THE TARBELL CASSETTE INTERFACE

Plugs directly into ALTAIR* 8800 or IMSAI 8080 Computer. Serial-Parallel and Parallel-Serial Conversion performed on board.

It's fast — Up to 540 bytes per second (2200 bits per inch) with high-quality cassette recorder. 187 bytes per second (800 bits per inch) suggested standard with medium quality ($39.95) recorder. 27 bytes per second (with modification) for Kansas City format. 1000 bytes per second @ 10 inches per second with PHI-DECK.

Encoding method — Phase encoded self-clocking method requires only one channel, and withstands large amounts of wow and flutter. Has been used in the computer industry for many years. Can also be used to generate and detect Kansas City format tapes with small modification.

Will work with most audio cassette recorders. Has been used with reel-reel tape recorders. Can easily be adapted to automatically (software) controlled digital cassette units. The J. C. Penny #6536 is presently being recommended ($39.95). Suggested tape is Scotch Low-Noise, High-Density audio tape.

The device code (address) to which the interface responds is selectable in increments of 4 by an onboard dip-switch.

Four extra status lines are available to read external conditions.

Four extra control lines are available, which may be used to drive relays for extra cassette units, start-stop control, or controlled cassette drives, such as the PHI-DECK.

Two spare IC slots are provided on the board to do your own thing.

Software provided includes input/output driver routines, bootstrap, read-only memory program, and a BASIC program for saving and loading data. A Processor-Technology Software Package #1, which is an editor and assembler combined, and which has been extensively modified for saving and loading files to and from cassette, is available separately.

The 37-page owner’s manual includes assembly instructions and drawings, parts and pin-function lists, soldering, cleaning, and installation notes, operating instructions, initial adjustment procedures, notes about using the PHI-DECK, start-stop control, interrupts, and writing programs for the cassette interface, the software mentioned above, and information about all the integrated circuits on the board.

The parts provided include all resistors, capacitors, integrated circuits, cassette cables, ribbon cables, dip connector, low-noise cassette with test stream, and a high-quality double-sided board with plated-through holes and gold-plated edge-connector fingers.

The first deliveries were made in September, 1975. Delivery time is from one to three weeks after receiving order. If you are not completely satisfied with the operation of the cassette interface, for whatever reason, the unit may be returned for refund or free repair within 90 days after you purchased it.


*ALTAIR is a trademark/tradename of MITS, Inc.

TARBELL ELECTRONICS — 20620 S. Leapwood Ave., Suite P
Carson, California 90746 — Phone (213) 538-4251
I HAVE BEEN USING AN INEXPENSIVE AUDIO CASSETTE RECORDER IN MY HOME-DESIGNED COMPUTER SYSTEM SINCE 1972. I HAVE OVER 600 FILES ON CASSETTES, MOSTLY ABOUT 4 KBYTES EACH. MY ESTIMATE IS THAT THE ERROR RATE IS LESS THAN 1 ERROR IN 1,000,000 BITS. I SAY THIS BECAUSE I CAN USUALLY RECORD 30 4 KBYTE FILES ON ONE SIDE OF A C-60 CASSETTE WITHOUT ANY ERRORS. THIS INTERFACE GAVE ME VERY GOOD SERVICE WHILE I WAS WRITING THE DISK OPERATING SYSTEM FOR MY 500 KBYTE DISK. SINCE I STARTED USING MY DISK SYSTEM A FEW YEARS AGO, THE CASSETTE HAS SERVED AS BACKUP STORAGE - A RELIABLE PLACE TO STORE DATA AND PROGRAMS AFTER THEY ARE DEBUGGED.

THE ENCODING METHOD I USE IS VERY SIMPLE, AND HAS BEEN IN USE IN INDUSTRY FOR QUITE SOME TIME. PICTURE A SHIFT REGISTER WHICH IS LOADED WITH THE DATA TO BE RECORDED. THE REGISTER IS THEN CLOCKED WITH A SQUARE WAVE. THE OUTPUT OF THE SHIFT REGISTER IS EXCLUSIVE-OR ED WITH THE CLOCK, PRODUCING THE BI-PHASE DATA. THIS DATA GOES DIRECTLY TO THE CASSETTE RECORDER'S INPUT. THE MOST DIFFICULT PART OF THE PROCESS IS RECOVERING THE DATA. MANY LONG HOURS WERE SPENT STUDYING THIS PROBLEM AND TRYING DIFFERENT METHODS. IN THE ORIGINAL INTERFACE THIS WAS ACCOMPLISHED WITH A 760 HIGH-SPEED COMPARATOR, A 74121 NON-RETRIGGERABLE ONE-SHOT, AND A DOUBLE-GLITCH GENERATOR MADE WITH AN EXCLUSIVE-OR GATE. SINCE THEN, THE 8T20 HAS BEEN DEVELOPED, WHICH COMBINES THESE THREE FUNCTIONS ON A SINGLE CHIP, AND IS THE UNIT USED IN THE ALTAIR INTERFACE.

THE PRESENT DESIGN IS EVEN MORE RELIABLE THAN THE PREVIOUS ONE, AND IS CAPABLE OF RECORDING AND RECOVERING ERROR-FREE DATA AT A RATE OF 540 BYTES PER SECOND ON A STANDARD AUDIOCASSETTE RECORDER. (YES, THAT IS OVER 2200 BITS PER INCH!) I AM STILL, HOWEVER, ENCOURAGING USERS TO EXCHANGE DATA RECORDED AT 187 BYTES PER SECOND (1500 BITS PER SECOND, 800 BITS PER INCH). THE MAIN ADVANTAGE OF THIS METHOD OVER OTHERS IS IT'S ABILITY TO WITHSTAND A LARGE AMOUNT OF WOW AND FLUTTER, WHICH MAY BE INTRODUCED BY CHEAP RECORDERS, AND STILL RECORD THE DATA RELIABLY. THIS FEATURE STEMS FROM THE SELF-CLOCKING NATURE OF THE RECORDED SIGNAL: THE RECOVERED CLOCK VARIES RIGHT ALONG WITH THE DATA, SO THAT THE SPEED VARIATIONS ARE ESSENTIALLY IGNORED. THE MAIN DISADVANTAGE OF THIS METHOD IS THAT IT REQUIRES GOOD LOW-NOISE TAPE AND A DECENT FREQUENCY RESPONSE ON THE CASSETTE UNIT. THE CASSETTE UNIT I'VE BEEN USING LATELY (J.C. PENNY #6536) HAS A FREQUENCY RESPONSE OF 80-8,000 HZ. THE MOST IMPORTANT PART IS THE HIGH END. THESE REQUIREMENTS ARE DUE TO THE HIGH SPEED OF THE INTERFACE, AND WOULD BE THE SAME FOR ANY HIGH SPEED DEVICE.

THE SPEED MAY NOT SEEM VERY IMPORTANT TO YOU NOW. BUT A GOOD PORTION OF YOUR TIME IS GOING TO BE SPENT SAVING AND LOADING DATA, PROGRAMS, AND OTHER TEXT. THERE IS A WORLD OF DIFFERENCE BETWEEN LOADING BASIC AT SAY, 30 BYTES/SEC (4 MINUTES), AND AT 187 BYTES/SEC (40 SEC). IT DOESN'T SEEM LIKE MUCH, BUT WHEN YOU HAVE TO DO IT OVER, AND OVER, AND OVER .... IT GETS TO BE A BIT MUCH. ESPECIALLY WHEN YOU'RE DEVELOPING YOUR OWN PROGRAMS, AND THEY TEND TO RUN AMUCK AND WIPE OUT CORE. THINK ABOUT IT, THEN BUY THE TARBEll CASSETTE INTERFACE. THE ONLY METHOD PROVEN WITH TIME. ASK YOUR FRIEND WHO HAS ONE.
SELECTING A CASSETTE UNIT FOR DIGITAL RECORDING

First of all, the most expensive cassette recorders are not necessarily the best for recording digital data. There are several factors that combine to make a good unit for the hobbyist:

1. It should have a good high-frequency response, preferably up to at least 8,000 Hz.

2. It should have a tone control, so that the inherent frequency response may be realized.

3. Although automatic volume control is more convenient from an operational point of view, it also requires a few seconds of settling time before starting to record.

4. If it does not have automatic volume control, it is good to have a recording level meter. This allows easier adjustment for the correct recording level.

5. It is very important to have a digital counter. This makes it possible to quickly locate the desired program among several.

6. It should be capable of running directly on the AC line. Batteries tend to make the motor get slower as they wear.

7. It is handy to have an auxiliary input, so that a fairly high level may be fed to the recorder, and noise kept to a minimum.

8. A remote input jack is valuable to control start-stop during assembler and compiler operations.

9. Jacks for mic, aux, remote, and earphone are usually included, but you should check to make sure they are there, anyway.

10. Low wow and flutter characteristics are important, but are much more important when using an asynchronous interface, which is not self-clocking.

11. If it is desired to do automatic rewind, fast-forward, and record/playback switching under program control, you must purchase a recorder that has these facilities. Another feature to look for on this type of unit is a way to keep track of where you are on the cassette tape.

12. People have experienced problems with Panasonic recorders. I am presently recommending the J.C. Penney Model 6536 at $39.95 as the best buy I know, and several people are using them with my interface with good results. Other brands that I know have been successful are Sony and Realistic.

Of course, none of the above items is absolutely necessary for recording digital data on an audio cassette. But the more of these requirements that are filled, the more capable your unit will be, and the easier it will be to use.
IDEAS FOR USING THE CASSETTE INTERFACE

SAVING AND LOADING PROGRAMS

Programs may be toggled into memory, loaded from paper tape, or loaded by some other means. They may then be dumped onto cassette using the cassette output routine. The output routine itself may be dumped along with the other program, so that it will be available later for further dumps. Routines may then be loaded from cassette, modified, and dumped back out to cassette in a continuous process of development.

USING A BACKUP

A backup is a method of making sure that valuable programs or data is not lost. One simple way of providing a backup is to record a particular program in two different places on the same tape, or on two different tapes. You may want to go back and forth between the two copies, each time the program is changed. In this way, you always have a copy of the last program, and only the most recent changes are lost if a power failure or other equipment trouble develops during the process of saving.

STARTING AND STOPPING AUTOMATICALLY

Most audio cassette recorders have an input labeled "remote." This is normally operated from a switch on the microphone, so that the recorder can be started and stopped while dictating. This input can be used to start and stop the recorder under program control from the computer. One of the 4 extra control lines coming from the cassette interface (DIP-socket pins 5, 6, 7, or 8) can be used to drive a relay which would have its contacts connected to the recorder remote jack. The main requirements are that the 40 mA available from the control line be able to drive the relay, and that the relay contacts are able to handle the current into the remote jack, which can be as high as one ampere.

OPERATING WITH MORE THAN ONE CASSETTE RECORDER

Sometimes it is desirable to operate with two or more cassette units. With the Tarbell cassette interface, it is already possible to read from one cassette recorder, while writing onto another. This is because the input and output sections are entirely independent, and may be programmed separately. If it is desired to read from one of two units, and write onto another, such as during a merging operation, relays may be used to switch back and forth between the two input units, under control of one of the 4 control lines. If, however, it is necessary to read simultaneously from two different units, or write different information onto two units at the same time, it will be necessary to have two cassette interfaces.
RANDY MILLER'S CHESS PROGRAM

This program runs under Altair 8K Basic 3.2, and will play a game of chess against the user. It requires a memory of about 20 KBytes total for the program, the Basic interpreter, and the variables. Your version of 8K Basic 3.2 should be modified to do a CLOAD from the Tarbell cassette interface.

Move the tape to the middle of the leader (high-pitched sound), then type in CLOAD C, then start the tape playing, then hit return. The console will come back with "OK" when the computer has loaded the chess program. Then type RUN.

When the chess program asks for your move, type two digits from one to eight, then a comma, then two more digits. This tells the program where the piece is that you want to move, and where you want to move it to. The chess board is set up with the squares numbered as shown in figure A.

<table>
<thead>
<tr>
<th>FIGURE A</th>
<th>FIGURE B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8</td>
<td>1 PAWN</td>
</tr>
<tr>
<td>11 21 31 41 51 61 71 81</td>
<td>2 KNIGHT</td>
</tr>
<tr>
<td>2 12 22 32 42 52 62 72 82</td>
<td>3 BISHOP</td>
</tr>
<tr>
<td>3 13 23 33 43 53 63 73 83</td>
<td>4 ROOK</td>
</tr>
<tr>
<td>4 14 24 34 44 54 64 74 84</td>
<td>5 QUEEN</td>
</tr>
<tr>
<td>5 15 25 35 45 55 65 75 85</td>
<td>6 KING</td>
</tr>
<tr>
<td>6 16 26 36 46 56 66 76 86</td>
<td></td>
</tr>
<tr>
<td>7 17 27 37 47 57 67 77 87</td>
<td></td>
</tr>
<tr>
<td>8 18 28 38 48 58 68 78 88</td>
<td></td>
</tr>
</tbody>
</table>

So, for example, to move your pawn from square 17 to square 15 you would type 17,15 when it asks for your move. If you would like to see the position of all the pieces on the board, you may type 0,0 when it is your move. The pieces are identified by the numbers 1 through 6 as shown in figure B.

To have the computer give a running account of the moves it is considering and the evaluation each move receives, type in the following lines:

```
681 PRINT PS(D,1);PS(D,2);
722 PRINT EV
```

Don't expect the program to come back with it's move real quick. It generally takes a few minutes to make up it's mind. This shouldn't bother you, since most human players take as long. Don't be surprised if you beat the computer. Since the program is fairly small, compared to the big chess-playing programs, it isn't hard to beat. In fact, I've beat it 3 times out of 3, and I don't consider myself a very good player.

Tarbell Electronics does not provide support for this program. All we claim is that it does play chess, and that we will probably update from time to time as people find bugs or make improvements. We will not automatically send out these improvements, and if a new tape or documentation is required, the customer is expected to pay the usual price of $6. California residents please add 6%.
The main purpose of this software is to allow development of assembly-language programs. Programs may be entered from the keyboard in standard assembly-language format, edited, assembled, and saved on cassette in either source or machine format. These programs may be then loaded back into memory at some future time for listing, editing or running. Following is a summary of the commands. All address and byte information is in hex.

**FILE /NAME/ AAAA** - Creates a file of name NAME at address AAAA (hex). Up to six text files can be in memory at the same time. The file just created is the current file.

**FILE /NAME/ O** - This command deletes the file named NAME.

**FILE /NAME/** - Makes the named file the current file.

**FILE** - Prints the starting and ending addresses of the current file, and it's name.

**FILES** - Lists the name, starting and ending address for each of the files in the system. Current file at top.

**DUMP SSSS EEEE** - Dumps the contents of memory from SSSS to EEEE.

**EXEC AAAA** - Executes the machine-language program at address AAAA.

**EXEC /NAME/** - Searches the cassette for named file, loads and exec.

**ENTR AAAA** - Enter data into memory: B1 B2 B3 B4/

**LIST NNNN** - Lists the current file starting at line NNNN.

**DELT NNNN MMMM** - Deletes lines NNNN through MMMM from current file.

**NNNN (text entered)** - Enter a line of text into the current file. Always use 4 decimal digits. It works similar to BASIC, in that the numbered lines are ordered automatically.

**ASSM AAAA** - Assembles current file into address AAAA.

**ASSM AAAA BBBB** - Assembles at address AAAA, but puts code at BBBB.

**CUST** - Execute a customer (user) routine at address E000.

**SFIL** - Save the current file on cassette.

**LFIL** - Load the current file from cassette. The name of the file on tape must match the name of the current file. Be sure there is enough room in memory.

