda-list. Types not used in the error-handler are marked *. These types seem to be historical remnants. Some of these types are used in CADR or Lambda microcode; they aren't used on the Explorer.

In the argument lists below, an argument of the form "FOO-location" will have the value of one of the following registers when the trap occurs: M-A, M-B, M-C, M-D, M-E, M-F, M-G, M-H, M-I, M-J, M-K, M-L, M-Q, M-R, M-S, M-T, M-1, M-2, M-3, M-4, A-SG-PREVIOUS-STACK-GROUP, PP (the top of the PDL), VMA, MD, or (PP number) where number is a negative index into the PDL. These locations are saved in stack groups and are accessed by EH:SG-CONTENTS for the purposes of examination, replacement, and restarting from errors.

Since these registers are the ones saved in stack groups, they will be pushed on the PDL, written to memory, GC-WRITE-TESTed, and so forth. Therefore, they must be fully tagged or have all ones or zeros in the data type field (M-1, M-2, M-3 and M-4 are exempt from this restriction).

An argument of the form "FOO-tag" is a pseudo-label defined by a RESTART entry. See section on restart below.

17.2.1.1 Special Error Table Entries.

The error table entries in this section are those not directly related to trap messages. They provide useful ancillary information for the processing of traps.

* ARG-POPPED arg1 arg2 arg3 arg4.... Saves (arg1 arg2...) as info on where to find popped args if trap after pop. The error handler collects these but never uses them.

CALLS-SUB subroutine-tag. The subroutine-tag will appear on the >>TRAP line after "->", as an indication of what function called the trapping routine.

* DEFAULT-ARG-LOCATIONS arg1 arg2 arg3.... Saves (arg1 arg2...) as info on where args may be found if no other info is available. The error handler collects these but never uses them.

RESTART restart-tag. Defines restart-tag to be this micro-pc, as a pseudo-lable for other error table entries. A restart-tag marks the micro-pc at which to resume execution, for proceedable traps. Note: RESTART is the only way to define restart-tags. It is not important whether or not the restart-tag is defined as a label in the microcode (they tend to be similar, just for mnemonic value), but it must be unique among restart-tags.

17.2.1.2 Normal Error Table Entries.

The error table entries in this section are used to signal the actual traps. Their names are used in DEF-UCODE-ERRORs to generate condition instances.

AREA-OVERFLOW area-number-location. Signalled during region consing when the area has a maximum size "Allocation in the ~A area exceeded the maximum of ~D. words."

ARGTYP description arg-location &optional arg-number restart-tag function-name. Description is what was expected - see EH:DATA-TYPE-NAMES. It can be a list of allowable types.

Arg-location contains the failing arg. M-1, M-2, M-3 and M-4 are currently not allowed, because the error handler wants a locative to the slot in the erring stack group, and the high bits of M-1, M-2, M-3 and M-4 are stored separately from the pointer field (thats how all 32 bits are preserved). VMA is also not allowed.

Arg-number is the bad argument's number, zero origin.

Restart-tag is the label to restart from, if other than the current pc.

Function-name is the erring function, if that is not the obvious one.

ARRAY-HAS-NO-LEADER array-location restart-tag. Some array leader operation was attempted on an array with no leader.

ARRAY-NUMBER-DIMENSIONS ignore number-of-dimensions array-location restart-tag. Number-of-dimensions is a constant (array-decode-*), or nil if variable (most cases).

Array-location is where to find the array.

Restart-tag is the label to restart from, if other than the current pc.

BAD-ARRAY-DIMENSION-NUMBER array-location dimension-number-location. "The dimension number ~S is out of range for ~S."

BAD-ARRAY-TYPE array-header-location. The array type of this array was not one of the legal types.

BAD-CDR-CODE address-location. "A bad cdr-code was found in memory (at address ~O)".

BAD-INTERNAL-MEMORY-SELECTOR-ARG object-location. "~S is not valid as the first argument to %WRITE-INTERNAL-PROCESSOR-MEMORIES." Currently, only 1, 2, 4, and 5 are legal.

BIGNUM-NOT-BIG-ENOUGH-DPB. "There is an internal error in bignums; please report this bug."

BITBLT-ARRAY-FRACTIONAL-WORD-WIDTH. "An array passed to BITBLT has an invalid width. The width, times the number of bits per pixel, must be a multiple of 32."

BITBLT-DESTINATION-TOO-SMALL. "The destination of a BITBLT was too small."

BREAKPOINT. Caused by executing a BPT instruction. This is the misc entry smashed in for breakpoints in FEFs.

CALL-TRAP. Microcode support for things like BREAKON. This is the entry half.

CANT-INITIATE-ON-THIS-DEVICE-TYPE device-type-location. "Can't initiate on device type ~S."

CONS-ZERO-SIZE location. Attempt to allocate zero storage.

DATA-TYPE-SCREWUP name. This happens when some internal data structure contains the wrong data type. Name is the type of structure.

DIVIDE-BY-ZERO &optional dividend-location.

EXIT-TRAP. Microcode support for things like breakon. This is the exit half.

FILL-POINTER-NOT-FIXNUM array-location restart-tag "The fill-pointer of the array given to ~S, ~S, is not a fixnum."

FLOATING-EXPONENT-OVERFLOW arg "~S produced a result too large in magnitude to be a :[;small] flonum." "Result is to be placed in M-T and pushed on the pdl. Arg is SFL or FLO. In the case of SFL the PDL has already been pushed."

