A KIM-1 DISASSEMBLER

Received: 77 Oct 28

I have just got my KIM1 up and running with a FORETHOUGHT motherboard kit, and an 8K ECONORAM kit from GODBOUT. I was impressed with the quality of both kits. The ECONORAM kit was a little too crowded. I had to unsolder some of the despiking capacitors that were too fat to fit between the 2102s. Next time, I will put the thin ones in these locations.

My first job was to get "A 6502 DISASSEMBLER FROM APPLE", (*DDJ* VOL 1 no. 8) to work. I had more than a little trouble, till I discovered that the KIM1 Outch and Prtbyt routines were clobbering the Y register that the disassembler was using as a counter. You might like to publish my version of this program which is enclosed. If anyone wants to avoid the problem of hand keying it in, I would be glad to furnish a copy on paper tape for $1.00 or on cassette for $3.00.

I am mainly interested in engineering applications. I would think that the micro computer is so cheap now that it would displace many applications now on time-sharing. Of course, the answer is in the cost of programming. That is where we computer buffs could perform a real service and have a good time doing it. Time sharing is good but the overhead is high. There is nothing like having your own computer right there when you want it.

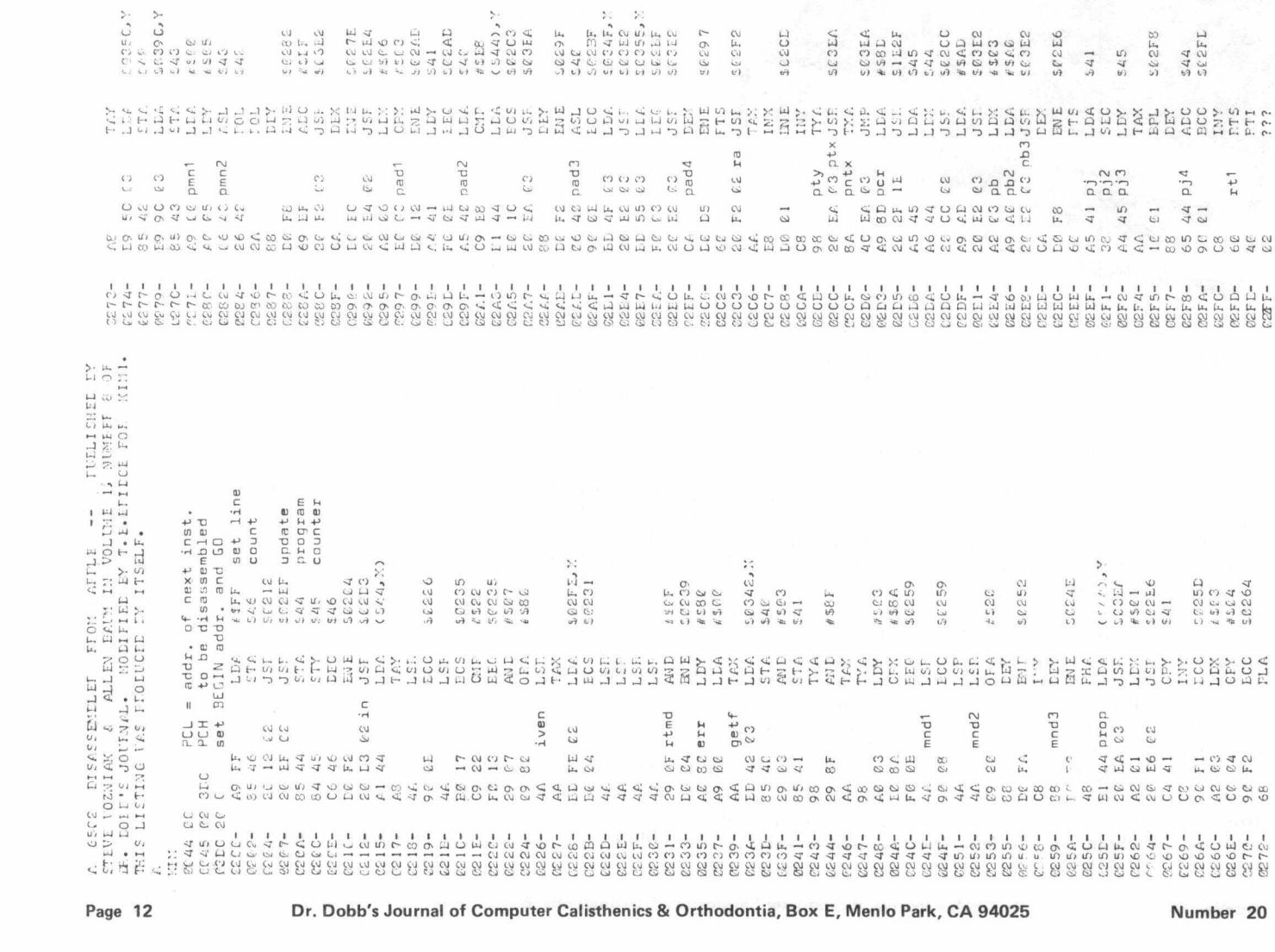
I would like to use BASIC, but I fear it will be too slow for some of my problems. For example, a single problem in PIPE STRESS may need to invert a 6 X 7 matrix 20 times. If we could only debug with a small problem on BASIC, and then compile the source tape into machine language for running large problems. The construction of such a compiler is far beyond my capability. Does anyone market a BASIC with floating point, and arrays that will run on my KIM1?

