BACKGROUND INFORMATION AND IMPORTANT VARIABLES

IX1 always points at loc 0 of 28 word work area

IX2 always points at the top of variable stack, where next value goes

IX3 is used for various purposes

```
Work area (A) (28 words pointed to by IX1)
    -6  address of top of WORD stack * 2
    -5  next spot in variable part of stack *2
    -4  address of save routine (SAVE0 or SAVE)
    -3  address current character to examine in disk buffer, *2
    -2  address of ACCEP routine or RETRY
    -1  address of BCD character table end (BCDBL)
    0   current character EBCDIC then FORTH
    1   zero - think this is unused
    2   address of word * 2 inside WORD stack
    3   first word  (WORD)
    4   second word (WORD+1)
    5-22 rest of 20 word long WORD buffer
```

E points to next open location in fixed dictionary

E1 points to highest active entry in dictionary

IC points to highest active entry in variable dictionary

INTST is the return stack

R points at the current return entry in return stack INTST

WORD is a 20 word long buffer for blank delimited strings

STACK is a 16 word deep stack for operands while executing

The dictionary consists of two parts. The fixed size portion begins at the end of the program, starting with the predefined operations such as LOC, ENTRY, INTEGER etc and growing to higher addresses in memory.

The variable size portion begins at the address set by the startup routine (e.g. 19FF) as the last used entry, and it grows to higher addresses as the data part of the entry is stored. Thus first variable data is at 1A00

As an example, the first card read from disk defines DEP to be LOC followed by DEPOSIT. The card image is .DEP LOC DEPOSIT, with the dot or colon triggering a definition. The first part of the dictionary has the words

DE and Pblank, a third word pointing to
the INTER routine to cause this entry to
be interpreted if the word DEP is used later,
and a fourth word that points to the
second part of the dictionary (IC) where
the words LO, Cblank, DE, PO, SI, Tcomma,
and blankblank are stored before IC is dumped

EXPLANATIONS OF OPERATION AND OTHER COMMENTS

How a definition works:
WHen a card image has a colon or '.' character, the next blank delimited string is added to the fixed dictionary with an address of the INTER (Interpreter) routine and the remainder of the definition stored as text in the variable section of the dictionary. This includes the terminating semicolon or ',' which are not executed in any way until the word that was defined is subsequently interpreted.

Example - the card has ".DEP LOC DEPOSIT," ignoring the quotes. The first period starts a definition, reading the string DEP and entering it into the dictionary. The dictionary entry has the two words DE and Pblank, the address of the interpreter routine, and a pointer to the point in the variable dictionary where the string "LOC DEPOSIT," is stored.

How an OPERATION works:
The variable dictionary entry for this word is branched to by the 1130 and executed as machine language instructions. This is different from colon definitions which branch to the INTER routine when they are encountered.