

Modification and Relocation of FOCAL 65-E Into Erasible Prom

William C. Clements, Jr.
Dept. of Chemical & Metallurgical
Engineering
The University of Alabama
P. O. Box 2662
University, Alabama 35486

After using FOCAL for awhile, I became interested in storing the machine code in EPROM. Not only would this eliminate much of the waiting for tapes to load, but more important, it would free over 5K of user RAM for other purposes such as storing more FOCAL source code and variables, or for graphics routines.

The relocation of FOCAL and execution of it from PROM is not as straightforward as for some other programs, because the machine code is self-modifying in several places. Also, there are multitudes of data bytes used for address pointers scattered through the program, and these are in such a form that the ordinary kind of relocation routine would not convert them. Thanks to the excellent documentation supplied with FOCAL, I was successful in relocating it in a "clean", non self-modifying form. The code, together with an initialization routine that sets up page zero and other RAM locations used for user statements and to make the code "clean", fits neatly into three 2716's with plenty of room left over for other modifications such as tape load and save, "user" function, etc., which I have added to my version of FOCAL as well.¹ The modifications which follow are concerned with cleaning the code up for storage in PROM, and pertain to FOCAL 65-E for the KIM-1, obtained from the 6502 Program Exchange in Reno, Nevada.

The first order of business in preparing FOCAL for PROM is to get rid of the self-modifying parts. The three places I found where FOCAL modifies itself in the main code are at locs. \$2348-2353, \$282C-283D, and \$3408-3414. A fourth place occurs in page zero, where it doesn't matter since page zero is always RAM in 6502

systems. The other places are easily fixed. I borrowed a few locations from an obscure corner of KIM's on-board RAM to do it; neither KIM nor FOCAL seems to mind. The changes are as follows:

```
2348 was 8C 52 23 change to 8C DE 17 STY DJADR
234E was 8C 53 23 change to 8C DF 17 STY DJADR + 1
2351 was 4C 00 00 change to 4C DD 17 JMP $17DD
282C was 8C 3C 28 change to 8C DB 17 STY DJADR1
2835 was 8D 3D 28 change to 8D DC 17 STA DJADR1 + 1
283B was 6C 00 00 change to 4C DA 17 JMP $17DA
3408 was 8E 12 34 change to 8E E1 17 STX MOV11
340C was 8C 14 34 change to 8C E3 17 STY MOV22 + 1
3411 was B5 00 00 change to 4C E0 17 JMP MOVIT
3413 was 95 00 00 change to EA NOP
```

Additional code needed in page 17 is:

```
17DA 6C 00 00 JMP (0000)
17DD 4C 00 00 JMP 0000
17E0 B5 00 MOVIT LDA(X) 00
17E2 95 00 STA(X) 00
17E4 4C 15 34 JMP 3415
```

The address overwriting now occurs in page 17 RAM instead of in the main code, which can now be safely put into PROM.

Before doing so, however, we must relocate it. Note that relocation should not alter existing page boundaries (see warning on p. 44 of FOCAL 65-E Manual). This actually makes the job easier, because only the high-order bytes of addresses and address-words can be changed. Relocation then is accomplished by (a) adding the desired offset to the third byte of all three-byte instructions which do not reference page zero; (b) Offsetting the data words for routines such as PUSHJ and POPJ, the software stack manipulators. These words are scattered here and there through all the code. A listing of their high-order halves is given in Table 1; they are address words, so only the second byte is to be offset. (c) Offsetting the high-order bytes of the address tables at the end of the FOCAL code, which are at hex locs. 34FA-3515, 3546-3557, 356A-356E, 3598-359C, 35A2-35A6, 35AC-35B0, 35B6-35BA, 35C0-35C4, and 35CA-35CE. (d) Adding the offset to the IRQ-vector initialization byte at loc. 34AE (I date your cleverest relocation program to find *that* one!).

A final change necessary to execute FOCAL from PROM is to change the RAM allocation for program statements and variables so it is located in RAM, instead at the end of the machine code to go in PROM. The original start of this allocation is at loc. 35F3, but if you are going to PROM your FOCAL I suggest you save some PROM locations by deleting the heading that is printed as if it were line number 00.00 by the Write command. I retained only the line number zero and a carriage return in my version, since the program expects to print something there. This saves twenty-seven bytes of memory. In my system, I decided to start the RAM storage for statements and data at loc. 4000, so initialization there is as follows:

```

4000 00 ;line number
4001 00 ;of 00.00
4002 0D ;ASCII 'CR'
4003 FE ;PBEG
4004 FF ;VEND

```

¹See 6502 User Notes, issue #16, and errata in issue #17.

To tell FOCAL where to put its statements and variables, some page zero locations need to be changed:

```

002F was D4 35 change to 00 40 ;beginning of RAM allocation
0031 was F2 35 change to 03 40 ;start of user's text
003E was F3 35 change to 04 40 ;start of variable list
0040 was F3 35 change to 04 40 ;start of variables for "erase all"
0042 was F3 35 change to 04 40 ;end of variable list

```

The code to accomplish page zero and page 17 setup and initialize the user RAM is given in Table 3. The code begins at loc. 3677 instead of right after the FOCAL code because I have some other modifications in between; the user will want to relocate this to suit his system anyhow.

Table 1. Table of High-Order Data Bytes Used by POPJ and PUSHJ. Add Offset to Relocate.

Hex Location	Original Contents
2088	23
20B2	23
20D7	29
212F	21
219E	21
21D0	23
21FE	23
2440	2B
2452	29
24BB	29
2502	2B
2516	29
2533	29
2546	29
256A	29
257A	23
25EB	29
29DC	29
29E5	2D
2A45	2B
2A5D	29
2ABE	29
2B97	29
2EFF	29
2F7F	29
2FA3	29
2FE8	29
300D	2B
309E	29
316A	2B
3186	29
31A8	21
34AE	2C

Table 2. FOCAL Initialization

3677	A2 00	COLDST	LDX \$00	;Initialize table & instructions
	BD AO 36	LOOP1	LDA(X) TABL1	;at page zero
	95 20		STA(X)	
	E8		INX	
	E0 BD		CPX \$BD	;Initialize \$17BA-\$17E6 for
	D0 F6		BNE LOOP1	;removal of self-modifying code
	A2 00		LDX \$00	;in FOCAL
	BD 5D 37	LOOP2	LDA(X) TABL2	
	9D DA 17		STA(X) \$17DA	
	E8		INX	
	E0 0D		CPX \$0D	
	D0 F5		BNE LOOP2	;Initialize User RAM
3690	A2 00		LDX \$00	;with line number
	BD 6A 37	LOOP3	LDA(X) TABL3	;zero and data bytes
	9D 00 40		STA(X) \$4000	
	E8		INX	
	E0 05		CPX \$05	;Go to FOCAL cold start
	D0 F5		BNE LOOP3	;page zero constants & code
	4C 00 20		JMP FOCAL	
36A0	{ contents of FOCAL locs. \$0020-\$00DC go here }			TABL1
375C				;Table for patches to
375D	6C 00 00	TABL2		;remove self-modifying
	4C 00 00			;code in FOCAL
	B5 00			
	95 00			
	4C 15 34			;Line no.
367A	00	TABL3		;of 00.00
	00			;ASCII 'CR'
	0D			;PBEG
	FE			;VEND
367E	FF			