A KIM-1 DISASSEMBLER

Received: 77 Oct 28

I have just got my KIM1 up and running with a FORE-THOUGHT 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 despikeying 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", (BOF VOL 1 no. 8) to work. I had more than a little trouble, till I discovered that the KIM1 Outch and Prbyt 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 50 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.

Page 12  Dr. Dobbs Journal of Computer Calisthenics & Orthodontia, Box E, Menlo Park, CA 94025  Number 20

426
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.

(Yes! Don't tell me I swapped my CPU board just for speed—the software potential is fantastic.)

Michael Settle
751 Washington #115
Arlington, TX 76101

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 can 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 SSI contains cross references and over 130 program divisions. Many of the cited programs specify chip description. In addition, the SSI 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 SSI, 4327 East Grove Street, Phoenix, Arizona 85040. Price paidpost in the United States, Mexico and Canada is $5.00. The SSI is also available from a number of microcomputer dealers across the U.S.

VDM-1 DRIVER MODS

by John Moorhead

0005 * THIS IS THE VDM-1 DRIVER ORIGINATED
0010 * BY LI CHEN WANG (DR. DOBB'S JOURNAL,
0015 * VOLUME #1, ISSUE #6).
0020 * 
0025 * MODIFIED TO INCLUDE SPEED CONTROL AND
0030 * GENERALLY EMULATE THE SOFTWARE DRIVER FOR
0035 * PROCESSOR TECHNOLOGY'S VDM-1 VIDEO DISPLAY.
0040 * THIS PROGRAM WILL ALSO ACT AS THE DRIVER
0045 * FOR SOLID STATE MUSIC'S VS-1 DISPLAY.
0050 * 
0055 * WRITTEN BY JOHN MOORHEAD, 928 J STREET,
0060 * DAVIS, CA. 95616 TEL NO 916-759-2495
0065 * 
0070 * MAY 5, 1977
0075 * 
0080 * BACKSPACE = CONTROL H OR B.S. KEY
0085 * SCREEN CLEAR = "CANCEL" = CTRL-X
0090 * AUTO CRSP ON INPUT OF A CARRIAGE RETURN
0095 * CHANGE SPEED BY TYPING NUMBER DURING OUTPUT
0100 * STOP DISPLAY BY TYPING SPACE BAR
0105 * START DISPLAY BY TYPING ANY CHARACTER
0110 * 
0115 * NOTE - A SCREEN CLEAR MUST BE THE FIRST
0120 * CHARACTER SENT IN ORDER TO INITIIZE THE
0125 * CURSOR AT THE BOTTOM LEFT CORNER OF THE
0130 * SCREEN AND ENSURE PROPER SCROLLING.
0135 * 
0140 * ENTER WITH CHARACTER IN THE ACCUMULATOR
0145 * 
0150 * 
0155 * 
0160 STS T PUSH H SAVE SYSTEM REGISTERS
0165 STS T PUSH B
0170 STS T PUSH PSW CHAR IS IN THE ACCUM
0175 STS T PUSH Clh SPD GET SCREEN POSITION POINTER
0180 STS T CPI 0DH IS THE CHAR A CARRIAGE RET?
0185 STS T CPI J2 CR YES = SCROLL, OUTPUT A CR
0190 STS T CPI 0BH IS IT A BACKSPACE?
0195 STS T CPI J2 BS YES = MOVE CURSOR BACK.
0200 STS T CPI 18H IS IT A CTRL - X (SCREEN CLEAR)?
0205 STS T CPI J2 CLEAR YES = ERASE ENTIRE SCREEN
0210 STS T CPI 7FH: IS IT A ROUBOUT CHARACTER?
0215 STS T CPI J2 SPEED DON'T DISPLAY CONTROL CHAR.
0220 STS T CPI 20H SPEED EXIT TO CHANGE SPEED.
0225 STS T CPI P H, A SAVE TO DATA
0230 STS T MOV M, A IT HAS TO BE DATA
0235 STS T INX H UPDATE CHAR POSITION ON SCREEN
0240 STS T MOV M, 0AH PUT CURSOR ON SCREEN
0245 STS T JMP LINOV TEST FOR LINE OVERFLOW
0250 STS T H, PAH, 20H REMOVE CURSOR
0255 STS T MOV X, H BACK UP POINTER
0260 STS T MOV BS-5
0265 STS T MOV M, 20H CHAR IS A CARRIAGE RETURN
0270 STS T MOV A, L, A UPDATE NEXT CHAR POSITION
0275 STS T ANI OCOH
0280 STS T MOV L, A ADDRESS OF NEW LINE
0285 STS T MOV A, 0
0290 STS T ADD H ADD WITH CARRY
0295 STS T MOV M, H, A
0300 STS T MOV M, 20H CHAR IS A CARRIAGE RETURN
0305 STS T MOV L, A UPDATE NEXT CHAR POSITION
0310 STS T MOV M, 0AH PUT CURSOR ON SCREEN
0315 STS T MOV L, 7F EXIT SPEED
0320 STS T MOV M, 20H SPEED EXIT
0325 STS T MOV M, 0AH PUT CURSOR ON SCREEN
0330 STS T MOV M, 20H SPEED EXIT
0335 STS T LINOV CMD 1, 0, 00, CH, 0, 0 SET UP DATA LINE (LINE 16)
0340 STS T SHLD VDM SAVE POINTER FOR NEXT CHAR
0345 STS T SHLD VDM SAVE POINTER FOR NEXT CHAR
0350 STS T SHLD VDM SAVE POINTER FOR NEXT CHAR
0355 STS T SHLD VDM SAVE POINTER FOR NEXT CHAR
0360 STS T MOV A, M START SCROLLING UP

Number 20

Dr. Dobb's Journal of Computer Calisthenics & Orthodontia, Box E, Menlo Park, CA 94025

Page 13

427