APPENDIX A

S-RECORD OUTPUT FORMAT

The S-record format for output modules was devised for the purpose of encoding programs or data files in a printable format for transportation between computer systems. The transportation process can thus be visually monitored and the S-records can be more easily edited.

S-RECORD CONTENT

When viewed by the user, S-records are essentially character strings made of several fields which identify the record type, record length, memory address, code/data, and checksum. Each byte of binary data is encoded as a 2-character hexadecimal number: the first character representing the high-order 4 bits, and the second the low-order 4 bits of the byte.

The 5 fields which comprise an S-record are shown below:

type	record length	address	code/data	checksum

where the fields are composed as follows:

FIELD	PRINTABLE CHARACTERS	CONTENTS
type	2	S-record type — S0, S1, etc.
record length	2	The count of the character pairs in the record, excluding the type and record length.
address	4, 6, or 8	The 2-, 3-, or 4-byte address at which the data field is to be loaded into memory.
code/data	0-2n	From 0 to n bytes of executable code, memory-loadable data, or descriptive information. For compatibility with teletypewriters, some programs may limit the number of bytes to as few as 28 (56 printable characters in the S-record).
checksum	2	The least significant byte of the one's complement of the sum of the values represented by the pairs of characters making up the record length, address, and the code/data fields.

Each record may be terminated with a CR/LF/NULL. Additionally, an S-record may have an initial field to accommodate other data such as line numbers generated by some time-sharing systems.

Accuracy of transmission is ensured by the record length (byte count) and checksum fields.

S-RECORD TYPES

Eight types of S-records have been defined to accommodate the several needs of the encoding, transportation, and decoding functions. The various Motorola upload, download, and other record transportation control programs, as well as cross assemblers, linkers, and other file-creating or debugging programs, utilize only those S-records which serve the purpose of the program. For specific information on which S-records are supported by a particular program, the user's manual for that program must be consulted. TUTOR, the firmware supplied with the educational computer, supports S0, S1, S2, S8, and S9 records. The S2 and S8 records are not often used however because all of the on-board RAM and ROM can be addressed with a 2-byte address.

An S-record-format module may contain S-records of the following types:

- S0 The header record for each block of S-records. The code/data field may contain any descriptive information identifying the following block of S-records. Under VERSAdos, the resident linker's IDENT command can be used to designate module name, version number, revision number, and description information which will make up the header record. The address field is normally zeroes.

- S1 A record containing code/data and the 2-byte address at which the code/data is to reside.

- S2 A record containing code/data and the 3-byte address at which the code/data is to reside.

- S3 A record containing code/data and the 4-byte address at which the code/data is to reside.

- S5 A record containing the number of S1, S2, and S3 records transmitted in a particular block. This count appears in the address field. There is no code/data field.

- S7 A termination record for a block of S3 records. The address field may optionally contain the 4-byte address of the instruction to which control is to be passed. There is no code/data field.

- S8 A termination record for a block of S2 records. The address field may optionally contain the 3-byte address of the instruction to which control is to be passed. There is no code/data field.

- S9 A termination record for a block of S1 records. The address field may optionally contain the 2-byte address of the instruction to which control is to be passed. Under VERSAdos, the resident linker's ENTRY command can be used to specify this address. If not specified, the first entry point specification encountered in the object module input will be used. There is no code/data field.

Only one termination record is used for each block of S-records. S7 and S8 records are usually used only when control is to be passed to a 3- or 4-byte address. Normally, only one header record is used, although it is possible for multiple header records to occur.

CREATION OF S-RECORDS

S-record-format programs may be produced by several dump utilities, debuggers, VERSAdos' resident linkage editor, or several cross assemblers or cross linkers. On EXORmacs, the Build Load Module (MBLM) utility allows an executable load module to be built from S-records, and has a counterpart utility in BUILDS, which allows an S-record file to be created from a load module.

Several programs are available for downloading a file in S-record format from a host system to an 8-bit microprocessor-based or a 16-bit microprocessor-based system. Programs are also available for uploading an S-record file to or from an EXORmacs system.

