

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A. I. LABORATORY

HARDWARE MEMO 12
DK-10 PROGRAMMING
Donald Eastlake

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

DK-10 Programming

1. Introduction

The Systems Concepts DK-10 terminal controller provides direct from memory output and buffered input of characters at program set speeds from a DEC PIP-10 system. The following memo describes the programming of the DK-10 and some aspects of its operation.

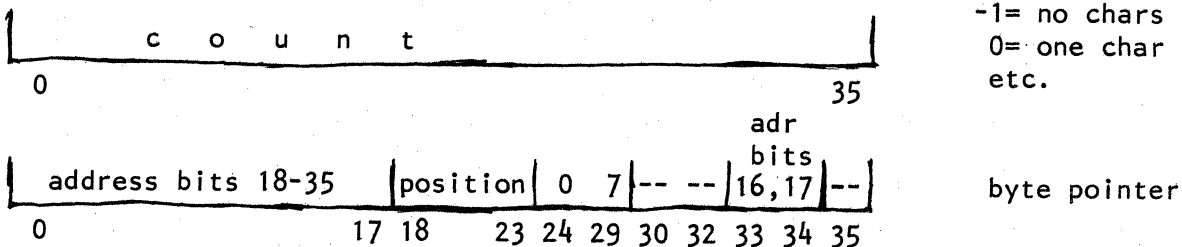
2.1 Input

The DK-10 has provisions for setting the speed of each input line and for enabling and disabling each input line (see section 3.4). When a character is received on an enabled input line it is placed in a 16 slot first-in-first-out stack along with the line number on which it was received. A character received when this stack is full will be lost.

As long as there is at least one character in the input stack, the input done flag will be on in the DK-10's control register and it will request an interrupt if its FIA is non-zero. Character and line number pairs may be read in from the stack by doing a DATAI. The entire input stack can be cleared by doing a CONC to clear the input done flag.

2.2 Output

The DK-10 performs character output directly from memory and handles enough bookkeeping that, for many applications, no output interrupts need be taken. The DK-10 has a base address register (see section 3.2) that is used to access a list of word pairs for the output lines. The first word of each pair, at the base address plus twice the line number, has a count of characters remaining to be output on that line. The second word, at the base address plus one plus twice the line number, is a pointer to the buffer of characters to be output. This pointer is similar to a regular byte pointer with its halves swapped and is effectively ILDBed by the DK-10 to get characters. The size field is always treated and stored back as if it were 7. The position field should be of the form 36-7k.



When the count word is counted down to -1 for an output line, the output done flag is set for this line. If output interrupts have been enabled, the DK-10 will then request an interrupt and it will be possible to read in the

number of the line that stopped. The DK-10 has provisions for starting and stopping each output line and for setting the speed of each output line by doing CONOs.

To make it easy to ring buffer output, the DK-10 actually examines each word that it fetches from memory for the purpose of extracting a character from it. If the bottom bit is one (this bit is not used in packed 7 bit fields) then the word is treated as a new "swapped byte pointer" which is effectively ILDBed to get the character to output and then stored back on top of the second word of the word pair associated with this output line. Thus one can store a byte pointer back to the start of a buffer at the end of the buffer.

If the DK-10 gets a non-existent memory indication on a read or write or a parity error on a read it sets an error flag in its control register and stops all output processing until reset.

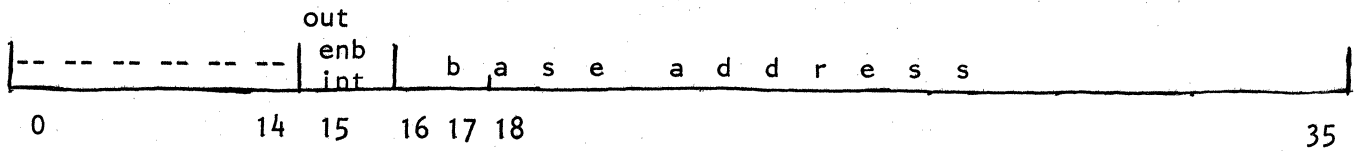
3.1 DATAI



14-17 input line on which character was received

28-35 character code

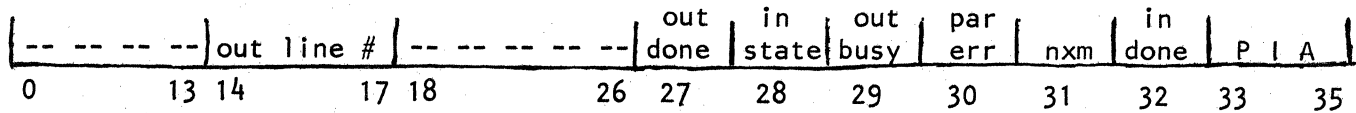
3.2 DATA0



15 if a one, enables output interrupts

16-35 base address for output control words

3.3 CCNI



14-17 number of an output line with done flag set

27 in on, indicates some output line done

28 state of selected input line

29 selected output line busy

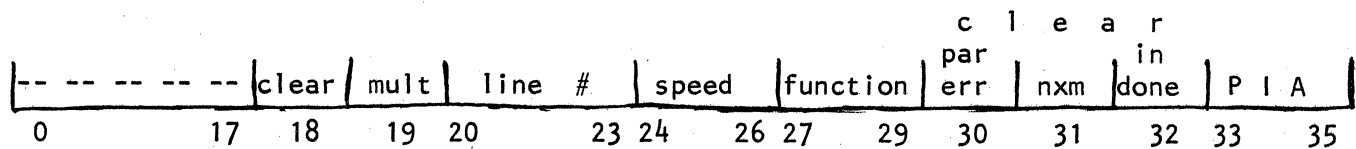
30 parity error

31 non-existent memory error

32 input done flag

33-35 priority interrupt channel assignment

3.4 CCNO



- 18 master clear
- 19 makes function apply to selected line through highest
- 20-23 line number
- 24-26 speed if used by function
- 27-29 function as follows:
 - 0 set output done
 - 1 start output
 - 2 disable input
 - 3 enable input
 - 4 stop output & clear output done
 - 5 set output speed & start output
 - 6 set input speed & disable input
 - 7 set input speed & enable input
- 30 clear parity error
- 31 clear non-existent memory error
- 32 clear input done flag
- 33-35 priority interrupt channel assignment