**CFIL** - Check a cassette file written with the SFIL command above, for errors, without overwriting current file.

**AFIL** - Append a file from cassette to the current file.

**SAVE SSSS EEEE** - Save a block of memory from SSSS to EEEE on cassette.

**LOAD SSSS** - Load a block of memory from cassette starting at SSSS.

**CHEK SSSS** - Check a cassette file written with SAVE command.

**NAME /NAME/** - Rename the current file to NAME.

**RNUM N** - Renumber the current file by increment N.

**NLIS NNNN** - Lists the current file at NNNN without line numbers.

This software is useful, but may not be completely free of bugs. It loads into the first 4096 bytes of memory, and uses about 2048 bytes following, for tables and scratch pad. Ctl-C escapes from any printing. Ctl-W freezes printing until another character is typed. The package is available from Tarbell Electronics for the prices listed below:

- **Cassette Tape with software and instructions:** $5.00
- **A reassembled and patched (updated) listing:** $5.00
INSTRUCTIONS FOR CASSETTE READ-ONLY MEMORY PROGRAM 1 (ROMP1)  
AUG 12, 1976

THIS PROGRAM, WHICH WILL RUN IN EITHER READ-ONLY OR READ-WRITE MEMORY, ALLOWS THE USER TO PERFORM THE 6 FUNCTIONS LISTED BELOW:

LETTER EXAMPLE DESCRIPTION
E E3000 EXECUTE PROGRAM AT 3000 (HEX).
C C100020E7 CHECK CHECKSUM OF CASSETTE RECORD.
S S GENERATE SYNC STREAM.
O 020000000 OUTPUT RECORD OF LENGTH 2000, STARTING AT 0000.
I I10003000 INPUT RECORD OF LENGTH 1000, STARTING AT 3000.
L L47DF1257 LOAD AND EXECUTE, LENGTH 47DF, STARTING AT 1257.


THE SECOND AND THIRD BYTES OF THIS PROGRAM SPECIFY WHERE THE STACK IS LOCATED, AND SHOULD BE THE TOP OF SOME RANDOM-ACCESS MEMORY (NOT READ-ONLY MEMORY) WHICH IS NOT USED FOR OTHER THINGS. THE STACK WILL THEN EXPAND DOWN FOR ABOUT 8 OR 10 BYTES FROM THIS ADDRESS.
ASSN. FFOO E400

FF00
FF00
FF00
FF00
FF00 31 00 EC
FF03 3E 0D
FF05 CD 56 FF
FF08 3E 0A
FF0A CD 56 FF
FF0D 3E 3F
FF0F CD 56 FF
FF12 CD 64 FF
FF15 FE 45
FF17 CC 36 FF
FF1A FE 43
FF1C CC BO FF
FF1F FE 53
FF21 CC 73 FF
FF24 FE 4F
FF25 CA 7B FF
FF29 FE 49
FF2B CC BO FF
FF2E FE 4C
FF30 CC D0 FF
FF33 CC BO FF
FF36
FF36
FF36
FF36 CD 3A FF
FF39 E9
FF3A
FF3A
FF3A 21 00 00
FF3D CE 04
FF3F CD 64 FF
FF42 29
FF43 29
FF44 29
FF45 29
FF46 CD 50 FF
FF49 85
FF4A 6F
FF4B 0D
FF4C C2 3F FF
FF4F C9
FF50
FF50
FF50
FF50 D6 30
FF52 FE 0A
FF54 D8
FF55 D6 07
FF57 C9
FF58

0005 *
0010 *** READ-ONLY MEMORY PROGRAM ***
0015 *
0020 START LXI SP, 00C00H SET STACK PTR.
0024 NVI A0DH PRINT CR, LF.
0026 CALL PTCN
0028 NVI A0AH
0040 CALL PTCN G0N CONSOLE.
0042 NVI A,? PRINT "?"
0043 CALL PTCN G0N CONSOLE.
0050 CALL RDCN READ KEYBOARD.
0060 CPI 'E' IF E,
0070 CZ EXEC EXECUTE A PROGRAM.
0080 CPI 'C' IF C,
0090 CZ CINR G0 TO INPUT ROUTINE.
0100 CPI 'S' IF S,
0110 CZ CSYNC G0 TO SYNC GEN.
0120 CPI 'O' IF O,
0130 JZ COUTR G0 TO CASS OUT.
0140 CPI 'I' IF I,
0150 CZ CINR G0 TO CASS IN.
0160 CPI 'L' IF L,
0170 CZ CINR DO A LOAD & GE.
0200 JMP START START OVER.

6005 *
6010 *** EXECUTE THE PROGRAM AT THE ADDRESS ***
6020 *
6030 EXEC CALL AREX READ ADDRESS FROM KB.
6040 PCHL JUMP TO IT.
7000 *
7002 *** CONVERT UP TO 4 HEX DIGITS TO BINARY ***
7004 *
7005 AREX LXI H, 0 GET A 16-BIT ZERO.
7006 NVI C,4 COUNT OF 4 DIGITS.
7009 AREX CALL RDCN READ A BYTE.
7010 DAD H SHIFT 4 LEFT.
7012 DAD H
7014 DAD H
7016 DAD H
7018 CALL AHS1 CONVERT TO BINARY.
7020 ADD L
7022 LVI LA
7024 DCR C 4 DIGITS?
7026 JNZ AREX KEEP READING.
7028 RET RETURN FROM AREX.

7100 *
7102 *** CONVERT AN ASCII DIGIT TO HEX ***
7104 *
7106 AHS1 SUI 4E ASCII BIAS.
7108 CPI 10 DIGIT 0-10
7110 RC
7112 SUI 7 ALPHA BIAS.
7114 RET RETURN FROM AHS1.

8002 *** PRINT REGISTER A ON CONSOLE ***
FFBC  F1
FFBD  E5
FFBE  F5
FFBF  06  00
FFC1  CD  F2  FF
FFC4  4F
FFC5  F1
FFC6  F5
FFC7  FE  43
FFC9  79
FFCA  CA  05  FF
FFCD  77
FFCE  4C
FFCF  47
FFD0  23
FFD1  1B
FFD2  97
FFD3  2A
FFD4  C2  C1  FF
FFD7  B3
FFD8  C2  C1  FF
FFDB  CD  F2  FF
FFDE  B5
FFDF  3E  45
FFE1  C2  EC  FF
FFE4  F1
FFE5  FE  4C
FFE7  C2  EC  FF
FFEA  E1
FFEB  E9
FFEC  CD  58  FF
FFEF  C3  00  FF
FFFF  DB  6E
FFF4  E5  10
FFF6  C2  F2  FF
FFF9  DB  6F
FFFF  C9
FFFC
FFFC
FFFC
FFFC

9365  P0P  PSW  GET  CONTROL  CHAR.
9370  PUSH  H  SAVE  STARTING  ADDRESS
9375  PUSH  PSW  UNDER  CONTROL  CHAR.
9380  MVI  B,0  SET  CHECKSUM  =  0.
9390  CALL  CIN  READ  A  BYTE  FROM  CASS.
9400  MVI  C,A  SAVE  IT  IN  REG  C.
9410  POP  PSW  GET  CONTROL  CHAR.
9420  PUSH  PSW  SAVE  IT.
9430  CPI  'C'  IS  IT  A  C?
9440  MVI  A,C  GET  BACK  DATA  BYTE.
9450  JZ  CIN0  IF  C  DON'T  STORE  IT.
9460  MVI  M,A  IF  NOT,  DO  STORE  IT.
9470  CIN  ADD  B  ADD  TO  CHECKSUM.
9480  CIN  ADD  TO  C.
9490  INX  H  INCREMENT  POINTER.
9500  DCY  D  DECREMENT  COUNTER.
9510  CLI  CLEAR  A.
9520  CMP  D  IF  D  NOT  =  0.
9530  JNZ  CILOP  READ  MORE.
9540  CMP  E  IF  E  NOT  =  0.
9550  JNZ  CILOP  READ  MORE.
9560  CALL  CIN  READ  LAST  BYTE.
9570  CMP  B  COMPARE  TO  CHECKSUM.
9580  MVI  A,'E'  PRINT  E  FOR  ERROR.
9590  JNZ  CERR  PRINT  NOW  IF  ERR0R.
9600  POP  PSW  RECOVER  CTL  CHAR.
9610  CPI  'L'  IF  IT'S  NOT  L.
9620  JNZ  CERR  DON'T  EXECUTE.
9630  P0P  H  OTHERWISE,  EXECUTE.
9640  PCHL  AT  STARTING  ADDRESS.
9650  CERR  CALL  PTCN  PRINT  C,E,  OR  I.
9660  JRP  START  READ  STATUS.
9670  CIN  IN  CASC  LOOK  AT  BIT  4.
9710  ANI  IOH  WAIT  UNTIL  LOW.
9720  JNZ  CIN  READ  DATA  FROM  CASS.
9730  IN  CASD  RETURN  FROM  CIN.
9740  RET  CASSETTE  DATA  PORT.
9795  CASD  EQU  6FH  CASSETTE  STATUS  PORT.
9996  PSW  EQU  6  EQU  6.
The frequency of the output oscillator may be changed from its normal value of 3000 hz to 4800 hz by putting a resistor of about 14 kohms in parallel with R6 (27 kohm).
**KANSAS CITY INPUT ROUTINE**

**READS ONE BYTE INTO REGISTER A.**

*NOTE:* IN ORDER TO USE THIS ROUTINE,

MAKE THE CHANGES ON YOUR BOARD

* AS SHOWN IN THE DIAGRAM BELOW.

**

**

PUSH B
SAVE B,C.

BLIN MVI C,$8
SET COUNT=8 BITS.

MVI A,$10H
RESET RECEIVER.

OUT CASC

CALL CASS IN
READ A BYTE (BIT).

ANI $3Ch
LOOK AT MIDDLE 4 BITS.

JZ ONE
IF XX0000XX, BIT=1.

CPI $3Ch
IF XX1111XX, BIT=1.

JZ ZER0

CPI $14H
IF XX0101XX, BIT=0.

JZ ZER0
IF XX1010XX, BIT=0.

MUTE BE NEISE.

CPI
CLEAR CARRY.

JMP A
DO REST LIKE ONE.

STC
SET CARRY=1.

MIV A,B
GET RESULT.

RAR
SHIFT CARRY INTO MSB.

MIV B,A
PUT RESULT BACK.

DCR C
DONe WITH BYTE?

JNZ BLIN
IF NOT, KEEP READING.

CALL CASS IN
READ AN EXTRA BYTE.

MIV A,B
GET RESULT.

POP B
RESTORE B,C.

RET
RETURN.

CASS IN
READ CASS STATUS.

IN CASC
LOOK AT INPUT BIT.

IN CASD
READ DATA BYTE.

CASC
STATUS/CONTROL PORT.

CASD EQU $6Fh
DATA PORT.

Cut the traces going to pins 1 and 2 of IC 9, install a 74L30 in one of the spare slots, and make the circuit look like this.

This change will not affect the normal operation at 187 bytes/sec.
**EOO 0005**  
**EOO 0010** *** READ-ONLY MEMORY PROGRAM ***  
**EOO 0015** * MODIFIED FOR KANSAS-CITY *  
**EOO 0016**  

**EOO 0020** START LXI SP, $6000H SET STACK PTR.  
**EOO 0024** MVI A, $0DH PRINT CR, LF.  
**EOO 0026** CALL PTCN  
**EOO 0028** MVI A, $0AH  
**EOO 002A** CALL PTCN  
**EOO 002C** MVI A, '?' PRINT '?'  
**EOO 002E** CALL PTCN  
**EOO 0030** MVI A, $00H PRINT CR.  
**EOO 0032** MVI A, $80H PRINT LF.  
**EOO 0034** CALL AHEX READ ADDRESS FROM FRIOM.  
**EOO 0036** MVI A, $0AH  
**EOO 0038** CALL PTCN  
**EOO 003A** MVI A, 'E' IF E.  
**EOO 003C** READ-ONLY MEMORY PROGRAM.  
**EOO 003E** CMP H, $00H READ KEYBOARD.  
**EOO 0040** CALL PTCE EXECUTE A PROGRAM.  
**EOO 0042** CMP H, 'C' IF C.  
**EOO 0044** CZ CINR GO TO INPUT ROUTINE.  
**EOO 0046** CMP H, 'S' IF S.  
**EOO 0048** CZ CSYNC GO TO SYNC GEN.  
**EOO 004A** CMP H, '0' IF 0.  
**EOO 004C** JZ COUTB GO TO Cass OUT.  
**EOO 004E** CMP H, '1' IF 1.  
**EOO 0050** CZ CINR GO TO Cass IN.  
**EOO 0052** CMP H, 'L' IF L.  
**EOO 0054** CZ CINR DO A LOAD & G0.  
**EOO 0056** JMP START START OVER.  
**EOO 0060**  
**EOO 0064** EXECUTE THE PROGRAM AT THE ADDRESS ***  
**EOO 0066** EXEC CALL AHEX READ ADDRESS FROM KB.  
**EOO 0068** PCHL JUMP TO IT.  
**EOO 0070**  
**EOO 0074** CONVERT UP T0 4 HEX DIGITS T0 BINARY *  
**EOO 0076** AHEX LXI H, $0 GET A 16-BIT ZERO.  
**EOO 0078** MVI C, $4 COUNT OF 4 DIGITS.  
**EOO 007A** AHEI CALL RDCN READ A BYTE.  
**EOO 007C** DAD H SHIFT 4 LEFT.  
**EOO 007E** DAD H  
**EOO 0080** DAD H  
**EOO 0082** DAD H  
**EOO 0084** CALL AHS1 CONVERT TO BINARY.  
**EOO 0086** ADD L  
**EOO 0088** MOV L, A  
**EOO 008A** DCR C 4 DIGITS?  
**EOO 008C** JNZ AHEI KEEP READING.  
**EOO 008E** RET RETURN FROM AHS1.  
**EOO 0090**  
**EOO 0094** CONVET AN ASCII DIGIT TO HEX ***  
**EOO 0096** AHS1 SUI 48 ASCII BIAS.  
**EOO 0098** CPI 10 DIGIT 0-10  
**EOO 009A** RC  
**EOO 009C** SUI 7 ALPHA BIAS.  
**EOO 009E** RET RETURN FROM AHS1.  
**EOO 00A0**  
**EOO 00A4** PRINT REGISTER A ON CONSOLE ***  
**EOO 00A6**
5E58 F5
5E59 DB 00
5E5B E6 80
5E5D C2 59 5E
5E60 F1
5E61 D3 01
5E63 C9
5E64
5E64
5E64 DB 00
5E66 E6 01
5E68 C2 64 5E
5E6B DB 01
5E6D E6 7F
5E6F CD 58 5E
5E72 C9
5E73
5E73
5E73 3E 55
5E75 CD 9C 5E
5E78 C3 73 5E
5E7B
5E7B
5E7B CD 3A 5E
5E7E EB
5E7F CD 3A 5E
5E82 06 00
5E84 7E
5E85 CD 9C 5E
5E88 80
5E89 47
5E8A 23
5E8B 1B
5E8C 97
5E8D BA
5E8E C2 E4 5E
5E91 BB
5E92 C2 E4 5E
5E95 76
5E96 CD 9C 5E
5E99 C3 00 5E
5E9C C3 00 5F
5E9F 5F
5E9F
5E9F F5
5EA0 3E 10
5EA2 D3 6E
5EA4 CD 3A 5E
5EA7 EB
5EA8 CD 3A 5E
5EAB F1
5EAC E5
5EAD F5
5EAE 06 00
5EB0 CD E1 5E
5EB3 4F
5EB4 F1
5EB5 F5
5EB6 FE 43
5EB8 79
5ECA 8006 PTCN PUSH PSW SAVE REG A.
5ECA 8008 PTLOP IN C0NC READ PRINTER STATUS.
5ECA 8010 ANI 80H IF BIT 7 NOT 0,
5ECA 8012 JNZ PTLOP WAIT TILL IT IS.
5ECA 8014 POP PSW THEN RECOVER A,
5ECA 8016 OUT C0NDF AND PRINT IT.
5ECA 8018 RET RETURN FROM PTCN.
5ECA 8100 ** READ FROM CONSOLE TO REGISTER A ***
5ECA 8104 *
5ECA 8106 RDCN IN C0NC READ KEYBOARD STAT.
5ECA 8108 ANI 1 IF BIT 1 NOT 0.
5ECA 8110 JNZ RDCN REPEAT UNTIL IT IS.
5ECA 8112 IN C0NDF READ FROM KEYBOARD.
5ECA 8114 ANI 7FH STRIP OFF MSB.
5ECA 8115 CALL PTCN ECH0 ONTO PRINTER.
5ECA 8116 RET RETURN FROM RDCN.
5ECA 8116 C0NC EQU 0 CONSOLE STATUS PORT.
5ECA 8120 C0NDF EQU 1 CONSOLE DATA PORT.
5ECA 9100 ** SYNC CODE GENERATOR PROGRAM ***
5ECA 9103 *
5ECA 9104 CSYNC MVI A, 55H WRITE SYNC BYTE.
5ECA 9106 CALL COUT 0NT0 CASSETTE.
5ECA 9108 JMP CSYNC KEEP DOING IT.
5ECA 9200 ** CASSETTE INTERFACE OUTPUT ROUTINE ***
5ECA 9204 *
5ECA 9206 COUTR CALL AHEX READ BLOCK LENGTH.
5ECA 9210 XCHG PUT INTO D,E.
5ECA 9212 CALL AHEX READ STARTING ADDR.
5ECA 9216 MVI B, 0 START CHECKSUM = 0.
5ECA 9226 C0LEP M0V A,N GET DATA FROM MEMORY.
5ECA 9228 CALL COUT SEND TO CASSETTE.
5ECA 9230 ADD B ADD T0 CHECKSUM.
5ECA 9232 MOV B, A
5ECA 9234 INX H INCREMENT POINTER.
5ECA 9236 DCX D DECREMENT COUNTER.
5ECA 9238 SUB A CLEAR A.
5ECA 9240 CMP D IF D NOT 0,
5ECA 9242 JNZ C0LEP REPEAT LOOP.
5ECA 9244 CMP E IF E NOT 0.
5ECA 9246 JNZ C0LEP REPEAT LOOP.
5ECA 9248 MOV A, B GET CHECKSUM.
5ECA 9250 CALL COUT OUTPUT IT.
5ECA 9252 JMP START GET ANOTHER COMMAND.
5ECA 9254 COUT JMP SFOOH G0 TO BLOCK.
5ECA 9300 ** CASSETTE INPUT ROUTINE ***
5ECA 9304 *
5ECA 9310 CINR PUSH PSW SAVE CONTROL CHAR.
5ECA 9320 MVI A, 10H USE BIT 4 IN REG A.
5ECA 9330 OUT CASC TO RESET CASS INT.
5ECA 9340 CALL AHEX READ BLOCK LENGTH.
5ECA 9350 XCHG PUT INTO D,E.
5ECA 9360 CALL AHEX READ STARTING ADDRESS.
5ECA 9365 POP PSW GET CONTROL CHAR.
5ECA 9370 PUSH H SAVE STARTING ADDRESS.
5ECA 9375 POP PSW UNDER CONTROL CHAR.
5ECA 9380 MVI B, 0 SET CHECKSUM = 0.
5ECA 9390 C0LEP CALL CIN READ A BYTE FROM CASS.
5ECA 9400 MOV C, A SAVE IT IN REG C.
5ECA 9410 POP PSW GET CONTROL CHAR.
5ECA 9420 PUSH IS IT A C?
5ECA 9430 MOV A, C GET BACK DATA BYTE.
5EB9 CA BD 5E 9450      JZ CIN0  IF C, DON'T STORE IT.
5EBC 77           9460      MOV MA  IF NOT, DO STORE IT.
5EBD 80           9470      CIN0 ADD B  ADD TO CHECKSUM.
5EBE 47           9480      MOV BA
5EBF 23           9490      INX H  INCREMENT POINTER.
5EC0 1B           9500      DCX D  DECREMENT COUNTER.
5EC1 97           9510      SUB A  CLEAR A.
5EC2 BA           9520      CMP D  IF D NOT = 0,
5EC3 C2 B0 5E     9530      JNZ CIL0P  READ MORE.
5EC6 BB           9540      CMP E  IF E NOT = 0,
5EC7 C2 B0 5E     9550      JNZ CIL0P  READ MORE.
5ECA CD E1 5E     9560      CALL CIN  READ LAST BYTE.
5ECD B8           9570      CMP B  COMPARE TO CHECKSUM.
5ECE 3E 45        9580      NVI A,'E'  PRINT E FOR ERROR.
5ED0 C2 DB 5E     9590      JNZ CERR  PRINT NOW IF ERROR.
5ED3 F1           9600      POP PSV  RECOVER CTL CHAR.
5ED4 FE 4C        9610      CPI 'L'  IF IT'S NOT L,
5ED6 C2 DB 5E     9620      JNZ CERR  DON'T EXECUTE.
5ED9 E1           9630      POP H  OTHERWISE, EXECUTE
5EDA E9           9640      PCHL  AT STARTING ADDRESS.
5EDB CD 58 5E     9650      CERR CALL PTCN  PRINT C,E, OR I.
5EDE C3 00 5E     9660      JMP START
5EE1 C3 40 5F     9670      CIN JMP 5F40H  GO TO BLIN.
5EE4              9700      CASD EQU 6FH  CASSETTE DATA PORT.
5EE4              9996      CASD EQU 6EH  CASSETTE STATUS PORT.
5EE4              9997      CASC EQU 6 E
5EE4              9998      PSW EQU 6
5EE4              9999      SP EQU 6
THE
TARBELL
CASSETTE
INTERFACE