EXAMPLE

Shown below is a typical S-record-format module, as printed or displayed:

```
S00600004844521B
S1130000285F245F2212226A000424290008237C2A
S11300100002000800082629001853812341001813
S113002041E900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

The module consists of one S0 record, four S1 records, and an S9 record.

The S0 record is comprised of the following character pairs:

S0 S-record type S0, indicating that it is a header record.

06 Hexadecimal 06 (decimal 6), indicating that six character pairs (or ASCII bytes) follow.

00
00 Four-character 2-byte address field, zeroes in this example.

48
[illegible]

1B The checksum.

The first S1 record is explained as follows:

S1 S-record type S1, indicating that it is a code/data record to be loaded/verified at a 2-byte address.

13 Hexadecimal 13 (decimal 19), indicating that 19 character pairs, representing 19 bytes of binary data, follow.

00
00 Four-character 2-byte address field; hexadecimal address 0000, where the data which follows is to be loaded.

The next 16 character pairs of the first S1 record are the ASCII bytes of the actual program code/data. In this assembly language example, the hexadecimal opcodes of the program are written in sequence in the code/data fields of the S1 records:

OPCODE	INSTRUCTION	
285F	MOVE.L	(A7)+,A4
245F	MOVE.L	(A7)+,A2
2212	MOVE.L	(A2),D1
226A0004	MOVE.L	4(A2),A1
24290008	MOVE.L	FUNCTION(A1),D2
237C	MOVE.L	#FORCEFUNC,FUNCTION(A1)

. (The balance of this code is continued in the
. code/data fields of the remaining S1 records,
. and stored in memory location 0010, etc.)

2A The checksum of the first S1 record.

The second and third S1 records each also contain \$13 (19) character pairs and are ended with checksums 13 and 52, respectively. The fourth S1 record contains 07 character pairs and has a checksum of 92.

The S9 record is explained as follows:

S9 S-record type S9, indicating that it is a termination record.

03 Hexadecimal 03, indicating that three character pairs (3 bytes) follow.

00
00 The address field, zeroes.

FC The checksum of the S9 record.

Each printable character in an S-record is encoded in hexadecimal (ASCII in this example) representation of the binary bits which are actually transmitted. For example, the first S1 record above is sent as:

<table>
<tr><td colspan="4">type</td><td colspan="4">length</td><td colspan="8">address</td><td colspan="8">code/data</td><td></td><td colspan="4">checksum</td></tr>
<tr><td colspan="2">S</td><td colspan="2">1</td><td colspan="2">1</td><td colspan="2">3</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">2</td><td colspan="2">8</td><td colspan="2">5</td><td colspan="2">F</td><td>...</td><td colspan="2">2</td><td colspan="2">A</td></tr>
<tr><td>5</td><td>3</td><td>3</td><td>1</td><td>3</td><td>1</td><td>3</td><td>3</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>2</td><td>3</td><td>8</td><td>3</td><td>5</td><td>4</td><td>6</td><td>...</td><td>3</td><td>2</td><td>4</td><td>1</td></tr>
<tr><td>0101</td><td>0011</td><td>0011</td><td>0001</td><td>0011</td><td>0001</td><td>0011</td><td>0011</td><td>0011</td><td>0000</td><td>0011</td><td>0000</td><td>0011</td><td>0000</td><td>0011</td><td>0000</td><td>0011</td><td>0010</td><td>0011</td><td>1000</td><td>0011</td><td>0101</td><td>0100</td><td>0110</td><td>...</td><td>0011</td><td>0010</td><td>0100</td><td>0001</td></tr>
</table>

APPENDIX B

OPERATION WITH MECHANICAL AND LOW SPEED TERMINALS

Difficulties may be encountered when the MEX68KECB is tied to a mechanical terminal at Port 1. Mechanical terminals are inherently slower than CRT terminals when performing certain functions such as a carriage return because of the physical movement required. The paper printout used with mechanical terminals also presents problems which are not encountered with a CRT terminal. These problems are discussed in this appendix.

INITIALIZATION SEQUENCE FOR MECHANICAL TERMINALS

When a mechanical terminal is interfaced to the educational computer, an added initialization sequence is required. Using the Port Format command for Port 1 (PF1), the user must change the number of null characters sent after each character and/or after each line. Without the correct number of nulls, the TUTOR prompt may or may not be displayed; in some cases, only the last part of the prompt will be displayed. Other transmissions from the educational computer may also be garbled. Mechanical terminals need to receive a number of null characters after each carriage return/line feed and, in some cases, after each character to allow their mechanism to catch up; that is, a carriage return/line feed (CR/LF) sequence requires more time than two printable characters, and the additional nulls fill in the extra time. Without the nulls, part of the message (or prompt) is lost during the CR/LF sequence.

At lower baud rates, mechanical terminals usually require nulls only after a CR/LF and not after each character. For example, a TI 700 Series terminal requires only CR/LF nulls at 110, 150, and 300 baud (refer to Table B-1). When only CR/LF nulls are required, characters are missed at the beginning of each line but the rest of the line is received correctly. At higher baud rates where nulls are required after each character, all characters are unrecognizable until the nulls have been added. The entire line will be garbled without the nulls.

