

## BINARY LOAD AND PUNCH

To easily decrease the amount of time it takes to load a long tape (Cassette or paper) a BINARY formatting technique can be used instead of the conventional ASCII format used by the punch and load routines in MIKBUG. The two following programs, BILOAD and BIPNCH are two such programs necessary for punching your own binary data tapes. The punch routine is designed to automatically punch a program that is in several sections or a program and its program counter. Load your program and the BIPNCH into your system and set up the following locations in the MIKBUG RAM:

A014	Number of sections to be dumped (02 if you have a one piece program and program counter)
A015	Starting address of program dump (1st block)
A016	
A017	End address of program dump (1st block)
A018	
A019	Starting address of second block (or pgm. ctr.)
A01A	
A01B	End address of second block
A01C	
etc.	

(Above similar to setting up A002-etc. for MIKBUG punch)

A028	The hex value of the program counter for your program. The
A029	data in these locations is transferred to A048 and A049 automatically and punched in binary.

Executing the BIPNCH program at 1E04 will punch your program onto either cassette or paper tape. Note - Be sure to have your READ switch on a cassette tape loader such as the AC-30 in the off position during a binary punch.

Since you are dumping in binary rather than ASCII, do not expect to see the usual S11 format as during a MIKBUG punch. All you will see will be random characters.

Using the BILOAD program is quite straightforward. Simply use it the same way you use the "L" command in MIKBUG except that you are executing a loader program at 1703 instead of typing an L. The loader will give you a register dump when loading is complete.

If you desire, the loader program can be put on the beginning of each binary tape to save you time in loading. Use the following procedure to make such a tape.

- 1) Load in BILOAD and BIPNCH into memory. Load in the program to be dumped. Set A048 and Ad49 to 1703. Set A002- A005 to 1700 177F. Set up locations A614, A028, etc. as described earlier.
- 2) Execute the MIKBUG P command. Set A002-A005 to A048 and A049 and execute P.
- 3) Switch to local and put an S9 G on your tape. Be sure to leave a second or two dead time on both sides of the G.
- 4) In the remote mode, change A048 and A049 to 1E04. Type G and the program will be punched in binary (Be sure to have locations A0", etc. set up correctly as described earlier).

The tape made this way will have the following stored on it:

BINLD FORMATTED IN ASCII	BINLD PGM CTR. (ASCII)	S9	G	USER PROGRAM FORMATTED IN BINARY
-----------------------------	------------------------------	----	---	-------------------------------------

IMPORTANT NOTE:

Some terminals (such as the SWTPC CT-1024) will treat a 94<sub>16</sub> the same as a 14<sub>16</sub> (Punch off). The BIPNCH is set up to correct for this. If your terminal does not see a 94<sub>16</sub> as a 14<sub>16</sub> you must change locations 1EB1-1EB4 to NOP'S (01).

If the program you wish to dump or load occupies the same area of memory as either BTPNCH or BILOAD, you will need to re-assemble them to move them to other areas of memory. Be careful because the programs use the EXTENDED addressing mode in several places.

The BILOAD and BIPNCH are very similar to the ones SWTPC will use when formatting long cassette tapes in binary.