TARBELL ELECTRONICS
20620 S. Leapwood Ave., Suite P
Carson, California 90746
(213) 538-4251
TABLE OF CONTENTS

PAGE

1. Specifications
2. Interface History and Sales Pitch
3. Selecting a Cassette Unit for Digital Recording
4. The Kansas City (Byte/Lancaster/CUTS) Format

5. Parts List
6. Assembly Instructions
7. Assembly Drawings
8. Modifications
9. Soldering, Cleaning, and Installation Notes
10. Initial Adjustment Instructions

11. Operating Instructions
12. Bootstrap and Sync Code Generator Programs
13. Output Routine With Checksum
14. Input Routine With Checksum
15. How to Save and Load Data From a BASIC Program
16. Controlling the Start-Stop Function

17. Theory of Operation - Output Section
18. Theory of Operation - Input Section
19. Schematic - Output Section
20. Schematic - Input Section
21. Timing Diagram
22. Pin Function List

23. If You Have Problems
24. Ideas For Using the Cassette Interface
25. Modifications on Cassette BASIC CSAVE and CLOAD
26. Processor Technology Software Package #1 Modified
27. Writing Programs For the Cassette Interface
28. Interrupt Control
29. PHI-DECK Adapter Information

30. 74L00, 7403, 74L11, 7404, 74L04, 7406, 7425, 7408, 74L30
31. 74L73, 7473, 74L74
32. 74L75, 74L86
33. 7496, DM8131
34. 74L164, NE555
35. 8T20

36. Customer Evaluation and Registration

January 14, 1977
THE TARBELL CASSETTE INTERFACE (FOR ALTAIR 8800 USERS)

SPEED: UP TO 540 BYTES PER SECOND (2200 BITS PER INCH).
187 BYTES PER SECOND FOR TARBELL STANDARD 800 BITS/INCH.
30 BYTES PER SECOND FOR "BYTE/LANCASTER" STANDARD.

ENCODING METHOD: PHASE-ENCODED (EXCLUSIVE-OR OF CLOCK AND DATA).
SELF-CLOCKING (CLOCK VARIES ALONG WITH TAPE SPEED).
USED ON MY OWN SYSTEM FOR THE LAST 4 YEARS.
CAN BE USED TO GENERATE "BYTE/LANCASTER" TAPES.

CASSETTE: WILL WORK WITH MOST AUDIO CASSETTE UNITS. MAY BE ADAPTED
TO AUTOMATIC DIGITAL CASSETTE UNITS. WILL ALSO WORK
WITH REGULAR REEL-REEL TAPE RECORDERS. I HAVE BEEN USING
A REALISTIC (RADIO SHACK) CTR-19 AND A J.C. PENNY 6536
($39.95). TAPE SHOULD BE OF LOW-NOISE TYPE.

8192 BYTE LOAD TIME: 15 SEC @ 540 BYTES PER SECOND.
43 SEC @ 187 BYTES PER SECOND.
4 MINUTES @ 30 BYTES PER SECOND.

DEVICE-CODE EASILY SELECTED WITH ON-BOARD DIP-SWITCH.

STATUS: 4 EXTRA STATUS LINES AVAILABLE FOR INPUT.

CONTROL: 4 EXTRA CONTROL LINES AVAILABLE FOR OUTPUT, WHICH MAY
BE USED TO DRIVE RELAYS FOR EXTRA CASSETTE UNITS.
2 SPARE IC SLOTS TO LET YOU DO YOUR OWN THING.

COMPATIBILITY: PLUGS DIRECTLY INTO ALTAIR 6800 OR IMSAI 8080.
HAS SERIAL-PARALLEL AND PARALLEL-SERIAL CONVERSION
ON BOARD. PATCHES PROVIDED FOR POPULAR SOFTWARE.

SOFTWARE: COMES WITH COMPLETE SET OF INPUT/OUTPUT SUBROUTINES,
BOOTSTRAP, AND "BYTE STANDARD" (LANCASTER) SOFTWARE.

COST: $120 FOR COMPLETE KIT, $175 ASSEMBLED AND CHECKED-OUT.

MANUAL:
ASSEMBLY INSTRUCTIONS AND DRAWING, PARTS AND PIN FUNCTION LISTS
SOLDERING, CLEANING, AND INSTALLATION NOTES; OPERATING INSTRUCTIONS
INITIAL ADJUSTMENT PROCEDURES, INPUT/OUTPUT ROUTINES WITH CHECKSUM
BOOTSTRAP PROGRAM AND TEST-STREAM GENERATOR PROGRAM

PARTS:
ALL RESISTORS, CAPACITORS, AND INTEGRATED CIRCUITS
CASSETTE CABLE, RIBBON CABLE, AND DIP CONNECTOR
LOW-NOISE CASSETTE WITH TEST STREAM
DOUBLE-SIDED BOARD WITH PLATED-THRU HOLES AND GOLD EDGE CONNECTOR

WARRANTY: IF NOT COMPLETELY SATISFIED, RETURN BOARD FOR REFUND
OR FREE REPAIR WITHIN 90 DAYS AFTER PURCHASE.

FIRST DELIVERIES WERE MADE IN SEPTEMBER, 1975. DELIVERY IS
1 TO 3 WEEKS AFTER RECEIVING ORDER. THE 25-PAGE MANUAL IS AVAILABLE
AT $4. CALIFORNIA RESIDENTS PLEASE ADD 6% SALES TAX. MAKE
CHECK OR MONEY ORDER PAYABLE TO TARBELL ELECTRONICS.
HISTORY AND SALES PITCH

I HAVE BEEN USING AN INEXPENSIVE AUDIO CASSETTE RECORDER IN MY HOME-DESIGNED COMPUTER SYSTEM SINCE 1972. I HAVE OVER 600 FILES ON CASSETTES, MOSTLY ABOUT 4 KBYTES EACH. MY ESTIMATE IS THAT THE ERROR RATE IS LESS THAN 1 ERROR IN 1,000,000 BITS. I SAY THIS BECAUSE I CAN USUALLY RECORD 30 4 KBYTE FILES ON ONE SIDE OF A C-60 CASSETTE WITHOUT ANY ERRORS. THIS INTERFACE GAVE ME VERY GOOD SERVICE WHILE I WAS WRITING THE DISK OPERATING SYSTEM FOR MY 500 KBYTE DISK. SINCE I STARTED USING MY DISK SYSTEM A FEW YEARS AGO, THE CASSETTE HAS SERVED AS BACKUP STORAGE - A RELIABLE PLACE TO STORE DATA AND PROGRAMS AFTER THEY ARE DEBUGGED.

THE ENCODING METHOD I USE IS VERY SIMPLE, AND HAS BEEN IN USE IN INDUSTRY FOR QUITE SOME TIME. PICTURE A SHIFT REGISTER WHICH IS LOADED WITH THE DATA TO BE RECORDED. THE REGISTER IS THEN CLOCKED WITH A SQUARE WAVE. THE OUTPUT OF THE SHIFT REGISTER IS EXCLUSIVE-ORED WITH THE CLOCK, PRODUCING THE BI-PHASE DATA. THIS DATA GOES DIRECTLY TO THE CASSETTE RECORDER'S INPUT. THE MOST DIFFICULT PART OF THE PROCESS IS RECOVERING THE DATA. MANY LONG HOURS WERE SPENT STUDYING THIS PROBLEM AND TRYING DIFFERENT METHODS. IN THE ORIGINAL INTERFACE THIS WAS ACCOMPLISHED WITH A 760 HIGH-SPEED COMPARATOR, A 74121 NON-RETRIGGERABLE ONE-SHOT, AND A DOUBLE-GLITCH GENERATOR MADE WITH AN EXCLUSIVE-OR GATE. SINCE THEN, THE 8T20 HAS BEEN DEVELOPED, WHICH COMBINES THESE THREE FUNCTIONS ON A SINGLE CHIP, AND IS THE UNIT USED IN THE ALTAIR INTERFACE.

THE PRESENT DESIGN IS EVEN MORE RELIABLE THAN THE PREVIOUS ONE, AND IS CAPABLE OF RECORDING AND RECOVERING ERROR-FREE DATA AT A RATE OF 540 BYTES PER SECOND ON A STANDARD AUDIO CASSETTE RECORDER. (YES, THAT IS OVER 2200 BITS PER INCH!) I AM STILL, HOWEVER, ENCOURAGING USERS TO EXCHANGE DATARecorded AT 187 BYTES PER SECOND (1500 BITS PER SECOND, 800 BITS PER INCH). THE MAIN ADVANTAGE OF THIS METHOD OVER OTHERS IS IT'S ABILITY TO WITHSTAND A LARGE AMOUNT OF WOW AND FLUTTER, WHICH MAY BE INTRODUCED BY CHEAP RECORDERS, AND STILL RECOVER THE DATA RELIABLY. THIS FEATURE STEMS FROM THE SELF-CLOCLING NATURE OF THE RECORDED SIGNAL: THE RECOVERED CLOCK VARIeS RIGHT ALONG WITH THE DATA, SO THAT THE SPEED VARIATIONS ARE ESSENTIALLY IGNORED. THE MAIN DISADVANTAGE OF THIS METHOD IS THAT IT REQUIRES GOOD LOW-NOISE TAPE, AND A DECENT FREQUENCY RESPONSE ON THE CASSETTE UNIT. THE CASSETTE UNIT I'VE BEEN USING LATELY (J.C. PENNY #6536) HAS A FREQUENCY RESPONSE OF 80-8,000 HZ. THE MOST IMPORTANT PART IS THE HIGH END. THESE REQUIREMENTS ARE DUE TO THE HIGH SPEED OF THE INTERFACE, AND WOULD BE THE SAME FOR ANY HIGH SPEED DEVICE.

THE SPEED MAY NOT SEEM VERY IMPORTANT TO YOU NOW. BUT A GOOD PORTION OF YOUR TIME IS GOING TO BE SPENT SAVING AND LOADING DATA, PROGRAMS, AND OTHER TEXT. THERE IS A WORLD OF DIFFERENCE BETWEEN LOADING BASIC AT SAY, 30 BYTES/SEC (4 MINUTES), AND AT 187 BYTES/SEC (40 SEC). IT DOESN'T SEEM LIKE MUCH, BUT WHEN YOU HAVE TO DO IT OVER, AND OVER, AND OVER... IT GETS TO BE A BIT MUCH. ESPECIALLY WHEN YOU'RE DEVELOPING YOUR OWN PROGRAMS, AND THEY TEND TO RUN AMUCK AND WIPE OUT CORE.

THINK ABOUT IT, THEN BUY THE TARBEiL CASSETTE INTERFACE. THE ONLY METHOD PROVEN WITH TIME. ASK YOUR FRIEND WHO HAS ONE.
SELECTING A CASSETTE UNIT FOR DIGITAL RECORDING

FIRST OF ALL, THE MOST EXPENSIVE CASSETTE RECORDERS ARE NOT NECESSARILY THE BEST FOR RECORDING DIGITAL DATA. THERE ARE SEVERAL FACTORS THAT COMBINE TO MAKE A GOOD UNIT FOR THE HOBBYIST:

1. IT SHOULD HAVE A GOOD HIGH-FREQUENCY RESPONSE, PREFERABLY UP TO AT LEAST 8,000 Hz.

2. IT SHOULD HAVE A TONE CONTROL, SO THAT THE INHERENT FREQUENCY RESPONSE MAY BE REALIZED.

3. ALTHOUGH AUTOMATIC VOLUME CONTROL IS MORE CONVENIENT FROM AN OPERATIONAL POINT OF VIEW, IT ALSO REQUIRES A FEW SECONDS OF SETTLING TIME BEFORE STARTING TO RECORD.