To work on any big engineering program, you need at least two people. One must be knowledgeable with the problem. You need feedback in working up the documentation, and in other ways too. I am 70, retired and available. I have a degree and 40 years experience in mechanical engineering. I can run a typewriter. Let me know if I can help you with a problem. I submitted the following programs to the DECUS library: PIPE STRESS, CRITICAL PATH METHOD, REGRESSION ANALYSIS, BUILDING HEAT LOSS. Having only 4 K of core on my PDP8f that I recently sold, BASIC was out of the question. Having no feedback from anybody, my documentation was bad, in spite of very great effort.

Very truly yours,
Theodore E. Bridge

54 Williamsburg Dr.
Springfield, MA 01108

PS In writing the above mentioned programs for the DECUS library, I made good use of a FLOATING POINT INTERPRETER in the DECUS library. It uses much less memory than BASIC and runs very much faster. Would anyone like to help me write a similar interpreter for my KIM1? I worked up a trace routine for the interpreter. I grant that debugging with BASIC would be much easier. Also only a few people can communicate with you if you are using your own interpreter. However, a BASIC interpreter is not really useable on a very large engineering application.



KICKING THE 8080 HABIT

Received: 77 Oct 17

Maybe it's just my imagination, but it seems that a lot of people aren't utilizing the Z-80 to its fullest. Everyone is so used to writing code for the 8080 that they don't seem to bother upgrading their software when they upgrade their CPU. Or maybe, as in my case, since we've been eating and sleeping in Intel Octal for so long that we just can't seem to catch on to all the enhancements of the added Z-80 functions and instructions. I would like to see you guys (or, someone else) come out and explain all the nifty Z-80 tricks. I know I can't be the only one that is stuck in the rut of 8080 code. (Please!! Don't tell me I swapped my CPU board JUST for speed—the software potential is fantastic.)

Michael Settle
751 Washington #115
Arlington, TX 76011

How about it? Anyone care to send us something on "Taking full advantage of Z-80's inner mysteries"?—TRW

MICROCOMPUTER SOFTWARE INDEX

News Release *Received 77 Nov 28*

A great quantity of software has been published in microcomputer books and magazines since late 1975. These programs range from complete BASICs to assemblers, routines, games and direct application programs.

The Schreier Software Index, An Index to Published Microcomputer Software, indexes hundreds of published microcomputer programs. The S S I contains cross references and over 130 program divisions. Many of the cited programs specify chip description. In addition, the S S I features complete publication documentation and bibliographic data. Locating a random number routine for a 6800 or an 8080, for example, takes but seconds.

Direct orders may be placed with S S I, 4327 East Grove Street, Phoenix, Arizona 85040. Price postpaid in the United States, Mexico and Canada is $5.00. The S S I is also available from a number of microcomputer dealers across the U.S.

```
THESE CHANGES WERE ADDED TO SAVE
THE  Y  REGISTER WHEN CALLING
KIM ROUTINES  OUTCH & PRTBYT

KIM
03DC 20 G
03DC-    20 00 02    JSR   $0200  begin
03DF-    4C 4F 1C    JMP   $1C4F  start
03E2-    84 EE       STY   $EE    savey
03E4-    20 A0 1E    JSR   $1EA0  outch
03E7-    A4 EE       LDY   $EE    gety
03E9-    60          RTS
03EA-    84 EE       STY   $EE    savey
03EC-    20 3B 1E    JSR   $1E3B  prtbyt
03EF-    A4 EE       LDY   $EE    gety
03F1-    60          RTS

THESE ARE THE TABLES

KIM
03DC 20 44
0044 F2 DC.
0045 03 3DC
03DC 20 17F7
17F7 FF DC.     end of tables
17F8 FF 03.
17F9 DA 0300    start of tables
0300 45 0
;1803004503D00840093022453 3D0084009400245 33D008400940000 58A
;18031840E0D004000002244 33D08C4400112244 33D08C449A10220782
;1803304433D00840091 0224433D00840096213 78A90021818200000667
;180348594D91928 64AA59DACA9ACA3A8A4D900D8A4A4001C8A1C230C0C
;1803605D8E1EA19D8A1D239D8E1DA1002919AE69A8192324531E23085E
;180378245319A1001A5E5EA5692424AEAEA8AD29007C00159C6D00085E
;1803906A569295384133411A56923A0D8625A48266294885444C8540A16
;1803A86844E89400E4088474E4286E74F4CC4A72F2A48A00AAA2A20CE7
;1803C07474747244 68E232E2002200 1A1A2600727288C8C4CA26480997
;1803D84444A2C8200024C4F1C84EE20AC1EA4EE6084EF203E1EA40A8F
;00000A000A
```

