

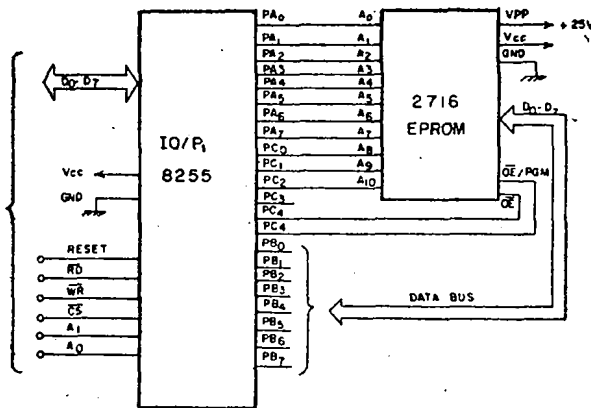
## PRACTICAL DESIGN IDEAS

### EPROMS PROGRAMMING

The instruments used for programming EPROMS at present, such as the Universal PROM programmer are very costly. This note describes an inexpensive method for programming industry-standard EPROMS.

A SIMPLE METHOD is described here for the programming of industry-standard EPROMS such as the 2758 (1K × 8), 2716 (2K × 8) and 2732 (4K × 8) using the Intel 8255 Programmable Peripheral Interface IC. Figure 1 describes the configuration for programming a 2716 EPROM. All the three ports A, B & C of 8255 are defined as output ports. PA0 to PC2 (11 lines) are connected to the address lines, PBO—PB7 to the data line whilst PC4 & PC5 are used for mode selection and program pulse respectively. Programming a particular location is accomplished by setting the corresponding address and data and giving a 50 ms. TTL pulse at  $\overline{CE}/\overline{PGM}$  with  $\overline{OE}$  low and pin  $V_{pp}$  held at 25 V. The complete 8086 microprocessor assembly language program is described below for writing into 2716 EI EPROMS. The data is initially entered in the microprocessor memory and on the execution of the following program, the data is loaded into the EPROM. The procedure was developed using an Intel SDK-86 microprocessor kit.

FIG. 1 EPROM PROGRAMMING



2716 EPROM CONNECTED SDK 86 via 8255 I/O P1 FOR PROGRAMMING

Flowchart of the program is given in Fig. 2. and Assembly language listing in Appendix I.

### ACKNOWLEDGEMENT

The author wishes to thank Dr. V.V.R. Varadachari, Director, National Institute of Oceanography and Dr. E. Desa, Head, Instrumentation Division for the encouragement received and the laboratory facilities extended.

Manuscript received 1981 March 26; revised 1982 January 7.

### FLOWCHART OF THE PROGRAM

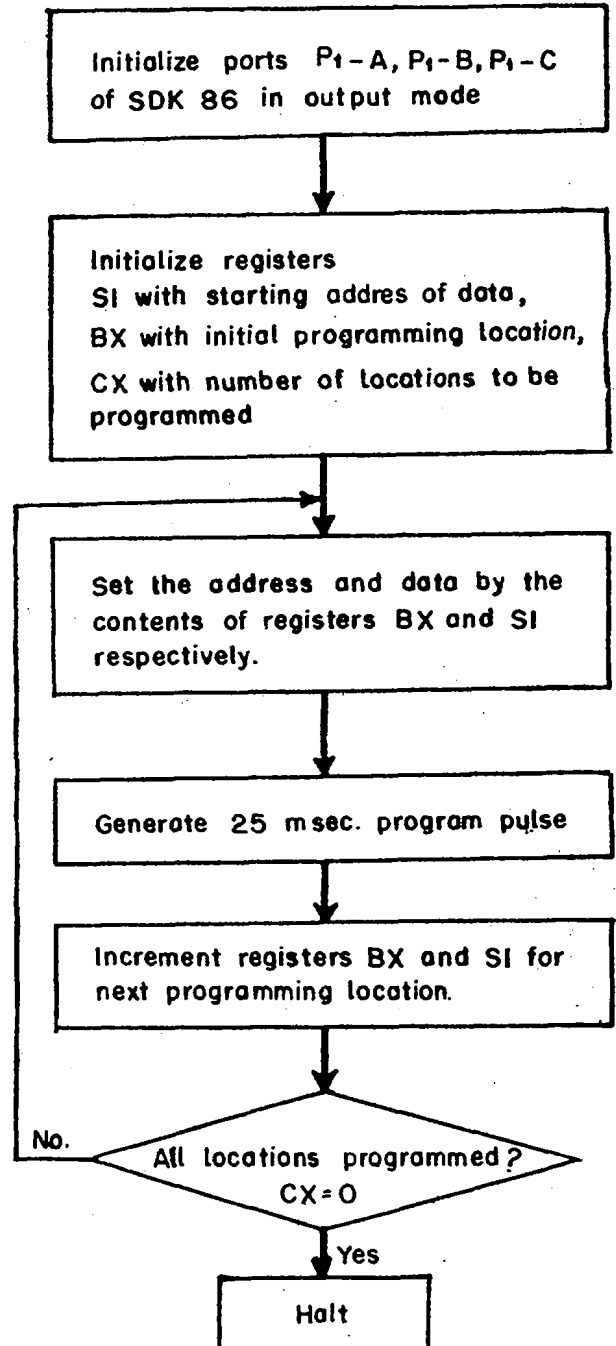


Fig. 2

R.G. PRABHU DESAI  
National Institute of Oceanography,  
Dona Paula 403 004 Goa.

## APPENDIX 1

## Assembly language listing

Address	Opcode	Label	Mnemonic	Comments
100	B0, 80		MOV AL, 80	
102	BA, FF, FF		MOV DX, FFFF	
105	EF		OUT AL	
106	BE, 00, 04		MOV SI, 0400;	Loads SI with the starting address of data.
109	BB, 00, 00		MOV BX, 0000;	Load BX with initial programming location.
10C	B9, 50, 01		MOV CX, 07FF;	Load CX with the number of locations to be programmed.
10F	SB, C5	S1:	MOV AX, BX;	Set address lines
111	BA, F9, FF		MOV DX, FFF9	
114	EE		OUT AL	
115	86, C4		XCHG AH, AL	
117	OC, 20		OR AL, 20	
119	BA, FD, FF		MOV DX, FFFD (PI-C)	
11C	EE		OUT AL	
11D	50		PUSH AX;	Set data lines.
11E	8A, 04		MOV AL, (SI)	
120	BA, FB, FF		MOV DX, FFFB (PI-B)	
123	EE		OUT AL	
124	58		POP AX	
125	OC, 30		OR AL, 30;	Generate program pulse.
127	BA, FD, FF		MOV DX, FFFD (PI-C)	
12A	EE		OUT AL	
12B	51		PUSH CX;	Generate delay.
12C	B9, 00, 20		MOV CX, 2000	
12F	E2, FE,	S2:	Loop S2	
131	59		POP CX	
132	24, 2F		AND AL, 2F	
134	EE		OUT AL	
135	43		INC BX	
136	46		INC SI	
137	F2, D6.		Loop S1	
139	F4		HLT	

## REFERENCES

1. 1980—Intel Component Data Catalogue.
2. SDK86/MCC-86 System Design Kit User's guide.
3. The 8086 Family User's manual from Intel Corporation, 3065 Bowers Avenue, Santa Clara, CA 95051.