FLOATING-EXPONENT-UNDERFLOW arg. "~S produced a result too small in magnitude to be a :[;small] flonum." "Arg is SFL or FLO."

FLONUM-NO-GOOD. A subset of argtyp. ARGTYP is not usable.

TOO-FEW-ARGS "Function ~S called with only ~D argument ~1@*~P."

TOO-MANY-ARGS "Function ~S called with too many arguments (~D)."

ILLEGAL-AREA "Tried to cons in free, fixed, or unused-code region. Please report this error."

ILLEGAL-INSTRUCTION. Illegal macroinstructions that aren't unimplemented miscops. "There was an attempt to execute an invalid instruction: ~O."

INDIVIDUAL-SUBSCRIPT-OOB array-location dimension-number-location restart-tag. Dimension-number is the location of the offending dimension's index. "We assume that the current frame's args are the array and the subscripts, and find the actual losing subscript that way."

INSTANCE-LACKS-INSTANCE-VARIABLE var-location instance-location. "Signalled by LOCATE-IN-INSTANCE." "There is no instance variable ~S in ~S."

INTERNAL-MEMORY-LOCATION-OOB memory-selector-location index-location. "Internal memory location is out of range."

MAR-BREAK direction. "The MAR has gone off because of an attempt to write ~S into offset ~O in ~S." "The MAR has gone off because of an attempt to read from offset ~O in ~S." Direction is WRITE or READ. This trap is for the MAR feature.

MICRO-CODE-ENTRY-OUT-OF-RANGE misc-number-location. "MISC-instruction ~S is not an implemented instruction."

NO-CURRENTLY-PREPARED-SHEET location. "There was an attempt to draw on the sheet ~S without preparing it first."

NO-MAPPING-TABLE. "Flavor ~S is not a component of SELF's flavor, ~S, on a call to a function which assumes SELF is a ~S."

NO-MAPPING-TABLE-1. "SYS:SELF-MAPPING-TABLE is NIL in a combined method."

NONEXISTENT-INSTANCE-VARIABLE. "Compiled code referred to instance variable ~S, no longer present in flavor ~S."

NUMBER-ARRAY-NOT-ALLOWED array-location restart-tag. "The array ~S, which was given to ~S, is not allowed to be a number array." This one occurs when making a locative to an array element. None of the current uses has a restart-tag.

NUMBER-CALLED-AS-FUNCTION number-location. "The number, ~S, was called as a function."

PDL-OVERFLOW pdl-type. "The ~A push-down list has overflowed." Pdl-type is either REGULAR or SPECIAL.

RASTER-WIDTH-TOO-WIDE. The raster width of a font was too wide.

RCONS-FIXED. "There was an attempt to allocate storage in the fixed area ~S." The area number is in M-S.

REGION-TABLE-OVERFLOW. "Unable to create a new region because the region tables are full."

RPLACD-WRONG-REPRESENTATION-TYPE first-arg-location. "Attempt to RPLACD a list which is embedded in a structure and therefore cannot be RPLACD'ed. The list is ~S." First-arg-location tells where to find the first arg to RPLACD.

SELF-NOT-INSTANCE. "A method is referring to an instance variable, but SELF is ~S, not an instance."

SG-RETURN-UNSAFE. "An unsafe stack group attempted to STACK-GROUP-RETURN." "No args, since the frob is in the previous-stack-group of the current one."

STACK-FRAME-TOO-LARGE. "Attempt to make a stack frame larger than 256. words." Called from %ASSURE-PDL-ROOM.

STEP-BREAK. Interface to microcode support for single-stepping.

SUBSCRIPT-OOB index-location limit-location restart-tag indices-flag. Index-location is where to find the index used, and should always be present.

Limit-location is where to find the legal limit for the subscript, and should always be present.

Restart-tag may be a list, which will be pushed sequentially. "This is used to get the effect of making the microcode restart by calling a subroutine which will return to the point of the error."

Indices-flag is "T if indices are on the stack, 1 if ar-1-force (etc.) and there is only one index, or absent if array's rank should be used to decide where the args are."

*THROW-TAG-NOT-FOUND.

TRANS-TRAP For the conditions unbound-symbol, unbound-instance-variable, unbound-closure-variable, undefined-function, bad-data-type-in-memory.

"The variable ~S ~A unbound."

"The function ~S ~A undefined."

"The instance variable ~S ~A unbound in ~S."

"The variable ~S ~A unbound (in a closure value-cell)."

"The word #<~S ~S> was read from location ~O ~O[in~A~]."

USER-NUBUS-ERROR high-address-location low-address-location nubus-tms-type-location "User NuBus Error of type ~S, at address #x ~16R 16,6, 48~R. %Error Bits: #x ~16R."

USER-NUBUS-GACBL-LIMIT. "Number of GACBLs exceeded limit in user NuBus operation."

USER-NUBUS-PARITY-LIMIT. "Number of Parity Errors exceeded limit in user NuBus operation."

VIRTUAL-MEMORY-OVERFLOW. "You've used up all available virtual memory!"

WRITE-IN-READ-ONLY address-location. "There was an attempt to write into ~S, which is a read-only address."

WRONG-SG-STATE sg-location. "The state of the stack group, ~S, given to ~S, was invalid." Sg-location is where to find the invalid stack group.

ZERO-ARGS-TO-SELECT-METHOD select-method-location. "~S was applied to no arguments."