			NAM	BILOAD
E1D1		OUTEEEE	EQU	\$E1D1
E1EF		DEL	EQU	\$E1EF
E1F3		DE	EQU	\$E1F3
E1E3		IOUT2	EQU	\$E1E3
E115		DMPREG	EQU	\$E115
E040		LOAD19	EQU	\$E040
E1A5		SAV	EQU	\$E1A5
1700			ORG	\$1700
1700		CKSM	RMB	1
1701		TW	RMB	2
1703	8E A0 47		LDS	#\$A047
1706	8D 49	BILOAD	BSR	LOAD
1708	8D 3C	OVER	BSR	INPUT
170A	81 58		CMP A	#'X
170C	26 FA		BNE	OVER
170E	8D 36		BSR	INPUT
1710	81 31		CMP A	#'1
1712	27 07		BEQ	READ
1714	81 39		CMP A	#'9
1716	26 F0		BNE	OVER
1718	7E E1 15		JMP	DMPREG
171B	7F 17 00	READ	CLR	CKSM
171E	8D 26		BSR	INPUT
1720	16		TAB	
1721	5C		INC B	
1722	8D 22		BSR	INPUT
1724	B7 17 01		STA A	TW
1727	8D 1D		BSR	INPUT
1729	B7 17 02		STA A	TW+1
172C	FE 17 01		LDX	TW
172F	8D 15	STORE	BSR	INPUT
1731	A7 00		STA A	0,X
1733	01		NOP	
1734	A1 00		CMP A	0,X
1736	26 0B		BNE	OUT
1738	08		INX	
1739	5A		DEC B	
173A	26 F3		BNE	STORE
173C	8D 08		BSR	INPUT
173E	7C 17 00		INC	CKSM
1741	27 C5		BEQ	OVER
1743	7E E0 40	OUT	JMP	LOAD19
1746	8D 14	INPUT	BSR	INCHP
1748	36		PSH A	
1749	BB 17 00		ADD A	CKSM
174C	B7 17 00		STA A	CKSM
174F	32		PUL A	
1750	39		RTS	
1751	86 11	LOAD	LDA A	#\$11
1753	BD E1 D1		JSR	OUTEEEE
1756	86 3C		LDA A	#\$3C
1758	B7 80 07		STA A	\$8007
175B	39		RTS	
175C	37	INCHP	PSH B	

```

175D BD E1 A5      JSR   SAV
1760 A6 00      IN1 LDA A 0,X
1762 2B FC      BMI   IN1
1764 6F 02      CLR   2,X
1766 BD E1 F3      JSR   DE
1769 BD E1 EF      JSR   DEL
176C C6 04      LDA B #4
176E E7 02      STA B 2,X
1770 58      ASL B
1771 BD E1 EF IN3 JSR   DEL
1774 0D      SEC
1775 69 00      ROL   0,X
1777 46      ROR A
1778 5A      DEC B
1779 26 F6      BNE  IN3
177B BD E1 EF      JSR   DEL
177E 7E E1 E3     JMP  IOUT2
      END

```

NO ERROR(S) DETECTED

SYMBOL TABLE:

BILOAD	1706	CKSM	1700	DE	E1F3	DEL	E1EF	DMPREG	E115
IN1	1760	IN3	1771	INCHP	175C	INPUT	1746	IOUT2	E1E3
LOAD	1751	LOAD19	E040	OUT	1743	OUTEEEE	E1D1	OVER	1708
READ	171B	SAV	E1A5	STORE	172F	TW	1701		