4. IF IT DOES NOT HAVE AUTOMATIC VOLUME CONTROL, IT IS GOOD TO HAVE A RECORDING LEVEL METER. THIS ALLOWS EASIER ADJUSTMENT FOR THE CORRECT RECORDING LEVEL.

5. IT IS VERY IMPORTANT TO HAVE A DIGITAL COUNTER. THIS MAKES IT POSSIBLE TO QUICKLY LOCATE THE DESIRED PROGRAM AMONG SEVERAL.

6. IT SHOULD BE CAPABLE OF RUNNING DIRECTLY ON THE AC LINE. BATTERIES TEND TO MAKE THE MOTOR GET SLOWER AS THEY WEAR.

7. IT IS HANDY TO HAVE AN AUXILLIARY INPUT, SO THAT A FAIRLY HIGH LEVEL MAY BE FED TO THE RECORDER, AND NOISE KEPT TO A MINIMUM.

8. A REMOTE INPUT JACK IS VALUABLE TO CONTROL START-STOP DURING ASSEMBLER AND COMPILER OPERATIONS.

9. JACKS FOR MIC, AUX, REMOTE, AND EARPHONE ARE USUALLY INCLUDED, BUT YOU SHOULD CHECK TO MAKE SURE THEY ARE THERE, ANYWAY.

10. LOW WOW AND FLUTTER CHARACTERISTICS ARE IMPORTANT, BUT ARE MUCH MORE IMPORTANT WHEN USING AN ASYNCRONOUS INTERFACE, WHICH IS NOT SELF-CLOCKING.

11. IF IT IS DESIRED TO DO AUTOMATIC REWIND, FAST-FORWARD, AND RECORD/PLAYBACK SWITCHING UNDER PROGRAM CONTROL, YOU MUST PURCHASE A RECORDER THAT HAS THESE FACILITIES. ANOTHER FEATURE TO LOOK FOR ON THIS TYPE OF UNIT IS A WAY TO KEEP TRACK OF WHERE YOU ARE ON THE CASSETTE TAPE.

12. PEOPLE HAVE EXPERIENCED PROBLEMS WITH PANASONIC RECORDERS. I AM PRESENTLY RECOMMENDING THE J.C. PENNEY MODEL 6536 AT $39.95 AS THE BEST BUY I KNOW, AND SEVERAL PEOPLE ARE USING THEM WITH MY INTERFACE WITH GOOD RESULTS. OTHER BRANDS THAT I KNOW HAVE BEEN SUCCESSFUL ARE SONY AND REALISTIC.

OF COURSE, NONE OF THE ABOVE ITEMS IS ABSOLUTELY NECESSARY FOR RECORDING DIGITAL DATA ON AN AUDIO CASSETTE. BUT THE MORE OF THESE REQUIREMENTS THAT ARE FILLED, THE MORE CAPABLE YOUR UNIT WILL BE, AND THE EASIER IT WILL BE TO USE.
USING THE TARBELL CASSETTE INTERFACE FOR THE KANSAS CITY FORMAT

Some time ago, there was a meeting of various cassette interface manufacturers to determine a standard for exchange of programs and data on cassettes among computer hobbyists. The format that was proposed as a result of the meeting is a modified version of the coding technique described by Don Lancaster in the first issue of "BYTE" Magazine. In this format, each 8-bit byte is written on tape in an asynchronous format, with one start bit (zero), 8 data bits (zero or one), and two stop bits (ones). A one is defined as 8 cycles at 2400 bits per second, and a zero is defined as 4 cycles at 1200 bits per second. This provides a data transfer speed of 300 baud, or a little less than 30 bytes per second, and may be generated and decoded using a variety of techniques.

Since the standard is fairly slow, it suggests that many people may want to have two methods available. One that provides for the Kansas City (BYTE/Lancaster) format, and another that is much faster, to speed program loading and development. The Tarbell Cassette Interface may easily be modified for both methods.

First, the output oscillator frequency will have to be raised from 3000 hz to 4800 hz. This is because a higher bit density is required of the tape, although the actual data transfer rate is much slower. A one may be generated by writing a word of all zeroes (00000000), and a zero may be generated by writing a word of alternating ones and zeroes (01010101). An output subroutine converts each byte to be written in this format from parallel to serial form (required only for this format).

On the input side, the adjustment of the potentiometer (R8) will have to be changed for the higher frequency. The sync detector circuit (IC's 9 and 10) will have to be changed so that it recognizes the alternating bit pattern as a sync byte in addition to the normal sync byte of E6 (hex). An input subroutine converts each byte from it's serial form to it's parallel form (required only for this format).

Using the method outlined above, the Tarbell Cassette Interface can be modified so that a double-pole, single-throw switch will determine which frequency will be used. The software determines the format. Another alternative is to change to the higher frequency permanently, so that no switch is necessary. The disadvantage of this is that you would have to readjust the potentiometer to read tapes made with the standard 3000 hz oscillator (187 bytes per second), and that a slightly higher frequency response is required on the part of your recorder.

A description of the hardware modifications, and listings of the subroutines for the operations described above are available upon request. Please include a self-addressed stamped envelope.
CASSette Interface Parts List

**Integrated Circuits**

<table>
<thead>
<tr>
<th>REF NOS</th>
<th>DESCRIPTION</th>
<th>QTY</th>
<th>PT NO</th>
<th>+5 GND</th>
<th>-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QUAD 2-INPUT AND</td>
<td>1</td>
<td>7405</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>DUAL J-K FLIP-FLOP</td>
<td>1</td>
<td>7473</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>6-BIT SHIFT REGISTER</td>
<td>1</td>
<td>741164</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>4, 23</td>
<td>DUAL TYPE-D FLIP-FLOP</td>
<td>2</td>
<td>74L74</td>
<td>4, 14</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>5-VOLT REGULATOR</td>
<td>1</td>
<td>LM309K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7, 13, 19, 25</td>
<td>QUAD 2-INPUT NAND O/C</td>
<td>4</td>
<td>7403</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>8, 29</td>
<td>DUAL J-K FLIP-FLOP</td>
<td>2</td>
<td>74L73</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>BI-DIRECTIONAL ONE-SHOT</td>
<td>1</td>
<td>7T20</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>9, 31</td>
<td>HEX INVERTER</td>
<td>2</td>
<td>74L04</td>
<td>4, 14</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>5-INPUT NAND</td>
<td>1</td>
<td>74L30</td>
<td>4, 14</td>
<td>7</td>
</tr>
<tr>
<td>14</td>
<td>TRIPLE 3-INPUT AND</td>
<td>1</td>
<td>74L11</td>
<td>4, 14</td>
<td>7</td>
</tr>
<tr>
<td>20</td>
<td>HEX INVERTER</td>
<td>1</td>
<td>7404</td>
<td>4, 14</td>
<td>7</td>
</tr>
<tr>
<td>21, 27</td>
<td>5-BIT SHIFT REGISTER</td>
<td>2</td>
<td>7496</td>
<td>5, 12</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>*TIMER</td>
<td>1</td>
<td>7555</td>
<td>6, 1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>QUAD 2-INPUT NAND</td>
<td>1</td>
<td>74L00</td>
<td>4, 14</td>
<td>7</td>
</tr>
<tr>
<td>28</td>
<td>DUAL 4-INPUT NOR</td>
<td>1</td>
<td>7425</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>30</td>
<td>QUAD EXCLUSIVE-OR</td>
<td>1</td>
<td>74L86</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>32</td>
<td>6-BIT COMPARATORS</td>
<td>1</td>
<td>DM6131</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>34</td>
<td>QUAD LATCH</td>
<td>1</td>
<td>74L75</td>
<td>5, 12</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>HEX INVERTER BUFFER O/C</td>
<td>1</td>
<td>7406</td>
<td>14, 7</td>
<td></td>
</tr>
</tbody>
</table>

-C1-C5, C10-C15 .1 MFD CAPACITOR

-C6 .02 MFD CAPACITOR

-C7 .033 OR .039 MFD CAPACITOR

-C8 .01 MFD CAPACITOR

-C9 22 OR 25 MFD CAPACITOR

-C16 2200 PF CAPACITOR

-C17 .01 MFD CAPACITOR

-R1 2.4 KOHM RESISTOR (RED, YELLOW, RED)

-R2 4.7 KOHM RESISTOR (YELLOW, VIOLET, RED)

-R3 1.5 KOHM RESISTOR (BROWN, GREEN, RED)

-R4 330 OHM RESISTOR (ORANGE, ORANGE, BROWN)

-R5 220.0HM RESISTOR 1W (RED, RED, BROWN)

-R6 *27 KOHM RESISTOR (RED, VIOLET, ORANGE)

-R7 *APPROX 10 KOHM RESISTOR (BROWN, BLACK, ORANGE)

-R8 50 KOHM POTentiometer

-R9 100 OHM RESISTOR (BROWN, BLACK, BROWN)

-R10-R17 1 KOHM RESISTOR (BROWN, BLACK, RED)

-CR1 IN914 SIGNAL DIODE

-CR2 IN750 4.7 VOLT ZENER DIODE

-CR3 LIGHT-EMITTING DIODE

-S1 DIP-SWITCH

-J1 DIP-SOCKET

-P1 DIP-PLUG

-2 CASSETTE COAX CABLES

-FLAT RIBBON CABLE

-PC1 PRINTED CIRCUIT BOARD

-CS1 SCOTCH LOW-NOISE CASSETTE

-70-3 INSULATING WAFER

-2 SETS OF 4-40 NUTS, SCREWS, AND WASHERS

*THE ITEMS MARKED WITH AN ASTERISK ARE MATCHED AND BAGGED SEPARATELY.

74LS MAY BE SUBSTITUTED FOR 74L SERIES IC'S IN MOST CASES.*
ASSEMBLY DRAWING

NOTE: IC5 HAS PIN 1 IN UPPER RIGHT, REST IN LOWER LEFT.

COMPONENT SIDE
ASSEMBLY INSTRUCTIONS

1. TAKE OUT ALL THE PARTS AND CHECK THEM AGAINST THE PARTS LIST. IF THERE ARE ANY PARTS MISSING, DROP US A NOTE, AND WE WILL SEND THEM TO YOU. NOTE THAT THE NE555, 10K (APPROX.) RESISTOR, 27K RESISTOR, AND .01 uF CAPACITOR WHICH ARE PACKAGED IN THE SEPARATE BAG ARE MATCHED FOR 1500 Hz AND ARE MARKED WITHASTERISKS ON THE PARTS LIST. OTHER COMPONENTS USED FOR THESE WILL CAUSE THE OUTPUT TO RUN AT A DIFFERENT FREQUENCY.

2. USING THE ASSEMBLY DRAWING ON PAGE 7, INSERT THE DISCRETE COMPONENTS (BAG WITH CAPACITORS, RESISTORS, LIGHT-EMITTING DIODE) INTO THEIR PROPER LOCATIONS. THE RED LEAD GOES TO THE RESISTOR. IF THE RED HAS BEEN RUBBED OFF THE LEAD, USE AN OMINETER TO DETERMINE WHICH LEAD GOES TO THE RESISTOR. THE 1N750 ZENER DIODE (SMALL GLASS DEVICE) SHOULD BE MOUNTED WITH THE LINE SIDE TOWARD THE LEFT. THE 1N914 DIODE (OTHER SMALL GLASS DEVICE) SHOULD BE MOUNTED WITH THE LINE SIDE TOWARD THE BOTTOM. THE ONLY CAPACITOR WHICH SHOULD BE ORIENTED IN A PARTICULAR DIRECTION IS THE FILTER CAPACITOR C9, WHICH SHOULD BE MOUNTED WITH THE PLUS SIDE TOWARD THE LEFT, AS MARKED ON THE BOARD. THE DIP-SOCKET SHOULD BE MOUNTED IN THE J1 POSITION, AS IS SHOWN ON THE ASSEMBLY DRAWING.

3. INSERT ALL THE INTEGRATED CIRCUITS (IC’S) (SMALL BLACK DEVICES AND ONE LARGE METAL DEVICES) INTO THEIR PROPER LOCATIONS AS INDICATED BY THE ASSEMBLY DRAWING AND THE PARTS LIST. NOTE THAT IC #5 (CT20) IS ORIENTED WITH PIN 1 AT UPPER RIGHT; WHEREAS ALL THE OTHERS ARE ORIENTED WITH PIN 1 AT LOWER LEFT. THERE ARE 2 WAYS TO TELL WHICH IS PIN 1 ON AN IC. ON SOME, THERE IS A SMALL DOT BY PIN 1. ON OTHERS, THERE IS A NOTCH AT ONE END OF THE IC, AND PIN 1 IS IN THE LOWER LEFT WHEN THE NOTCH IS AT THE LEFT. BE SURE TO PUT THE MICA INSULATOR UNDER THE REGULATOR (IC 6) BEFORE MOUNTING IT. BE SURE TO PUT IC 6 IN SO THAT THE SCREW HOLES LINE UP, AND THEN INSTALL THE SCREWS, WITH THE WASHERS UNDER THE NUTS. BE SURE THAT THE WASHERS DON’T TOUCH THE TRACE ON SIDE B OF THE PCB BOARD, AND THAT CAPACITOR C9 IS NOT TOUCHING ANY PART OF IC 6.

4. INSTALL THE SEVEN JUMPER WIRES, PREFERABLY USING SMALL INSULATED SINGLE-STRAND COPPER WIRE. THE JUMPER WIRES ARE MARKED ON THE ASSEMBLY DRAWING IN HEAVY BLACK LINES. FIVE OF THEM ARE ALSO MARKED ON THE BOARD WITH MATCHING LETTERS (A, B, C, D, E) BESIDE THEIR HOLES. NOTICE THAT THE BOTTOM OF JUMPER D GOES TO THE HOLE ABOVE THE "D". THERE ARE TWO MORE VERTICAL JUMPERS DOWN NEXT TO THE CONNECTOR. THE LAST IS A SHORT ONE (E), BELOW IC 30, TO SELECT THE OUTPUT PHASE. MINE IS CONNECTED BETWEEN THE CENTER AND LEFT HOLES, AS SHOWN IN THE SCHEMATIC. THIS MAY HAVE TO BE CHANGED LATER, AS NOTED IN THE ADJUSTMENT PROCEDURE.

5. CONSULT THE INSTALLATION NOTES FOR INSTRUCTIONS ABOUT THE CASSETTE CABLES.

6. SEE THE MODIFICATION SHEET (PAGE 8) FOR ANY LATE MODIFICATIONS. REVISION C HAS ALL MODIFICATIONS THROUGH MOD. #6. BE SURE THAT ALL MODIFICATIONS WHICH ARE NOT ON YOUR REVISION ARE INSTALLED.
MODIFICATIONS ON THE TARBELL CASSETTE INTERFACE AS OF SEPT 1, 1976.

INCLUDED ON REVISION B:

1. RESISTOR R1 HAS BEEN CHANGED TO A 2.4 KOHM RESISTOR.
2. ON THE OUTPUT VOLTAGE DIVIDER, R10 IS RECOMMENDED TO BE
   1 KOHM FOR AUXILIARY CASSETTE INPUTS, INSTEAD OF 10 KOHM.

INCLUDED ON REVISION C:

3. A .1 MFD CAPACITOR HAS BEEN ADDED IN PARALLEL WITH R11.
   THIS IMPROVES RELIABILITY WITH SOME TYPES OF RECORDERS.
4. A 1 KOHM RESISTOR HAS BEEN ADDED BETWEEN PINS 10 AND 14
   OF IC 35. THIS PROVIDES PULL-UP TO DRIVE THE 7475 LATCH.
5. THE TRACE TO PIN 8 OF IC 5 (GROUND) HAS BEEN CUT AND A
   LINE RUN DIRECTLY TO THE GROUND BUS ON THE BOTTOM. THIS
   ELIMINATES CROSS-TALK FROM THE NE555 OSCILLATOR.
6. A 2200 PF CAPACITOR SHOULD BE ADDED BETWEEN PINS 6 AND 8 ON
   IC 5. THIS REDUCES THE EFFECTS OF HIGH-FREQUENCY NOISE
   GENERATED IN SOME COMPUTERS.