VDM-1 DRIVER MODS

by John Moorhead

Received: 77 Dec 2

```
*  THIS IS THE VDM-1 DRIVER ORIGINATED
*  BY LI CHEN WANG (DR. DOBB'S JOURNAL,
*  VOLUME #1, ISSUE #6).
*
*  MODIFIED TO INCLUDE SPEED CONTROL AND
*  GENERALLY EMULATE THE SOFTWARE DRIVER FOR
*  PROCESSOR TECHNOLOGY'S VDM-1 VIDEO DISPLAY.
*  THIS PROGRAM WILL ALSO ACT AS THE DRIVER
*  FOR SOLID STATE MUSIC'S VB-1 DISPLAY.
*
*  WRITTEN BY JOHN MOORHEAD, 928 J STREET,
*  DAVIS, CA.  95616  TEL NO 916-758-2495
*
*  MAY 5, 1977
*
*  BACKSPACE = CONTROL H OR B.S. KEY
*  SCREEN CLEAR = "CANCEL" = CTRL-X
*  AUTO CRLF ON INPUT OF A CARRIAGE RETURN
*  CHANGE SPEED BY TYPING NUMBER DURING OUTPUT
*  STOP DISPLAY BY TYPING SPACE BAR
*  START DISPLAY BY TYPING ANY CHARACTER
*
*  NOTE - A SCREEN CLEAR MUST BE THE FIRST
*  CHARACTER SENT IN ORDER TO INITIALIZE THE
*  CURSOR AT THE BOTTOM LEFT CORNER OF THE
*  SCREEN AND ENSURE PROPER SCROLLING.
*
*
*  ENTER WITH CHARACTER IN THE ACCUMULATOR
*
*
STRT   PUSH  H        SAVE SYSTEM REGISTERS
       PUSH  D
       PUSH  B
       PUSH  PSW      CHAR IS IN THE ACCUM
       LHLD  VDMP     GET SCREEN POSITION POINTER
       CPI   0DH      IS THE CHAR A CARRIAGE RET?
       JZ    CR       YES - SCROLL, OUTPUT A CR
       CPI   08H      IS IT A BACKSPACE?
       JZ    BS       YES - MOVE CURSOR BACK
       CPI   18H      IS IT A CTRL - X (SCREEN CLEAR)?
       JZ    CLEAR    YES - ERASE ENTIRE SCREEN
       CPI   7FH      IS IT A RUBOUT CHARACTER?
       JZ    SPEED    DON'T DISPLAY (FOR ALS-8 ONLY)
       CPI   20H      DON'T DISPLAY CONTROL CHARS
       JC    SPEED    EXIT TO CHANGE SPEED
       MOV   M,A      IT HAS TO BE DATA
       INX   H        UPDATE CHAR POSITION ON SCREEN
       MVI   M,0A0H   PUT CURSOR ON SCREEN
       JMP   LINOV    TEST FOR LINE OVERFLOW
BS     MVI   M,20H    REMOVE CURSOR
       DCX   H        BACK UP POINTER
       JMP   BS-5
CR     MVI   M,20H    CHAR IS A CARRIAGE RETURN
       MOV   A,L      UPDATE NEXT CHAR POSITION
       ANI   0C0H
       ADI   40H      SETTING UP FOR NEW LINE
       MOV   L,A      ADDRESS OF NEW LINE
       MVI   A,0
       ADC   H        ADD WITH CARRY
       MOV   H,A
LINOV  SHLD  VDMP     SAVE POINTER FOR NEXT CHAR
       MVI   A,7FH
       ANA   L
       JNZ   SPEED    EXIT
       MVI   M,20H
       LXI   H,0CFC0H SET UP DATA LINE (LINE 16)
       SHLD  VDMP     SAVE POINTER
       LXI   H,0CC40H 15 LINES OF SCREEN DATA
       LXI   D,0CC00H TOP OF SCREEN.  SET UP
       LXI   B,03C0H  TO SCROLL 15 LINES
SCROL  MOV   A,M      START SCROLLING UP
```