

CDP18S842

Run-time BASIC High-Level Language Interpreter

Run-time Basic CDP18S842 is a variant of BASIC2 CDP18S840 and of BASIC3 CDP18S841. It is available in six EPROMs and is especially suited for custom Microboard applications not requiring disk I/O. Run-time BASIC provides an excellent way to generate software quickly in a high-level language for use in any Microboard system. The system can be configured to suit the application. The software for the application is generated by the user in a development system (COSMAC Development System III or IV using BASIC2, or the Microboard Computer Development Systems MCDS using BASIC3) and installed in memory, RAM or ROM. Then, with Run-time BASIC in the system, execution of the user program can begin immediately.

The primary difference between BASIC2 or BASIC3 and Run-time BASIC, other than that there is no program input mode for Run-time BASIC, is that the latter cannot execute PLOAD or PSAVE. See Table I for a list of statements and functions available on Run-time BASIC. The statements CLOSE, DLOAD, DSAVE, WFLN, RFLN, DIN, and DOUT in Run-time BASIC apply to a tape I/O system such as the Microboard Computer Development System MCDS CDP18S693 or CDP18S694. The CDP18S652 Combination Memory and Tape I/O Control Microboard is required for tape I/O with Run-time BASIC.

Memory Requirements

The Run-time BASIC Interpreter is located in memory from 0000H to 2FFFH (H denotes hexadecimal notation) as shown in Fig. 1. The Interpreter starts execution of a user program at 3000H in ROM and generates data in RAM at 9000H. There must be at least 1 kilobyte of RAM available at 9000H for the Interpreter to run. The work pages for Run-time BASIC are at 9000H and 9100H. Any references to the work pages in the user program should be to these addresses and not to the ones for BASIC2 or BASIC3.

Using Run-time BASIC

To generate a Run-time BASIC program from a BASIC2 or BASIC3 program, proceed as follows.

1. Save the BASIC program on tape or disk in ASCII.
2. Determine the RAM requirements for the program.
3. Make a ROM or EPROM containing the program for use with the Run-time Interpreter.

Detailed information on the development of a Run-time Basic program on the CDP18S008 Development System is given in the BASIC2 High-Level-Language Interpreter CDP18S840 User Manual, MPM-840A. Information on the development of a Run-time BASIC program on the CDP18S693, CDP18S694, or CDP18S695 series of Microboard Computer Development Systems (MCDS) is given in the BASIC3 High-Level Language Interpreter User Manual, MPM-841A.

Table I - Statements and Functions available in Run-time BASIC.

Command	CLD	ENINT	
	CLS	FORMAT	
	DISINT		
Comment and Definition	DEFINT	FIXED	REM
	DEG	LET	
	DIM	RAD	
Control	END	GOSUB	NEXT
	EXIT	GOTO	RETURN
	FOR	IF	WAIT
Program Data	DATA	READ	RESTORE
I/O	DMAPT	OUT	QST
	EF	PEEK	STQ
	INP	POKE	TIN
	INPUT	PRINT	TOUT
Tape	CLOSE	DOUT	WFLN
	DIN	DSAVE	
	DLOAD	RFLN	
Machine Language Subroutine	CALL	USR	
Arithmetic	ABS	INT	RND
	ATN	INUM	SGN
	COS	LOG	SIN
	EXP	MOD	SQR
	FNUM	PI	
String	ASC	FVAL	MID\$
	CHR\$	LEN	

CDP18S842

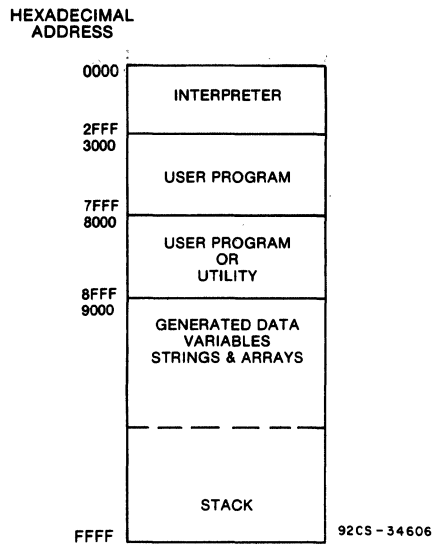


Fig. 1 - Typical memory configuration for Run-time BASIC.

CDP18S843**Assembler/Editor Upgrade Firmware**

The Assembler/Editor Upgrade Firmware CDP-18S843 is provided in three 2716 EPROM's and is intended only for upgrading the Microboard Computer Development System (MCDS) CDP18S693 to the CDP-18S694. (Use of the Assembler requires two cassette tape units.) These EPROM's fit the CDP18S652 Microboard Combination Memory and Tape I/O Control Module provided with the CDP18S693 and provide Editor and Assembler capability.

The Text Editor allows the user to generate ASCII files on cassette tape. These files can be Level I CDP1802 language, BASIC3 instructions with line numbers, or simply text. The Editor output file becomes the input file for the assembler. The Editor commands include: Move pointer to beginning of buffer, Move pointer to end of buffer, Move pointer by n characters, Move pointer by n lines, Define input tape, Append lines, Insert text, Delete

n characters, Delete n lines, Save n lines, Get saved text, Find text, Substitute text, Define output tape, Type n lines, Write n lines to output tape, Write entire buffer to output tape, Print n lines, Return to UT62, and Quit session and restart Editor.

The Assembler allows the user to convert a Level I source file on tape (source code) into an executable machine language program on another tape (object code). The object code can then be loaded into memory by the UT62 Monitor program for execution, or it can be placed in an EPROM by a PROM programmer. The Assembler permits the user to write programs using convenient mnemonic expressions rather than machine language. It is a two-pass assembler with CDP1800-series Level I syntax. The Assembler also provides error messages to assist in debugging.