INCLUDED ON REVISION D:

7. ADD A .01 MFD CAPACITOR BETWEEN PINS 15 AND 8 OF IC 5.
9. CONNECT ONE SIDE OF A 1K RESISTOR TO 5 VOLTS, THE OTHER TO
   IC29-7&10, IC2-14&3&7&10, IC6-14&3&7.
10. CONNECT ONE SIDE OF A 1K RESISTOR TO 5 VOLTS, THE OTHER TO
   IC33-4&10&13.
11. CONNECT ONE SIDE OF A 1K RESISTOR TO 5 VOLTS, THE OTHER TO
    IC4-13&10, IC3-9.
12. CONNECT PIN 1 TO PIN 2 ON IC3.

THERE HAVE BEEN SEVERAL QUESTIONS REGARDING THE USE OF RATES
HIGHER THAN THE STANDARD 187 BYTES PER SECOND. CHANGES ARE
REQUIRED ON BOTH THE INPUT AND OUTPUT SECTIONS. ON THE INPUT
SECTION, THE POTENTIOMETER THAT IS PROVIDED ON THE BOARD MAY
BE ADJUSTED TO CHANGE THE FREQUENCY. ON THE OUTPUT SECTION,
ANY OF THREE COMPONENTS MAY BE CHANGED TO CHANGE THE FREQUENCY:
R6; R7; OR C6. IF YOU WANT TO OPERATE AT TWO FREQUENCIES, FOR
EXAMPLE A HIGHER ONE, AND THE STANDARD, IT IS FEASIBLE TO INSTALL
A SWITCH FOR THE ABOVE MENTIONED COMPONENTS. FOLLOWING WOULD BE
A REASONABLE PROCEDURE FOR EXPERIMENTING WITH THE HIGHER RATES:

1. REDUCE THE VALUE OF R6, R7, OR C6 BY ABOUT THE AMOUNT YOU
   WANT TO INCREASE THE FREQUENCY.
2. USE THE CASSETTE OUTPUT ROUTINE AND A STOP WATCH TO VERIFY
   THAT THE INCREASE IN OUTPUT SPEED HAS BEEN ATTAINED.
3. USE THE SYNC GENERATOR PROGRAM TO MAKE A TAPE WITH A LONG
   STREAM OF SYNC BYTES AT THE NEW FREQUENCY.
4. PLAY IT BACK, ADJUSTING THE POT FOR THE LED TO COME ON.
   CLOSER ADJUSTMENT OF THE CASSETTE VOLUME CONTROL MAY ALSO
   BE NECESSARY.
5. THE .1 CAPACITOR MODIFICATION MENTIONED IN ITEM 3 ABOVE WILL
   ALSO HAVE TO BE REDUCED ACCORDINGLY.
SOLDERING, CLEANING, AND INSTALLATION NOTES

SOLDERING:

Be sure to use good resin-core solder. Acid-core solder will corrode. Use a small soldering element, preferably about 27 watts. Keep your tip clean by wiping on a sponge. Apply heat to the joint first, then solder, then remove solder, then remove the heat (soldering iron). Don't leave the heat applied to the connection more than a few seconds at a time. Some of the components can be destroyed by too much heat, especially the integrated circuits (IC's). Be sure there is a smooth flow of solder over the complete connection, and that the joint looks shiny.

CLEANING:

After you finish soldering, there will be many small conductive particles on the board which you cannot always see. Take a small pointed instrument of some sort, such as a jeweler's screwdriver, and scrape between all printed wiring which is close together, such as those leading to the IC pins. This may take some time, but it is well worth it. Then scrub the bottom (side B) of the board with alcohol. Then visually inspect the board under a strong light, and again remove any dangerous looking particles.

INSTALLATION:

First set the dip-switches to the following positions:

1-off, 2-off, 3-on, 4-off, 5-off, 6-on, 7-off (input phase inversion)

Switches 1 through 6 correspond to address bits 2 through 7 respectively, and off is a one, on is a zero. Address bit 1 can be either way, as it is ignored by the present interface. Address bit 0 is zero for status/control, and one for data. Therefore, the switch settings above correspond to device address 011011XX (most significant bit first), where X indicates bits that can be either way. This is the device select code that is used in all software for the cassette interface that is supplied by Tarbell Electronics.

Then insert the board into the 100-pin socket, being sure that the component side of the board is to the right (Altair) or front (IMSAI) as viewed from the front of the computer. Then install the ribbon cable between the dipconnector on the interface board and the 25-pin connector slot in the rear. Then install the coax cables between the 25-pin connector and the cassette recorder. The coax cables may also be connected directly between the interface board and the recorder. See the pin function list page for the proper connections.
INITIAL ADJUSTMENT INSTRUCTIONS

THIS INTERFACE WAS DESIGNED TO BE AS EASY AS POSSIBLE TO GET UP AND GOING. THERE ARE, HOWEVER, SOME INITIAL ADJUSTMENTS TO BE MADE, AFTER WHICH THERE NEED NEVER BE ANY MORE.

1. PUT THE TEST CASSETTE INTO YOUR CASSETTE RECORDER.

2. IF YOUR RECORDER HAS A TONE CONTROL, TURN IT TO THE MAXIMUM (BEST HIGH-FREQUENCY RESPONSE) POSITION.

3. TURN YOUR VOLUME CONTROL TO A MIDDLE POSITION.

4. TURN THE POTENTIOMETER ON THE INTERFACE TO A MIDDLE POSITION.

5. PRESS THE "PLAY" BUTTON ON YOUR RECORDER.

6. IF THE LED (RED LIGHT) ON THE INTERFACE DOES NOT COME ON AFTER A FEW SECONDS, ADJUST YOUR VOLUME AND THE INTERFACE POTENTIOMETER UNTIL THE LIGHT COMES ON.

7. IF THE LED STILL DOESN'T COME ON, CHANGE SWITCH #7 (INPUT PHASE REVERSAL) ON THE DIP-SWITCH TO THE OPPOSITE POSITION, THEN REPEAT STEP 6.

8. IF THE LED STILL DOESN'T COME ON AFTER ADJUSTING YOUR VOLUME AND THE INTERFACE POT, SOMETHING IS WRONG WITH YOUR RECEIVER SECTION.

9. WHEN THE LED COMES ONE, THIS INDICATES THAT THE RECEIVER IS OPERATING PROPERLY, AND IS DETECTING THE CONTINUOUS STREAM OF SYNC BYTES WHICH IS ON THE TEST TAPE. FURTHER ADJUST BOTH THE VOLUME CONTROL AND THE INTERFACE POT SO THAT YOU CAN TURN EACH OF THEM FROM SIDE TO SIDE A LITTLE WITHOUT THE LIGHT GOING OUT. THE LIGHT SHOULD BE VERY STABLE, WITH NO FLICKER.

NOTE: SINCE THE LED ONLY STAYS ON WHEN IT IS DETECTING CONTINUOUS SYNC BYTES, IN NORMAL OPERATION (WITH REAL DATA) IT WILL JUST FLICKER ONCE IN A WHILE.

1. RUN THE SYNC CODE GENERATOR PROGRAM WITH YOUR RECORDER IN RECORD MODE, ONTO A BLANK TAPE.

2. TRY THE PROCEDURE ABOVE. IF THE LIGHT DOES NOT COME ON CONTINUOUSLY, YOU MAY BE RECORDING AT TOO HIGH OR TOO LOW A LEVEL. TRY DIFFERENT LEVELS UNTIL YOU FIND THE BEST PLACE. YOU ALSO MAY BE RECORDING IN THE OPPOSITE PHASE. IF SO, CHANGE THE JUMPER FROM PIN 9 TO PIN 8 ON IC 23.

NOTE: IF YOU HAVE AN OSCILLOSCOPE, IC 4 PIN 11 SHOULD SHOW A NICE CLEAN WAVE FORM, WITH ABOUT 25% DUTY CYCLE. ALTHOUGH THERE MAY BE LONG-TERM JITTER, BECAUSE OF THE FLUTTER AND WOW ON THE CASSETTE RECORDER, FAST JITTER ON THE EDGES OF THE WAVE-FORM SHOULD BE FAIRLY SMALL. THE LESS THIS HIGH-SPEED JITTER IS, THE MORE TOLERANCE YOUR INTERFACE WILL HAVE TO TAPE SPEED VARIATIONS.
CASSETTE INTERFACE OPERATING INSTRUCTIONS

THESE INSTRUCTIONS PERTAIN TO OPERATING THE INTERFACE WITH AN ORDINARY AUDIO CASSETTE RECORDER, AND ASSUME THAT THE PROPER SOFTWARE (PROGRAMS, SUBROUTINES) IS IN THE COMPUTER TO COMMUNICATE WITH THE INTERFACE.

TO PERFORM AN OUTPUT (SAVE, WRITE) OPERATION:

1. IF YOUR VOLUME CONTROL HAS AN EFFECT ON THE RECORDING FUNCTION, FIRST TURN IT ALL THE WAY DOWN.

2. GET TO THE POINT IN YOUR PROGRAM WHERE ALL IT TAKES IS A PUSH OF A BUTTON TO START INTO THE CASSETTE OUTPUT ROUTINE.

3. USE FAST-FORWARD OR REWIND TO MOVE TO THE DESIRED LOCATION ON THE CASSETTE TAPE.

4. START YOUR CASSETTE RECORDING.

5. IF YOUR VOLUME CONTROL HAS AN EFFECT DURING RECORD, SLOWLY INCREASE THE VOLUME UNTIL YOUR INDICATOR SHOWS A CORRECT RECORDING LEVEL.

6. WAIT FOR ABOUT 5 SECONDS TO RECORD LEADER.

7. PUSH THE BUTTON THAT STARTS THE OUTPUT ROUTINE ON THE RECORDER. (THIS MIGHT BE THE CARRIAGE-RETURN AFTER "SAVE" IN BASIC, OR THE FRONT-PANEL "RUN" BUTTON FOR STAND-ALONE PROGRAMS.)

8. WHEN THE PROGRAM INDICATES THAT THE DATA TRANSFER IS COMPLETE, STOP YOUR CASSETTE RECORDER.

TO PERFORM AN INPUT (LOAD, READ) OPERATION:

1. BE SURE YOUR VOLUME CONTROL IS AT THE POSITION THAT YOU LEFT IT IN THE ADJUSTMENT PROCEDURE.

2. GET TO THE POINT IN YOUR PROGRAM WHERE ALL IT TAKES IS A PUSH OF A BUTTON TO START INTO THE CASSETTE INPUT ROUTINE.

3. USE FAST-FORWARD OR REWIND TO MOVE TO THE DESIRED LOCATION ON THE CASSETTE TAPE. THIS SHOULD BE A FEW SECONDS INTO THE LEADER OF A PREVIOUS RECORDING.

4. START YOUR CASSETTE IN THE PLAYBACK MODE.

5. PRESS THE BUTTON WHICH CAUSES THE INPUT ROUTINE TO START RUNNING. (IN BASIC, THIS MIGHT BE THE CARRIAGE-RETURN AFTER A LOAD, OR THE FRONT-PANEL "RUN" BUTTON FOR STAND-ALONE PROGRAMS, SUCH AS BOOTSTRAPS.)

6. WHEN THE PROGRAM INDICATES THAT THE DATA TRANSFER IS COMPLETE, STOP YOUR CASSETTE RECORDER.

NOTE: ALWAYS BE SURE THAT ALL MEMORY INTO WHICH PROGRAMS OR DATA IS TO BE READ, IS UNPROTECTED FIRST.
CASSETTE BOOTSTRAP PROGRAM

THIS PROGRAM LOADS DATA STARTING AT ZERO AND KEEPS ON GOING. THERE IS NO COUNT OF BYTES, AND NO CHECKSUM. IT IS ASSEMBLED TO RUN AT 2F00 (HEX), BUT MAY BE ASSEMBLED TO RUN ANYWHERE, PROVIDED THAT IT DOES NOT LOAD DATA OVER ITSELF. USE THE SECOND HAND ON YOUR WATCH TO DETERMINE HOW LONG TO WAIT UNTIL STOPPING THIS PROGRAM. ALLOW ABOUT 45 SECONDS TO LOAD AN 8 KBYTE BLOCK.

2F00 3E 10  MVI A, 10H  SET BIT 4 OF A = 1.
2F02 D3 6E  OUT CASC  RESET INTERFACE.
2F04 21 00 00  LXI H, 0  PUT STARTING ADDRESS IN H, L.
2F07 DB 6E  LOOP IN CASC  READ STATUS.
2F09 E6 10  ANI 10H  CLEAR ALL BUT BIT 4.
2F0B C2 07 2F  JNZ LOOP  WAIT IN LOOP UNTIL READY.
2F0E DB 6F  IN CASD  READ A DATA BYTE.
2F10 FB  EI  SIGNAL OPERATOR.
2F11 77  MOV M, A  PUT DATA INTO MEMORY.
2F12 23  INX H  INCREMENT MEMORY POINTER.
2F13 C3 07 2F  JMP LOOP  REPEAT THE ABOVE OPERATION.

CASC EQU 6EH  CASSETTE STATUS PORT.
CASD EQU 6FH  CASSETTE DATA PORT.
END

NOTE: IF YOU HAVE AN IMSAI OR ALTAIR WITH AN OUTPUT PORT ON THE FRONT PANEL (8 LED'S), YOU CAN USE THE BOOTSTRAP PROGRAM FOR TROUBLESHOOTING THE INPUT SECTION WITH THE FOLLOWING MODIFICATION: AT INSTEAD OF SUBSTITUTE
2F10 EI (FB)  CMA (2F)
2F11 MOV M, A (77) OUT (D3)
2F12 INX H (23) LEDS (FF)

SYNC CODE GENERATOR PROGRAM

THIS PROGRAM MAY BE USED TO GENERATE A CONTINUOUS STREAM OF E6 (HEX), THE SAME AS IS ON THE SUPPLIED CASSETTE. IF YOU FIND THAT THE RECEIVED STREAM IS INVERTED FROM THE ONE SUPPLIED, YOU MAY CHANGE THE CIRCUIT SO THAT IC23-8 IS HOOKED TO THE 74L86 INSTEAD OF IC23-9. THIS WILL MAKE YOUR RECORDINGS THE SAME PHASE AS MINE.