Although received lines can be garbled, the lines transmitted by the terminal do not require nulls and are not garbled. The Port Format command should be used to specify the number of null characters required. All user entries under the PF command (paragraph 3.5.21), including carriage returns, should be entered, regardless of how much of the educational computer's response is received; in some cases, the response may be unintelligible until all parameters have been entered. Table B-1 lists the number of nulls required by a TI 700 series terminal at various baud rates. The number of nulls required by other terminals must be determined by the user.

After all parameters have been entered, several carriage returns or a character followed by a carriage return will be required before the prompt is displayed on terminals which require nulls after each character. Terminals which require only CR/LF nulls should display the prompt as soon as the PF command is complete.

NOTE

A reset changes the Port Format parameters back to their initial values. The user must go through the above initialization sequence any time the RESET button is pushed or after a power-on reset.

TABLE B-1. TI 700 Series Null Requirements

BAUD RATE	CHARACTER NULLS	CR/LF NULLS
110	0	1
150	0	1
300	0	4
1200	3	17
2400	7	2F

PAPER PRINTOUT FOR MECHANICAL TERMINALS

Mechanical terminals suffer from a second difficulty, which is caused by the paper printout. CRT terminals allow erasures and overwrites, whereas paper-listing terminals do not.

The educational computer assembler/disassembler utilizes the overwrite capability of CRT terminals when inputting source lines. Under the Memory Modify command with the disassemble option, bytes in memory are read and the disassembled source line is displayed. The user may enter a new source line, which will result in the original address, object code, and source code being erased and replaced by the new address, object code, and source code. On a paper-type terminal, no erasure is possible; the two lines are written on top of each other and neither is legible.

The same problem occurs at the printer when it is attached. When using a CRT terminal, erasure and overwrite produce a current assembly listing where old source lines are not interspersed with new lines. To get a readable listing using a paper-listing terminal, make all changes and then use the Memory Display command with the disassemble option to produce the listing.

If an error is made when entering a source line using a paper-type terminal, the error indicator, which appears under the field suspected of causing the error, is of little use because the fields of the source line are illegible. This problem can be overcome by forcing the ECB to generate an auto line feed each time a carriage return is entered so that it will not overwrite. The third byte of the 6-byte OPTIONS variable is the auto line feed/no auto line feed flag (refer to paragraph 3.5.21). This byte is initialized to $00, indicating no auto line feed. To force the auto line feed mode, this byte should be set to a non-zero value. The old source line, new source line, and error indicator will now all be legible.

Any utilities supported by a host which use erasures or other screen control commands will cause the same type of problem when the educational computer is operated in the transparent mode.

TERMINAL BAUD RATES

The educational computer will operate at baud rates which range from 110 to 9600. A major source of problems is selecting a baud rate for Port 1 or Port 2 which is not the same as the baud rate of the terminal or host which is connected to the port. The baud rate selected on the educational computer must match the baud rate of the terminal or host connected to the educational computer. However, the baud rate of the terminal and of the host do not need to be the same except when the board is operated in the transparent mode.

When the educational computer is operated in the transparent mode, the terminal port (Port 1) and the host port (Port 2) are tied together. The educational computer is effectivly bypassed. Its only function in the transparent mode is to monitor the information sent by the terminal for the exit character. In the transparent mode, the terminal and host baud rates must be identical.

When using the educational computer in the trace mode or with breakpoints, at relatively low baud rates, it may be desirable to suppress the register display. The fourth byte of the OPTIONS variable (paragraph 3.5.21) determines whether or not the registers will be displayed. This byte is initialized to $00, indicating that the registers will be displayed. If this byte is set to a non-zero value, the registers will not be displayed after each instruction is traced or when a breakpoint is encountered. The registers can be examined, however, with the DF command. The register display flag is set to zero by RESET.

Example:

```
TUTOR  1.X > T
PHYSICAL ADDRESS=00001000
PC=00001004 SR=2700=.S7..... US=FFFFFFFF SS=00000786
D0=00306D4D D1=00000000 D2=00000000 D3=00000000
D4=00306D4D D5=0000002C D6=00000002 D7=00000000
A0=00010040 A1=00000618 A2=000004B8 A3=00000540
A4=00001006 A5=00000540 A6=00000541 A7=00000786
--------------------001004    6D1C                BLT.S    $001022

TUTOR  1.X :> MM 4E9                              Set non-zero register display
0004E9    00 ?F0.                                 flag.

TUTOR  1.X > T
PHYSICAL ADDRESS=00001004
001006    0C000039          CMP.B   #57,D0

TUTOR  1.X :>
```

APPENDIX C

RS-232C SERIAL COMMUNICATIONS

E.I.A. RS-232C STANDARD

Written in 1969 by the Electronic Industries Association (EIA), RS-232C is a serial communications standard established to define electrical and mechanical requirements for interconnecting data communications equipment (DCE) and data terminal equipment (DTE). The standard describes both synchronous and asynchronous serial binary communications with data rates ranging from zero to 20,000 bits/second. Twenty-five signal lines are described by the standard, although most are not used in typical applications. Table C-1 summarizes key features of the E.I.A. RS-232C standard.

TABLE C-1. E.I.A. RS-232C Standard

PARAMETER	RS-232C
Line length (recommended maximum - may be exceeded with proper design.)	50 ft.
Input Z	3k to 7k ohm 2500 pF
Maximum frequency (baud)	20k baud
Transition time (time in undefined area between "1" and "0") tr = 10 to 90%	4% of bit period or 1 ms
dV/dt (wave shaping)	30 V/us
Mark (Data "1") Space (Data "0")	-3 V +3 V
Common mode voltage (for balanced receiver)	-
Output Z	-
Open-circuit output voltage (V_o)	$3\ V < \lvert V_o \rvert < 25V$
V_t = loaded V_o	$5 < \lvert V_o \rvert < 15V$ 3k to 7k ohm load
Short circuit current	500 mA
Power-off leakage (V_o applied to unpowered device)	> 300 ohm $2\ V < \lvert V_o \rvert < 25V$ V_o applied
Minimum receiver input for proper V_o	> ± 3 V

The standard connector used for RS-232C compatible equipment is a 25-signal subminiature "D" type. Table C-2 lists the pin number, signal name, and signal description.

TABLE C-2. RS-232C Signal Description

RS-232C PIN NUMBER	RS-232C SIGNAL NAME	DESCRIPTION AND SIGNAL DIRECTION
1	AA	Frame ground
2	BA	Transmitted data (to DCE)
3	BB	Received data (from DCE)
4	CA	Request to send (to DCE)
5	CB	Clear to send (from DCE)
6	CC	Data set ready (from DCE)
7	AB	Signal ground
8	CF	Received line signal detector (from DCE)
9	--	Positive DC test voltage
10	--	Negative DC test voltage
11	--	Unassigned
12	SCF	Secondary received line signal detector (from DCE)
13	SCB	Secondary clear to send (from DCE)
14	SBA	Secondary transmitted data (to DCE)
15	DB	Transmitter signal element timing (from DCE)
16	SBB	Secondary received data (from DCE)
17	DD	Receiver signal element timing (from DCE)
18	--	Unassigned
19	SCA	Secondary request to send (to DCE)
20	CD	Data terminal ready (to DCE)
21	CG	Signal quality detector (from DCE)
22	CE	Ring indicator (from DCE)
23	CH/CI	Data rate selector (to/from DCE)
24	DA	Transmitter signal element timing (to DCE)
25	--	Unassigned

MEX68KECB RS-232C INTERFACE

Ports 1 and 2 of the MC68000 Educational Computer Board support asynchronous serial communications as described by the RS-232C standard. Because transmit and receive clocks are not sent out on the interface, synchronous communications are not supported. Port 1 constitutes a DCE or modem interface type; that is, data terminal equipment is connected to Port 1. Port 2 is a DTE interface and connects to data communication equipment. Baud rates at each port range from 110 to 9600 baud.

Of the 25 signal lines described in Table C-2, the educational computer supports a set of seven. These are:

BA - Transmitted data - TxDATA
BB - Received data - RxDATA
CA - Request to send - RTS
CB - Clear to send - CTS
CC - Data set ready - DSR
CF - Received line signal detector - DCD
CD - Data terminal ready - DTR

In addition, there are two ground signals:

AB - Signal ground - GND
AA - Frame ground

The frame ground is not connected to the educational computer's signal ground, but can be connected externally if necessary. Tables 8-3 and 8-4 list the pin number, signal mnemonic, and signal description for Port 1 connector J3 and Port 2 connector J4, respectively.

The following paragraphs are a description of the signal lines supported by Ports 1 and 2. The format used is:

Signal name
Signal direction
Signal function description

1. TxDATA Transmitted data

   Serial data output from terminal (DTE) to modem (DCE)

   The line/signal through which the terminal (DTE) sends data to the modem (DCE).

2. RxDATA Received data

   Serial data input to terminal (DTE) from modem (DCE)

   The line/signal through which the modem (DCE) sends data to the terminal (DTE).

3. RTS Request to send

Control output from terminal (DTE) to modem (DCE).

The line/signal through which the terminal (DTE) requests permission to transmit data to the modem (DCE).