		NAM	BIPNCH
E1D1	OUTEEEE	EQU	\$E1D1
E0E3	CONTRL	EQU	\$E0E3
A014	MAX	EQU	\$A014
A015	BEGA	EQU	\$A015
A017	ENDA	EQU	\$A017
A028	PCT	EQU	\$A028
1DF7		ORG	\$1DF7
1DF7 A7 00	STRT	STA A	0,X
1DF9 B6 A0 29		LDA A	PCT+1
1DFC A7 01		STA A	1,X
1DFE 20 0F		BRA	BIPNCH
1E00 01	CNTR	FCB	01
1E01	TW	RMB	2
1E03	TEMP	RMB	1
1E04 8E A0 47		LDS	#\$A047
1E07 CE A0 48		LDX	#\$A048
1E0A B6 A0 28		LDA A	PCT
1E0D 20 E8		BRA	STRT
1E0F BD 1E 93	BIPNCH	JSR	PNON
1E12 BD 1E 99		JSR	PNLDR
1E15 FE A0 15	BEG	LDX	BEGA
1E18 FF 1E 01		STX	TW
1E1B F6 A0 18	PUND10	LDA B	ENDA+1
1E1E F0 1E 02		SUB B	TW+1
1E21 B6 A0 17	BEG1	LDA A	ENDA
1E24 B2 1E 01		SBC A	TW
1E27 27 02		BEQ	PUND25
1E29 C6 FF		LDA B	#\$FF
1E2B 86 58	PUND25	LDA A	#'X
1E2D BD E1 D1		JSR	OUTEEEE
1E30 86 31		LDA A	#'1
1E32 BD E1 D1		JSR	OUTEEEE
1E35 37		PSH B	
1E36 5F		CLR B	
1E37 30		TSX	
1E38 8D 69		BSR	PUN
1E3A 32		PUL A	
1E3B 4C		INC A	
1E3C B7 1E 03		STA A	TEMP
1E3F CE 1E 01		LDX	#TW
1E42 8D 5F		BSR	PUN
1E44 8D 5D		BSR	PUN
1E46 FE 1E 01		LDX	TW
1E49 8D 58	PUND30	BSR	PUN
1E4B 7A 1E 03		DEC	TEMP
1E4E 26 F9		BNE	PUND30
1E50 FF 1E 01		STX	TW
1E53 53		COM B	
1E54 37		PSH B	
1E55 30		TSX	
1E56 8D 4B		BSR	PUN

```

1E58 33          PUL B
1E59 FE 1E 01   LDX TW
1E5C 09         DEX
1E5D BC A0 17  CP1  CPX ENDA
1E60 26 B9     BNE PUND10
1E62 B6 1E 00   LDA A CNTR
1E65 B1 A0 14   CMP A MAX
1E68 27 1C     BEQ EXIT
1E6A 7C 1E 00   INC CNTR
1E6D B6 1E 17   LDA A BEG+2
1E70 8B 04     ADD A #4
1E72 B7 1E 17   STA A BEG+2
1E75 B6 1E 23   LDA A BEG1+2
1E78 8B 04     ADD A #4
1E7A B7 1E 23   STA A BEG1+2
1E7D B7 1E 5F   STA A CP1+2
1E80 4C        INC A
1E81 B7 1E 1D   STA A PUND10+2
1E84 20 89     BRA BIPNCH
1E86 86 58     EXIT LDA A #'X
1E88 BD E1 D1   JSR OUTEEEE
1E8B 86 39     LDA A #'9
1E8D BD E1 D1   JSr OUTEEEE
1E90 7E E0 E3   JMP CONTRL
1E93 86 2C     PNON LDA A #812
1E95 BD E1 D1   JSR OUTEEEE
1E98 39        RTS
1E99 C6 05     PNLDR LDA B #5
1E9B 4F        PNULL CLR A
1E9C BD E1 D1   JSR OUTEEEE
1E9F 5A        DEC B
1EA0 26 F9     BNE PNULL
1EA2 39        RTS
1EA3 A6 00     PUN  LDA A 0,X
1EA5 BD E1 D1   JSR OUTEEEE
1EA8 EB 00     ADD B 0,X
1EAA A6 00     LDA A 0,X
1EAC 08        INX
1EAD 81 14     CMP A #$14
1EAF 27 E2     BEQ PNON
1EB1 81 94     CMP A #$94
1EB3 27 DE     BEQ PNON
1EB5 39        RTS
                END

```

NO ERROR(S) DETECTED

SYMBOL TABLE:

BEG	1E15	BEG1	1E21	BEGA	A015	BIPNCH	1E0F	CNTR	1E00
CONTRL	E0E3	CP1	1E5D	ENDA	A017	EXIT	1E86	MAX	A014
OUTEEEE	E1D1	PCT	A028	PNLDR	1E99	PNON	1E93	PNULL	1E9B
PUN	1EA3	PUND10	1E1B	PUND25	1E2B	PUND30	1E49	STRT	1DF7
TEMP	1E03	TW	1E01						