0000 DB 6E  LOOP IN CASC  READ STATUS.
0002 E6 20  ANI 20H  LOOK AT BIT 5.
0004 C2 00 00  JNZ LOOP  WAIT UNTIL READY.
0007 3E E6  MVI A, 0E6H  GET sync BYE.
0009 D3 6F  OUT CASD  WRITE IT ONTO CASSETTE.
000B C3 00 00  JMP LOOP  REPEAT.
CASC EQU 6EH  STATUS PORT.
CASD EQU 6FH  DATA PORT.
END

INPUT ADDR FROM CAS
OUTPUT ADDR TO CAS
SYNC BIT
CASSETTE INTERFACE OUTPUT ROUTINE

THIS PROGRAM WRITES A BLOCK OF MEMORY OUT ONTO CASSETTE TAPE. THE PROGRAM IS ASSEMBLED TO START AT 3100 (HEX), BUT MAY BE REASSEMBLED TO START ANYWHERE. THE BLOCK STARTING ADDRESS IS LOCATED AT ADDRESS 3104 (HEX). THE BLOCK LENGTH (2 BYTES) IS LOCATED AT ADDRESS 3107 (HEX). THE PROGRAM WILL WRITE A "W" ON THE COMMENT DEVICE WHEN IT IS THROUGH WITH IT'S DATA TRANSFER.

```
3100 31 43 31 LXI SP,STAK SET STACK POINTER.
3103 21 00 00 LXI H,0 GET BLOCK ADDRESS.
3106 01 00 20 LXI B,2000H SET BLOCK LENGTH = 6192.
3109 1E 00 MVI E,0 SET E=0.
310B 3E 3C MVI A,3CH GET START BYTE.
310D CD 32 31 CALL COUT OUTPUT START BYTE TO CASSETTE.
3110 3E E6 MVI A,0E6H GET SYNC BYTE.
3112 CD 32 31 CALL COUT OUTPUT SYNC BYTE TO CASSETTE.
3115 7E LOOP MOV A,M GET A DATA BYTE FROM MEMORY.
3116 CD 32 31 CALL COUT OUTPUT DATA BYTE TO CASSETTE.
3119 E3 ADD E ADD E (CHECKSUM) TO A.
311A 5F MOV E,A PUT NEW CHECKSUM INTO E.
311B 23 INX H INCREMENT MEMORY POINTER.
311C 0B DCX B DECREMENT COUNTER.
311D 3E 00 MVI A,0 MAKE A=0.
311F B8 CNP B IF B NOT = 0,
3120 C2 15 31 JNZ LOOP REPEAT LOOP.
3123 B9 CNP C IF C NOT = 0,
3124 C2 15 31 JNZ LOOP REPEAT LOOP.
3127 7B MOV A,E OTHERWISE, GET CHECKSUM
3128 CD 32 31 CALL COUT AND OUTPUT IT.
312B 3E 57 MVI A,"W" WRITE "W" (END OF WRITE).
312D D3 01 OUT 1 PRINT ON CONSOLE.
312F C3 2F 31 WAIT JMP WAIT WAIT HERE WHEN DONE.
3132 F5 COUT PUSH PSW SAVE A AND FLAGS.
3133 DB 6E CLOP IN CASC READ CASSETTE STATUS.
3135 E6 20 ANI 20H CLEAR ALL BUT BIT 5.
3137 C2 33 31 JNZ CLOP TRY AGAIN IF NOT READY.
313A F1 POP PSW RESTORE A AND FLAGS.
313B D3 6F OUT CASD OUTPUT DATA TO CASSETTE.
313D C9 RET RETURN FROM COUT.
313E 00 0
313F 00 0
3140 00 0
3141 00 0
3142 00 0
3143 00 STAK 0
3144 00 CASD EQU 6FH
```
Cassette Interface Input Routine

This program reads a block of bytes from cassette into memory. The program is assembled to start at 3100 (HEX), but may be reassembled to start anywhere, although care should be taken to insure that the data it is reading does not write over the program itself. This may be accomplished by locating the program immediately below or a block length above the data to be read in. The starting address for the block is located in address 3185 (HEX). The block length is located in address 3188 (HEX) (two bytes).

```
3180 3E 10 MVI A,10H SET BIT 4 OF A=1.
3182 D3 6E OUT CASC RESET INTERFACE.
3184 21 00 00 LXI H,0 GET STARTING ADDRESS.
3187 11 00 20 LXI D,2000H GET BLOCK LENGTH.
318A 06 00 MVI B,0 SET CHECKSUM = 0.
318C D3 6E LOOP IN CASC READ CASSETTE STATUS.
318E E6 10 ANI 1OH LOOK AT BIT 4.
3190 C2 EC 31 JNZ LOOP WAIT IF NOT READY.
3193 DB 6F IN CASD READ DATA FROM CASSETTE.
3195 77 MOV M,A PUT DATA INTO MEMORY.
3197 80 ADD B ADD CHECKSUM TO A.
3199 47 MOV B,A PUT IT BACK IN B.
319A 23 INX H INCREMENT MEMORY POINTER.
319C B8 DCX D DECREMENT COUNTER.
319C BA CMP D IF D NOT = 0.
319D C2 EC 31 JNZ LOOP READ MORE.
31A0 BB CMP E IF E NOT = 0.
31A1 C2 EC 31 JNZ LOOP READ MORE.
31A4 DB 6E CHEK IN CASC READ STATUS.
31A6 E6 10 ANI 1OH LOOK AT BIT 4.
31A8 C2 A4 31 JNZ CHEK WAIT IF NOT READY.
31AB DB 6F IN CASD READ CHECKSUM.
31AD B8 CMP B COMPARE TO B.
31AE 3E 45 MVI A,"E" PUT CODE FOR "E" IN A.
31B0 C2 B5 31 JNZ ERR IF CHECKSUMS NOT EQUAL, ERROR.
31B3 C6 02 ADI 2 ADD A 2 TO MAKE "G" IF EQUAL.
31B5 D3 01 ERR OUT CRTD PRINT "E" FOR "G".
31B7 C3 B7 31 END JMP END WAIT HERE WHEN DONE.
CASC EQU 6EH CASSETTE STATUS/CONTROL PORT.
CASD EQU 6FH CASSETTE DATA PORT.
CRTD EQU 01H CONSOLE DATA PORT.
```
I REM THIS PROGRAM SHOWS HOW TO SAVE DATA INTO A CASSETTE
2 REM AND LOAD IT BACK INTO MEMORY FROM A PROGRAM RUNNING
3 REM UNDER 8K BASIC 3.1. IT ALLOWS YOU TO ENTER LINES OF
4 REM TEXT FROM THE CONSOLE KEYBOARD, SAVE THEM ON CASSETTE.
5 REM LOAD THEM BACK INTO MEMORY, AND PRINT THEM.
10 CLEAR 3000: L=50: DIM A$(50) REM RESERVE FOR UP TO 50 LINES.
20 CC=110: CD=111: REM CASSETTE CONTROL AND DATA PORT NUMBERS.
30 TS=%*: REM END-OF-FILE CHARACTER.
40 D=100
100 INPUT "COMMAND"; CS
110 IF CS="ENTER" THEN 1000
120 IF CS="PRINT" THEN 2000
130 IF CS="SAVE" THEN 3000
140 IF CS="GET" THEN 4000
900 PRINT "INVALID COMMAND." : GOTO 100
1000 REM ENTER TEXT FROM THE CONSOLE KEYBOARD
1020 FOR N=1 TO L: REM ENTER A MAXIMUM OF L LINES.
1030 INPUT BS: REM READ A LINE FROM KEYBOARD.
1040 IF BS="%" THEN 1070: REM A % TERMINATES THE INPUT.
1050 A$(N)=BS: NEXT N: REM PUT LINE INTO BUFFER.
1070 N=N-1: GOTO 100: REM N=THE NUMBER OF LINES ENTERED.
2000 REM PRINT THE BUFFER AREA ON THE CONSOLE.
2010 FOR I=1 TO N: PRINT A$(I): NEXT I: GOTO 100
3000 REM SAVE THE BUFFER INTO CASSETTE TAPE.
3010 SS=CHRS(195)+CHRS(230): REM SS=START & SYNC BYTES.
3020 POK 123041 CC: POK 123241 CD: POK 123841 255
3030 FOR I=1 TO N
3040 F0R K=1 TO D: NEXT K: REM DELAY FOR COUNT OF D.
3050 BS=SS+A$(I): REM HOOK START & SYNC BYTES TO LINE.
3060 PRINT BS: REM WRITE LINE INTO CASSETTE.
3070 NEXT I
3080 BS=SS+TS: REM HOOK START & SYNC BYTES TO TERMINATOR.
3090 F0R K=1 TO D: NEXT K: REM DELAY FOR COUNT OF D.
3100 PRINT BS: REM WRITE THE END-OF-FILE MARK.
3110 REM CHANGE CONSOLE ROUTINE BACK TO NORMAL.
3120 POK 123041 16: POK 123241 128: POK 123841 1
3130 G0T0 100
4000 REM GET TEXT FROM CASSETTE AND PUT INTO BUFFER.
4010 REM CHANGE CONSOLE INPUT ROUTINE FOR CASSETTE.
4011 POK 124141 CC: POK 124341 16: POK 124841 CD: POK 123241 0
4012 POK 123841 255
4020 F0R I=1 TO L
4030 OUT CC, 16: REM RESET CASSETTE INPUT SECTION.
4040 INPUT BS: REM READ A LINE OF TEXT FROM CASSETTE.
4050 IF BS=TS THEN 4080
4060 A$(I)=BS
4070 NEXT I
4080 N=N-1
4090 REM CHANGE CONSOLE ROUTINE BACK TO NORMAL.
4091 POK 124141 0: POK 124341 1: POK 124841 0: POK 123241 0
4092 POK 123841 1
4100 G0T0 100
OK
SAVING AND LOADING DATA TO AND FROM CASSETTE FROM BASIC PROGRAMS

It is quite often desirable to save and load data to and from cassette, while running a BASIC program. For example, you might have a nice inventory program running in BASIC, but it's of limited utility if there is no way to save the inventory on cassette for overnight storage. You can leave the computer running, but if there is an interruption in power, or a computer failure, your inventory is lost. Other applications include accounts receivable, mailing lists, and payroll.

There are several ways of handling this function, depending on the version of BASIC you use. Some versions of BASIC have a command which can save and load a numerical array to and from cassette. ALTAIR® DISK BASIC is one of these. The only problem with this way, is that DISK BASIC is fairly large, and that strings have to be converted to numerical arrays. Another way is described explicitly by a program on page 75 of the Tarbell Cassette Interface Manual. In this method, the console (TTY, CRT) routines are modified by POKE commands, so that they are temporarily cassette I/O routines. Then the PRINT and INPUT statements may be used to transfer the data. After the data is transferred, POKE statements restore the console routines to their original form. The disadvantage here is that different versions of BASIC have their console I/O routines in different places, so the program has to be adapted when changing from one version to another.

Another possibility is to POKE the data into an unused area of memory. The USR function then is used to transfer to your own output routine (possibly an adaptation from one in the manual). This routine only needs to transfer a block of memory onto tape. The USR function can be used with a different argument to run an input routine, then the BASIC program can retrieve the data with the PEEK function. The main problem with this method is that the I/O routines generally have to be loaded separately, and are not yet written and tested.

Still another way is to use a general purpose monitor program. This program would also be loaded separately, but might reside in read-only-memory (ROM). The monitor would handle all input/output functions including console and cassette I/O. One of the commands in the monitor is to assign different functions to different I/O devices than normally handle these functions. For example, the console function could be assigned to the cassette interface instead of the TTY. Then PRINT and INPUT statements could be used for I/O to and from cassette.

The monitor described above is presently under development by Tarbell Electronics. In addition to the Assign command, it has commands for dumping and loading memory, checking records for errors, and moving data from one area in memory to another.
CONTROLLING THE START-STOP (REMOTE) FUNCTION ON YOUR CASSETTE

Most cassette recorders have a remote control input, meant for control from a microphone switch. This facility can be operated by a computer program to start and stop the tape automatically, according to the needs of the program. This is particularly important if the amount of data on tape is more than will fit into main memory all at once. In this case, the data may be "blocked"; that is, gaps may be inserted between blocks of data, which allow time for the tape to start and stop. The program may then start the tape, read some data, stop the tape, work on the data, start another tape, write some data, and stop the tape. This process may be repeated until all the data is processed. Some examples where this operation might be necessary are as follows:

1) An assembler, where the source is larger than memory, may read source from one tape and write machine code to another.
2) A compiler, in the same situation.
3) A merging program, where an old file is updated with changes to form a new file.

The circuit for controlling one tape unit is shown below:

![Circuit Diagram]

These routines may be used for starting and stopping the cassette before and after input and output operations:

START
LDA CTLS GET CONTROL STATUS BYTE.
ORI 01 SET Bit 0 = ONE.
STA CTLS UPDATE CONTROL STATUS BYTE.
OUT CASC START THE TAPE.
CALL DELAY WAIT FOR TAPE TO GET UP TO SPEED.
RET RETURN (NEXT DO YOUR I/O).

STOP
LDA CTLS GET CONTROL STATUS BYTE.
AND OFEH SETBit 0 TO ZERO.
STA CTLS UPDATE CONTROL STATUS BYTE.
OUT CASC STOP THE TAPE.
RET RETURN FROM I/O ROUTINE.

CTLS DB 0 CONTROL STATUS BYTE.
CASC EQU 6EH CASSETTE CONTROL PORT.

The start-up delay is determined by your recorder, and should be longer before a write than before a read operation.

Note: A module which allows the control of up to four cassette recorders with a Tarbell cassette interface is available from: Ro-Che Systems, 7101 Mammoth Ave, Van Nuys, CA 91405.
THEORY OF CASSETTE INTERFACE OPERATION

OUTPUT SECTION

THE PURPOSE OF THIS SECTION IS TO CONVERT 8-BIT PARALLEL BYTES FROM THE COMPUTER TO A SERIAL BI-PHASE ENCODED DATA STREAM FOR THE RECORDER. THE NE555 (IC 22) IS CONNECTED AS AN OSCILLATOR TO OSCILLATE AT TWICE THE FREQUENCY OF THE REQUIRED CLOCK RATE. FOR 800 BITS PER INCH, THE CLOCK RATE NEEDS TO BE 1500 HZ FOR A RECORDER RUNNING AT 1 7/8 INCHES PER SECOND. SO IC 22 RUNS AT 3000 HZ, DETERMINED BY THE MATCHED SET OF COMPONENTS (IC 22, C8, R6, AND R7). THE OUTPUT ON PIN 3 IS FED TO A J-K FLIP-FLOP, WHICH DIVIDES THE FREQUENCY BY TWO. THE MAIN FUNCTION OF THIS FLIP-FLOP IS TO MAKE THE WAVE-FORM PERFECTLY SYMMETRICAL. THIS SQUARE-WAVE IS THEN FED TO AN INPUT OF THE EXCLUSIVE-OR GATE AT IC 30 PIN 6.

IC 32, THE DM8131, IS A 6-BIT DIGITAL COMPARATOR. IT'S PURPOSE IS TO COMPARE THE ADDRESS ON BITS 2 THROUGH 7 OF THE ADDRESS BUS WITH THE SETTING ON THE DIP SWITCH. WHEN THEY MATCH, THE OUTPUT AT PIN 9 GOES LOW, INDICATING THAT THIS DEVICE IS BEING SELECTED. THIS SIGNAL IS INVERTED AND ANDED WITH THE WRITE SIGNAL AND THE STATUS OUTPUT SIGNAL ON THE BUS. THIS PRODUCES A HIGH OUTPUT AT IC 14 PIN 6 WHEN THIS DEVICE IS BEING WRITTEN TO.

THIS SIGNAL IS IN TURN ANDED WITH ADDRESS BIT 0 TO STROBE DATA INTO THE SHIFT REGISTER (IC 26 PIN 6 AND IC 20 PINS 4 AND 2). THE DATA IS SHIFTED OUT OF THE SHIFT REGISTER BY THE CLOCK PREVIOUSLY MENTIONED. THE SHIFT REGISTER IS MADE UP OF IC'S 21, 27, AND 23. DATA ENTERS AT THE BOTTOM AND IS SHIFTED OUT TOWARD THE RIGHT, WHERE IT IS COMBINED WITH THE CLOCK TO FORM THE BI-PHASE SIGNAL FOR THE RECORDER.

THE FIRST EIGHT OUTPUTS OF THE SHIFT REGISTER ARE FED INTO IC 28. SINCE A ZERO IS BEING FED INTO THE LEFT END OF THE SHIFT REGISTER FOR EACH SHIFT (IC 21 PIN 9), THESE LEFT EIGHT BITS WILL ALL BECOME ZERO AFTER 8 SHIFTS. AT THIS TIME THE OUTPUTS AT IC 28 PINS 6 AND 8 WILL BOTH BE HIGH, CAUSING THE OUTPUT AT IC 14 PIN 8 TO GO HIGH. THIS SIGNAL WILL BE USED BY THE COMPUTER TO DETERMINE WHEN THE INTERFACE NEEDS MORE OUTPUT DATA.