Assertion of RTS instructs the DCE to prepare to receive data from the DTE and to signify that it is ready to receive by asserting CTS. However, RTS is not monitored at Port 1; it is assumed that Port 1 is always ready to receive data. CTS is activated any time the terminal (DTE) asserts DTR indicating that it is ready to transmit or receive data.

At Port 2 RTS is asserted upon power up to prepare DCE connected to Port 2 for data reception.

4. CTS Clear to send

Control input to terminal (DTE) from modem (DCE).

The line/signal through which the modem (DCE) acknowledges the acceptance of a terminal (DTE) request to send data.

As stated above, Port 1 asserts CTS any time an active level is received from the DTE on DTR.

Port 2 receives CTS from the DCE connected to Port 2 and will interrupt data transmission when inactive.

5. DSR Data set ready

Control input to terminal (DTE) from modem (DCE).

The line/signal through which the modem (DCE) indicates its on-line, in-service, or active status.

Port 1 activates DSR whenever an active level is received on DTR. Port 1 is always on-line.

Port 2 uses only the CTS input from the DCE to indicate whether data may be sent. DSR is not used in making the decision.

6. DTR Data terminal ready

Control output from terminal (DTE) to modem (DCE).

The line/signal through which the terminal (DTE) indicates its on-line, in-service or active status.

DTR is used by Port 1 to enable and disable the transmission of data via the $\overline{\text{CTS}}$ input of the Port 1 ACIA -- MC6850. When $\overline{\text{CTS}}$ is driven high, transmission will stop following the completion of any in-process transmission. Port 2 activates DTR as part of the power-up/reset firmware. A write to the ACIA control register which causes the $\overline{\text{RTS}}$ output to go low will activate DTR at Port 2.

7. DCD Signal Detect

Control input to terminal (DTE) from modem (DCE).

The line/signal through which the modem (DCE) indicates that the communication channel to which the modem (DCE) interfaces (the other/non-terminal side of the modem) is in an acceptable active state. This signal has meaning only in a communication channel (i.e., telephone line) context. DCD is off when no signal is being received or when the received signal is unsuitable for demodulation. While Port 1 implements a DCE or modem interface, the communications is exclusively digital. There is no need to test the suitability of the signal. DCD, at Port 1, indicates only that DTR has been received from the DTE.

Port 2 does not monitor DCD. Again, all signals are exclusively digital.

MEX68KECB NON-COMPLIANCE WITH RS-232C

In addition to being a functional subset of the full RS-232C standard, the educational computer does not comply strictly to the signal specifications. In addition to signal definition and timing, the RS-232C standard specifies driver, receiver, and interface voltage and impedance levels (refer to Table C-1). The MC6850 ACIA's used in the serial interface are NMOS devices operating with a +5V supply and cannot meet the interface voltage and impedance requirements. Two linear integrated circuits, the MC1488 RS-232C line driver and the MC1489A RS-232C line receiver, provide the required buffering and drive to meet the specifications.

The maximum rate of voltage change is specified as 30 V/us in the RS-232C standard. The MC1488 line drivers have an inherent slew rate which is much too fast. The current limited output of the device can be used to control this slew rate by connecting a capacitor to each driver output. The required capacitance value is given by the formula:

$$C = I_{OS} \times \frac{\Delta T}{\Delta V}$$

where C is the capacitance in picofarads, I_{OS} is the output short-circuit current in microamps, and $\Delta T / \Delta V$ is 1/slew rate in microseconds per volt. A 330 pF capacitor on each output of the MC1488 will guarantee a worst case slew rate of 30 V/us. Bear in mind, however, that this capacitance includes cabling capacitance.

These capacitors are not present on the educational computer board.

SUGGESTION/PROBLEM REPORT

MICROSYSTEMS
QUALITY • PEOPLE • PERFORMANCE

Motorola welcomes your comments on its products and publications. Please use this form.

To: Motorola Inc.
Microsystems
2900 S. Diablo Way
Tempe, Arizona 85282
Attention: Publications Manager
Maildrop DW164

Product: ______________________ Manual: ______________________

COMMENTS: ____

Please Print

Name ______________________ Title ______________________

Company ______________________ Division ______________________

Street ______________________ Mail Drop __________ Phone __________

City ______________________ State ______________ Zip __________

For Additional Motorola Publications
Literature Distribution Center
616 West 24th Street
Tempe, AZ 85282
(602) 994-6561

Microsystems Field Service Support
(800) 528-1908
(602) 829-3100

MOTOROLA

MOTOROLA Semiconductor Products Inc.

P.O. BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.

15196-5 PRINTED IN USA (12/83) MPS 7M