WHEN ADDRESS BIT 0 IS LOW, THE OUTPUT AT IC 14 PIN 6 ANDED WITH IC 31 PIN 2 CAUSES A HIGH OUTPUT AT IC 35 PIN 10. THIS STROBES BITS 0, 1, 2, AND 3 ON THE DATA BUS INTO THE LATCH OF IC 34. THE OUTPUTS OF THIS LATCH ARE BUFFERED WITH IC 35, AND APPEAR AT PINS 2, 4, 6, AND 8. THESE ARE THEN FED TO THE DIP-SOCKET. IF DATA BUS BIT 4 IS HIGH, A PULSE APPEARS AT PIN 8 OF IC 26. THIS IS USED TO RESET THE COUNTER IN THE INPUT SECTION.
THEORY OF OPERATION

INPUT SECTION

THE PURPOSE OF THIS SECTION IS TO CONVERT THE BI-PHASE AUDIO SIGNAL COMING FROM THE CASSETTE RECORDER TO 8-BIT PARALLEL BYTES FOR THE COMPUTER. THE AUDIO SIGNAL COMING DIRECTLY FROM THE CASSETTE EARPHONE OR SPEAKER OUTPUT IS TERMINATED BY THE 100 OHM RESISTOR AND FED THROUGH THE .02 CAPACITOR TO THE INPUT OF THE 8T20 (IC 5 PINS 6 AND 7). THIS INPUT HAS A BUILT-IN VOLTAGE-DIVIDER, WHICH BIASES THE DC LEVEL TO A GOOD MIDWAY TTL REFERENCE VOLTAGE BETWEEN ONE AND ZERO. THIS IS ONE INPUT OF A HIGH-SPEED COMPARATOR. THE OTHER INPUT (IC 5 PIN 5) IS CONNECTED THROUGH A RESISTOR DIVIDER AND A 1N914 DIODE TO ONE OF THE COMPARATOR OUTPUTS (IC 5 PIN 9). THIS PROVIDES A SMALL AMOUNT OF HYSTERESIS TO COMBAT NOISE PROBLEMS. THE OTHER OUTPUT OF THE COMPARATOR (IC 5 PIN 1) IS EXCLUSIVE-ORED WITH SWITCH 7 ON THE DIP-SWITCH TO PROVIDE A WAY TO INVERT THE INPUT DATA STREAM. THIS IS THEN FED TO A D-TYPE FLIP-FLOP (IC 4 PIN 12).

MEANWHILE, INSIDE IC 5, THE OUTPUTS OF THE COMPARATOR SECTION ARE FED TO AN EDGE-DETECTOR, WHICH DETECTS BOTH POSITIVE AND NEGATIVE-GOING TRANSITIONS. THE OUTPUT OF THIS DETECTOR IS THEN USED TO TRIGGER A STABLE NON-RETRIGGERABLE ONE-SHOT. THE CAPACITOR FOR THIS ONE-SHOT IS BETWEEN PINS 12 AND 14 OF IC 5, AND THE RESISTOR IS THE 50 KOhM POTENTIOMETER. THE OUTPUT OF THE ONE-SHOT IS THE RECOVERED CLOCK (IC 5 PINS 10 AND 11). THIS IS FED TO THREE DIFFERENT PLACES: 1) IC 4 PIN 11, WHERE IT TRIGGERS THE FLIP-FLOP TO RECOVER THE SERIAL DATA STREAM, 2) IC 3 PIN 8 WHERE IT IS USED TO SHIFT THE SERIAL-PARALLEL SHIFT-REGISTER, 3) IC 29 PIN 1 AND IC 26 PIN 13, WHERE IT IS USED TO STEP THE 8 COUNTER.

IN A START-UP POSITION, IC 29 PIN 12 IS ZERO, HAVING BEEN RESET EITHER BY THE RESET SWITCH OR BY A RESET COMMAND FROM THE PROGRAM. THIS STOPS THE CLOCK FROM TRIGGERING THE 8 COUNTER, WHICH HAS ALSO BEEN RESET. AS THE SERIAL STREAM FLOWS THROUGH THE SHIFT-REGISTER (IC 3), IT IS CONTINUALLY INSPECTED BY THE SYNC DECODER MADE UP OF IC'S 9 AND 10. WHENEVER A SYNC CODE APPEARS IN THE SHIFT REGISTER, IC 10 PIN 8 GOES LOW. THIS LIGHTS THE SYNC LED AND ALSO ALLOWS THE FLIP-FLOP AT IC 29 PIN 12 TO GO HIGH. THIS ALLOWS THE CLOCK TO APPEAR AT IC 26 PIN 11, AND TRIGGER THE 8 COUNTER. WHEN THE COUNTER HAS COUNTED TO 8, IC 8 PIN 12 GOES HIGH, TRIGGERING THE READY FLIP-FLOP AT IC 8 PIN 9. THIS READY CONDITION INDICATES TO THE COMPUTER THAT THERE IS A BYTE IN THE SHIFT REGISTER READY TO READ. THE COMPUTER MAY THEN READ THIS BYTE THROUGH GATES OF IC'S 13 AND 19.

IN ORDER TO READ DATA THE SIGNAL AT POINT B, WHICH COMES FROM THE OUTPUT SECTION IS ANDED WITH THE TWO INPUT GATE SIGNALS PDBIN AND SINP (IC 14 PINS 13, 1 AND 2). WHEN ADDRESS BIT 0 IS HIGH, THIS SIGNAL IS ANDED WITH IT TO GATE DATA FROM THE SHIFT REGISTER ONTO THE INPUT DATA BUS (IC 1 PIN 3). WHEN ADDRESS BIT 0 IS LOW, THIS SIGNAL IS ANDED WITH IT TO GATE VARIOUS STATUS BITS ONTO THE INPUT DATA BUS. FOUR OF THESE INPUTS ARE GENERAL-PURPOSE, AND COME FROM THE DIP-SOCKET. THE OTHERS ARE PICKED UP AT IC 25 PIN 3 (INPUT STATUS), AND IC 25 PIN 6 (OUTPUT STATUS).
CASSETTE INTERFACE TIMING DIAGRAM

SERIAL BINARY DATA | 1 1 1 0 0 1 1 0 1 1 1 0 0 1 1 |
CLOCK (PIN 2-9) |
NRZ DATA (PIN 23-9) |
BI-PHASE TO TAPE (30-4) |
AUDIO FROM TAPE |
RECOVERED BI-PHASE (30-11) |
EDGE DETECTOR OUTPUT |
RECOVERED CLOCK (5-11) |
RECOVERED NRZ DATA (4-9) [INVALID HERE] |

* CLOCK IS IN PROPER PHASE AFTER FIRST DATA CHANGE.
CASSETTE INTERFACE PIN FUNCTIONS

J1 (DIP SOCKET)

1. GP STATUS IN BIT 0   16. GROUND
2. GP STATUS IN BIT 1   15. DATA FROM CASSETTE
3. GP STATUS IN BIT 2   14. SPARE
4. GP STATUS IN BIT 3   13. SPARE
5. GP CONTROL OUT BIT 3 12. +5 VOLTS
6. GP CONTROL OUT BIT 2 11. SPARE
7. GP CONTROL OUT BIT 1 10. DATA TO CASSETTE
8. GP CONTROL OUT BIT 0  9. GROUND

NOTES: GP STANDS FOR GENERAL-PURPOSE
DATA FROM CASSETTE SHOULD BE CONNECTED TO EARPHONE JACK
DATA TO CASSETTE SHOULD BE CONNECTED TO AUXILIARY JACK.
IF MIKE JACK IS USED, OR IF THE RECORDER ONLY OPERATES WITH
AUTOMATIC VOLUME CONTROL ON RECORD, THEN DIVIDER RESISTORS
MAY HAVE TO BE CHANGED FOR THE PROPER RECORDING LEVEL.
The Divider resistors are R10 and R11.

P2 (8800 BUS PINS)

1.  +8V
29. ADDRESS LINE #5
30. ADDRESS LINE #4
31. ADDRESS LINE #3
35. DATA OUT LINE #1
36. DATA OUT LINE #0
38. DATA OUT LINE #4
39. DATA OUT LINE #5
40. DATA OUT LINE #6
41. DATA IN LINE #2
42. DATA IN LINE #3
43. DATA IN LINE #7
45. OUT
46. INP
50. GROUND
51.  +8V
52. -16V
75. RESET-NOT
77. WRITE-NOT
78. DATA BUS IN
79. ADDRESS LINE #0
81. ADDRESS LINE #2
82. ADDRESS LINE #6
83. ADDRESS LINE #7
88. DATA OUT LINE #2
89. DATA OUT LINE #3
90. DATA OUT LINE #7
91. DATA IN LINE #4
92. DATA IN LINE #5
93. DATA IN LINE #6
94. DATA IN LINE #1
95. DATA IN LINE #0
100. GROUND
IF YOU CANNOT MAKE AT LEAST TEN 8K-BYTE TRANSFERS WITH NO ERRORS, YOU HAVE A PROBLEM, AND THE ITEMS BELOW MAY BE OF SOME HELP:

1. CHECK TO MAKE SURE THAT ALL THE COMPONENTS AND JUMPERS ARE IN THEIR PROPER LOCATIONS, AND THAT THEY ARE ORIENTED AS SHOWN IN THE ASSEMBLY DRAWING.

2. MAKE SURE THAT THE BOARD IS CLEAN, ESPECIALLY THAT THERE IS NO FLUX RESIDUE BETWEEN IC PINS OR OTHER CLOSE LINES.

3. DEMAGNETIZE AND CLEAN THE RECORD/PLAYBACK HEAD ON YOUR RECORDER.

4. WHEN YOU PLUG IN THE BOARD, BE SURE THAT THE PINS ON THE BOARD EDGE CONNECTOR LINE UP WITH THE PINS IN THE MOTHERBOARD CONNECTOR.

5. HAVE YOU TRIED BOTH PHASES WITH THE PHASE SWITCH, AND ARE THE OTHER SETTINGS ON THE DIP-SWITCH CORRECT?

6. YOU SHOULD BE ABLE TO ADJUST THE VOLUME ON YOUR RECORDER BY ABOUT 50% DURING PLAYBACK, AND STILL HAVE THE SYNC LIGHT LIT WHEN READING THE SYNC STREAM. IF IT DOESN'T LIGHT AT ALL, THERE IS PROBABLY SOME GROSS PROBLEM ON THE BOARD, SUCH AS A BAD PLATED THROUGH HOLE, A SOLDER BRIDGE, OR A BAD INTEGRATED CIRCUIT. IF YOU ARE USING A TAPE DECK THAT HAS ONLY A PREAMP, YOU MAY NEED TO ADD AN EXTRA STAGE OF AMPLIFICATION IN ONE OF THE EXTRA IC SLOTS.

7. IF YOU HAVE AN OSCILLOSCOPE, THE BEST PLACE TO LOOK TO SEE HOW THE RECEIVER INPUT SECTION IS OPERATING IS AT IC 4, PIN 11. THIS SIGNAL SHOULD BE FAIRLY CLEAN, WITH SOME OVERALL JITTER, DUE TO THE TAPE WOW AND FLUTTER, AND SOME HIGH-SPEED JITTER ON THE EDGE OF THE WAVEFORM. IT IS THIS HIGH-SPEED JITTER THAT YOU SHOULD TRY TO MAKE A MINIMUM.

8. IF YOU HAVE A VIDEO INTERFACE, OR OTHER SOURCE OF HIGH-FREQUENCY NOISE, TRY LOCATING IT FURTHER AWAY FROM THE CASSETTE INTERFACE.

9. ARE YOU SURE THAT YOUR RECORDER HAS A FREQUENCY RESPONSE TO 8KHZ?

10. HAVE YOU USED THE PROPER VOLTAGE DIVIDER (R10, R11) FOR YOUR PARTICULAR RECORDER? IF YOU ARE ABLE TO RECOVER THE SYNC STREAM I WROTE SATISFACTORILY, BUT ARE HAVING TROUBLE WITH RECORDINGS YOU MAKE YOURSELF, THE LEVEL GOING FROM THE INTERFACE TO THE RECORDER MAY BE TOO HIGH OR TOO LOW, ESPECIALLY IF YOU HAVE AUTOMATIC LEVEL CONTROL. YOU MAY ALSO WANT TO TRY OPERATING WITHOUT C15.

11. HAVE YOU CHECKED YOUR +5 VOLT POWER? TOO MANY BOARDS IN YOUR COMPUTER COULD INTRODUCE RIPPLE ON THIS SUPPLY.

12. DON'T USE DIGITALLY CERTIFIED TAPE, ONLY AUDIO LOW-NOISE TAPE.

13. IF YOUR RECORDER HAS AN INTERNAL MICROPHONE, BE SURE IT IS NOT ACTIVE WHILE YOU ARE MAKING A RECORDING (THE J.C. PENNY HAS A SWITCH ON THE TONE CONTROL WHICH CUTS OFF THE INTERNAL MIC).

14. IF YOU STILL HAVE PROBLEMS, PLEASE RETURN THE UNIT, PREFERABLY WITH YOUR CASSETTE RECORDER, AND I WILL GET IT OPERATING PERFECTLY WITHOUT CHARGE. THE REPAIR TURNAROUND TIME IS 1 TO 3 WEEKS.

15. IF YOU ARE COMPLETELY DISSATISFIED, YOU MAY RETURN THE INTERFACE FOR A COMPLETE REFUND WITHIN 90 DAYS AFTER YOU ACCEPTED DELIVERY.
IDEAS FOR USING THE CASSETTE INTERFACE

SAVING AND LOADING PROGRAMS

Programs may be toggled into memory, loaded from paper tape, or loaded by some other means. They may then be dumped onto cassette using the cassette output routine. The output routine itself may be dumped along with the other program, so that it will be available later for further dumps. Routines may then be loaded from cassette, modified, and dumped back out to cassette in a continuous process of development.

USING A BACKUP

A backup is a method of making sure that valuable programs or data is not lost. One simple way of providing a backup is to record a particular program in two different places on the same tape, or on two different tapes. You may want to go back and forth between the two copies, each time the program is changed. In this way, you always have a copy of the last program, and only the most recent changes are lost if a power failure or other equipment trouble develops during the process of saving.

STARTING AND STOPPING AUTOMATICALLY

Most audio cassette recorders have an input labeled "remote". This is normally operated from a switch on the microphone, so that the recorder can be started and stopped while dictating. This input can be used to start and stop the recorder under program control from the computer. One of the 4 extra control lines coming from the cassette interface (DIP-socket pins 5, 6, 7, or 8) can be used to drive a relay which would have its contacts connected to the recorder remote jack. The main requirements are that the 40 mA available from the control line be able to drive the relay, and that the relay contacts are able to handle the current into the remote jack, which can be as high as one ampere.

OPERATING WITH MORE THAN ONE CASSETTE RECORDER

Sometimes it is desirable to operate with two or more cassette units. With the Tarbell cassette interface, it is already possible to read from one cassette recorder, while writing onto another. This is because the input and output sections are entirely independent, and may be programmed separately. If it is desired to read from one of two units, and write onto another, such as during a merging operation, relays may be used to switch back and forth between the two input units, under control of one of the 4 control lines. If, however, it is necessary to read simultaneously from two different units, or write different information onto two units at the same time, it will be necessary to have two cassette interfaces.
### MODIFICATIONS ON CASSETTE BASIC FOR CSAVE AND CLOAD USING TARBEll CASSETTE INTERFACE

#### 8K BASIC 3.2

<table>
<thead>
<tr>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1008 06</td>
<td>EE 6E</td>
<td>155A 06</td>
<td>EE 6E</td>
<td>0E5E 06</td>
<td>EA</td>
<td>1DFE DB</td>
<td>DB</td>
</tr>
<tr>
<td>1009 06</td>
<td>EE 6E</td>
<td>155B 01</td>
<td>EE 10</td>
<td>0E59 D2</td>
<td>67</td>
<td>1DFF 06</td>
<td>EE 10</td>
</tr>
<tr>
<td>100A 01</td>
<td>10</td>
<td>155C 01</td>
<td>10</td>
<td>0E5A FA</td>
<td>0E</td>
<td>1DO0 E6</td>
<td>0E</td>
</tr>
<tr>
<td>100B 07</td>
<td>C2</td>
<td>155D 07</td>
<td>C2</td>
<td>0E5B 67</td>
<td>3E</td>
<td>1DO1 E0</td>
<td>20</td>
</tr>
<tr>
<td>100C 07</td>
<td>07</td>
<td>155E 59</td>
<td>59</td>
<td>0E5C 0E</td>
<td>19</td>
<td>1DO2 C2</td>
<td>C2</td>
</tr>
<tr>
<td>100D 10</td>
<td>10</td>
<td>155F 15</td>
<td>15</td>
<td>0E5D 76</td>
<td>CD</td>
<td>1DO3 FE</td>
<td>FE</td>
</tr>
<tr>
<td>100E 07</td>
<td>DB</td>
<td>1560 07</td>
<td>DB</td>
<td>0E5E CD</td>
<td>FD</td>
<td>1DO4 1D</td>
<td>1D</td>
</tr>
<tr>
<td>100F 07</td>
<td>6F</td>
<td>1561 07</td>
<td>6F</td>
<td>0E5F FA</td>
<td>1D</td>
<td>1DO5 F1</td>
<td>F1</td>
</tr>
</tbody>
</table>

#### 12K BASIC 3.2

<table>
<thead>
<tr>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1016 06</td>
<td>EE 6E</td>
<td>1568 06</td>
<td>EE 6E</td>
<td>0E60 1D</td>
<td>2F</td>
<td>1DO6 D3</td>
<td>D3</td>
</tr>
<tr>
<td>1017 06</td>
<td>EE 6E</td>
<td>1569 06</td>
<td>EE 6E</td>
<td>0E61 CD</td>
<td>CD</td>
<td>1DO7 07</td>
<td>6F</td>
</tr>
<tr>
<td>1018 06</td>
<td>EE 6E</td>
<td>156A 06</td>
<td>EE 6E</td>
<td>0E62 FA</td>
<td>FD</td>
<td>1DO8 C9</td>
<td>C9</td>
</tr>
<tr>
<td>1019 15</td>
<td>15</td>
<td>156B 06</td>
<td>C2</td>
<td>0E63 1D</td>
<td>1D</td>
<td>1DO9 06</td>
<td>06</td>
</tr>
<tr>
<td>101A 15</td>
<td>15</td>
<td>156C 06</td>
<td>15</td>
<td>0E64 C3</td>
<td>C3</td>
<td>1DOA 01</td>
<td>01</td>
</tr>
<tr>
<td>101B 0F</td>
<td>10</td>
<td>156D 01</td>
<td>15</td>
<td>0E65 74</td>
<td>74</td>
<td>1DOB FE</td>
<td>FE</td>
</tr>
<tr>
<td>101C 0F</td>
<td>10</td>
<td>156E 01</td>
<td>15</td>
<td>0E66 0E</td>
<td>0E</td>
<td>1DOC 2E</td>
<td>2E</td>
</tr>
<tr>
<td>101D D3</td>
<td>D3</td>
<td>156F 01</td>
<td>D3</td>
<td>0E67 0E</td>
<td>0E</td>
<td>1DDC F1</td>
<td>F1</td>
</tr>
<tr>
<td>101E 07</td>
<td>6F</td>
<td>1570 01</td>
<td>6F</td>
<td>0E68 04</td>
<td>10</td>
<td>1DDC CA</td>
<td>CA</td>
</tr>
<tr>
<td>101F 07</td>
<td>6F</td>
<td>1571 01</td>
<td>6F</td>
<td>0E69 CD</td>
<td>D3</td>
<td>1DDC 12</td>
<td>12</td>
</tr>
<tr>
<td>1020 03</td>
<td>15</td>
<td>1572 01</td>
<td>15</td>
<td>0E6A F0</td>
<td>6E</td>
<td>1DDC E5</td>
<td>E5</td>
</tr>
<tr>
<td>1021 03</td>
<td>CD</td>
<td>1573 01</td>
<td>CD</td>
<td>0E6B 1D</td>
<td>00</td>
<td>1DDC E5</td>
<td>E5</td>
</tr>
<tr>
<td>1022 03</td>
<td>CD</td>
<td>1574 01</td>
<td>CD</td>
<td>0E6C 00</td>
<td>00</td>
<td>1DDC E5</td>
<td>E5</td>
</tr>
<tr>
<td>1023 03</td>
<td>CD</td>
<td>1575 01</td>
<td>CD</td>
<td>0E6D 01</td>
<td>10</td>
<td>1DDC E5</td>
<td>E5</td>
</tr>
<tr>
<td>1024 03</td>
<td>CD</td>
<td>1576 01</td>
<td>CD</td>
<td>0E6E 1D</td>
<td>6E</td>
<td>1DDC E5</td>
<td>E5</td>
</tr>
<tr>
<td>1025 03</td>
<td>CD</td>
<td>1577 01</td>
<td>CD</td>
<td>0E6F 0E</td>
<td>00</td>
<td>1DDC E5</td>
<td>E5</td>
</tr>
</tbody>
</table>

### DISK BASIC

<table>
<thead>
<tr>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1053 06</td>
<td>EE 3E</td>
<td>15A5 06</td>
<td>EE 3E</td>
<td>0E71 06</td>
<td>EE 3E</td>
<td>1DDC FA</td>
<td>FD</td>
</tr>
<tr>
<td>1054 03</td>
<td>10</td>
<td>15A6 03</td>
<td>10</td>
<td>0E72 06</td>
<td>EE 3E</td>
<td>1DDC 1D</td>
<td>1D</td>
</tr>
<tr>
<td>1055 07</td>
<td>EE 6E</td>
<td>15A7 06</td>
<td>EE 6E</td>
<td>0E73 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>1056 07</td>
<td>EE 6E</td>
<td>15A8 06</td>
<td>EE 6E</td>
<td>0E74 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>1057 07</td>
<td>EE 6E</td>
<td>15A9 06</td>
<td>EE 6E</td>
<td>0E75 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>1058 07</td>
<td>EE 6E</td>
<td>15AA 06</td>
<td>EE 6E</td>
<td>0E76 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>1059 07</td>
<td>EE 6E</td>
<td>15AB 06</td>
<td>EE 6E</td>
<td>0E77 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>105A 07</td>
<td>EE 6E</td>
<td>15AC 06</td>
<td>EE 6E</td>
<td>0E78 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>105B 07</td>
<td>EE 6E</td>
<td>15AD 06</td>
<td>EE 6E</td>
<td>0E79 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>105C 07</td>
<td>EE 6E</td>
<td>15AE 06</td>
<td>EE 6E</td>
<td>0E7A 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>105D 07</td>
<td>EE 6E</td>
<td>15AF 06</td>
<td>EE 6E</td>
<td>0E7B 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>105E 07</td>
<td>EE 6E</td>
<td>15B0 06</td>
<td>EE 6E</td>
<td>0E7C 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>105F 07</td>
<td>EE 6E</td>
<td>15B1 06</td>
<td>EE 6E</td>
<td>0E7D 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
<tr>
<td>1060 07</td>
<td>EE 6E</td>
<td>15B2 06</td>
<td>EE 6E</td>
<td>0E7E 06</td>
<td>EE 6E</td>
<td>1DDC 06</td>
<td>EE 3E</td>
</tr>
</tbody>
</table>

---

25
The main purpose of this software is to allow development of assembly-language programs. Programs may be entered from the keyboard in standard assembly-language format, edited, assembled, and saved on cassette in either source or machine format. These programs may be then loaded back into memory at some future time for listing, editing or running. Following is a summary of the commands. All address and byte information is in hex.

**FILE /NAME/ AAAA** - Creates a file of name NAME at address AAAA (hex). Up to six text files can be in memory at the same time. The file just created is the current file.

**FILE /NAME/ 0** - This command deletes the file named NAME.

**FILE /NAME/** - Makes the named file the current file.

**FILE** - Prints the starting and ending addresses of the current file, and it's name.

**FILES** - Lists the name, starting & ending address for each of the files in the system. Current file at top.

**DUMP SSSS EEEE** - Dumps the contents of memory from SSSS to EEEE.

**EXEC AAAA** - Executes the machine-language program at address AAAA.

**EXEC /NAME/** - Searches the cassette for named file, loads and exec.

**ENTR AAAA** - Enter data into memory: B1 B2 B3 B4/

**LIST NNNN** - Lists the current file starting at line NNNN.

**DELT NNNN MMMM** - Deletes lines NNNN through MMMM from current file.

**NNNN (text entered)** - Enter a line of text into the current file. Always use 4 decimal digits. It works similar to BASIC, in that the numbered lines are ordered automatically.

**ASSM AAAA** - Assembles current file into address AAAA.

**ASSM AAAA BBBB** - Assembles at address AAAA, but puts code at BBBB.

**CUST** - Execute a customer (user) routine at address E000.

**SFIL** - Save the current file on cassette.

**LFIL** - Load the current file from cassette. The name of the file on tape must match the name of the current file. Be sure there is enough room in memory.

**CFIL** - Check a cassette file written with the SFIL command above, for errors, without overwriting current file.

**APFIL** - Append a file from cassette to the current file.

**SAVE SSSS EEEE** - Save a block of memory from SSSS to EEEE on cassette.

**LOAD SSSS** - Load a block of memory from cassette starting at SSSS.

**CHEK SSSS** - Check a cassette file written with SAVE command.

**NAME /NAME/** - Rename the current file to NAME.

**RNUM N** - Renumber the current file by increment N.

**NLIS NNNN** - Lists the current file at NNNN without line numbers.

This software is useful, but may not be completely free of bugs. It loads into the first 4096 bytes of memory, and uses about 2048 bytes following, for tables and scratch pad. Ctl-C escapes from any printing. Ctl-W freezes printing until another character is typed. The package is available from Tarbell Electronics for the prices listed below:

Cassette Tape with software and instructions: $5.00
A reassembled and patched (updated) listing: $5.00
WRITING PROGRAMS FOR THE CASSETTE INTERFACE

SOMETIMES IT IS NECESSARY TO WRITE ASSEMBLY OR MACHINE LANGUAGE PROGRAMS FOR A PARTICULAR INTERFACE. THE CLASS OF PROGRAMS WE ARE TALKING ABOUT HERE ARE CALLED "DRivers". THESE WOULD BE REQUIRED FOR LINKAGE TO A PIECE OF SOFTWARE FOR WHICH PATCHES ARE NOT PROVIDED IN THE MANUAL.

THIS INTERFACE IS A SYNCRONOUS DEVICE. ONE OF THE IMPLICATIONS OF THIS IS THAT DATA OR PROGRAMS ARE MOST EFFICIENTLY WRITTEN AS A CONTIGUOUS BLOCK, RATHER THAN AS SEPERATE BYTES. THERE ARE A FEW RULES THAT MUST BE FOLLOWED WHEN WRITING SOFTWARE FOR THIS DEVICE:

1. THE FIRST BYTE MUST BE A "START BYTE" WHICH MAY BE ANY BYTE EXCEPT 00, FF, OR E6 (HEXADECIMAL).
2. THE SECOND BYTE MUST BE A "SYNC BYTE" WHICH MUST BE E6 (HEX).
3. THE SOFTWARE MUST BE ABLE TO DELIVER BYTES TO THE INTERFACE AS FAST AS IT CAN ACCEPT THEM, WHICH AT THE STANDARD SPEED, IS 187 BYTES PER SECOND. THIS MEANS THAT ANY LOOP THAT THE PROGRAM GOES THROUGH WHICH IS BETWEEN BYTES, MUST LAST LESS THAN 5.3 MILLISECONDS. AN AVERAGE INSTRUCTION TIME ON THE 8080 WITH NO WAIT STATES IS 2 CYCLES, OR 1 MICROSECOND. THUS, THERE SHOULD BE NO MORE THAN ABOUT 5300 INSTRUCTION EXECUTIONS BETWEEN BYTES BEING SENT OUT. THIS IS NORMALLY NOT A PROBLEM.
4. A SIMILAR CONSTRAINT MUST BE OBSERVED WITH RESPECT TO THE INPUT SOFTWARE, WHICH SHOULD BE ABLE TO ACCEPT DATA AS FAST AS IT IS BEING MADE READY BY THE INTERFACE.

THERE ARE SITUATIONS IN WHICH THE DATA CANNOT BE PROVIDED OR ACCEPTED FAST ENOUGH BY THE SOFTWARE. ONE EXAMPLE OF THIS IS DATA WHICH IS BEING GENERATED BY A PROGRAM RUNNING IN BASIC. THIS PROBLEM IS SOLVED BY SENDING THE DATA OUT AND READING IT IN A LINE AT A TIME, WITH NULLS IN BETWEEN. THE PAGE ENTITLED "HOW TO SAVE AND LOAD DATA FROM A BASIC PROGRAM" IS A SAMPLE PROGRAM THAT SHOWS HOW TO DO THIS. THERE ARE AT LEAST TWO OTHER WAYS TO HANDLE THIS PROBLEM: 1) SEND EACH BYTE AS A SEPERATE BLOCK WITH IT'S OWN START AND SYNC BYTES. 2) ACCUMULATE BYTES IN A BUFFER AREA OF MEMORY, AND START AND STOP THE CASSETTE RECORDER UNDER CONTROL OF THE COMPUTER WHEN IT IS TIME TO DUMP AND REFill THE BUFFER.

OTHER ITEMS TO BE CONSIDERED WHEN WRITING SOFTWARE:
1. SINCE TAPE IS AN IMPERFECT MEDIUM, IT IS GENERALLY USEFUL TO INCORPORATE AN ERROR-CHECKING SCHEME, SUCH AS THE CHECKSUM SYSTEM THAT IS DEMONSTRATED ON THE PAGES ENTITLED "OUTPUT ROUTINE WITH CHECKSUM" AND "INPUT ROUTINE WITH CHECKSUM". THERE ARE MANY SYSTEMS POSSIBLE, EACH PROVIDING DIFFERENT KINDS AND LEVELS OF PROTECTION, AND BOOKS HAVE BEEN WRITTEN ABOUT THESE.
2. IT IS SOMETIMES HANDY TO HAVE AN IDENTIFIER, SUCH AS A NAME, WRITTEN ALONG WITH THE FILE ONTO CASSETTE. AN EXAMPLE IS THE ONE-LETTER NAME GIVEN TO ALTAIR BASIC PROGRAMS WHEN WRITING TO CASSETTE.
3. ANOTHER ITEM THAT IS USEFUL IS A WAY FOR THE PROGRAM TO TELL HOW LONG THE FILE IS. A ONE OR TWO-BYTE HEADER TO INDICATE LENGTH OF A FILE OR BLOCK IS SOMETIMES USED. IN THE MODIFIED PROCESSOR TECHNOLOGY SOFTWARE PACKAGE #1 THAT I SUPPORT, BLOCKS ARE PRECEEDED BY ONE LENGTH BYTE THAT MAY BE FROM 1 TO 255. A LENGTH OF ZERO INDICATES THE END OF THE FILE.
4. A "TYPE BYTE" IS A UNIQUE BYTE FOR A PARTICULAR FORMAT OF FILE, SO THAT A SOPHISTICATED LOADER MAY DISTINGUISH BETWEEN DIFFERENT TYPES, AND LOAD THEM APPROPRIATELY. I USE A 90 (HEX) TYPE BYTE IMMEDIATELY FOLLOWING THE SYNC BYTE ON THE PROC. TECH. SOFTWARE.
USING THE TARRELL CASSETTE INTERFACE UNDER INTERRUPT CONTROL

There is no built-in provision for interrupts in the design. However, it is not too difficult to make a modification to the board to provide for interrupts. Two bits in the control line register are used as interrupt-enable flip-flops. It is necessary to install another 7403 quad 2-input open-collector NAND gate in one of the spare IC positions at the top of the board. The circuit diagram below shows how to connect the 7403:

If you do not have a vectored interrupt card, connect the output pins 3 and 6 to the edge connector pin 73. An interrupt will then be caused when the following conditions are true: 1) Interrupts are enabled with an EI instruction. 2) Control bit 0 is high and the receiver is ready with an input byte; or control bit 1 is high and the transmitter is ready for an output byte. Both conditions 1 and 2 must be true for an interrupt to be caused.

If you do have a vectored interrupt card, connect the output pins 3 and 6 to one of the edge connector pins 4, 5, 6, 7, 8, 9, 10, or 11, depending on what level you want the cassette interface. Be sure that you have the correct polarity to cause an interrupt. The output at pins 3 and 6 goes low to cause an interrupt. If you have a vectored interrupt card that requires a high-going signal to cause an interrupt, you may use the extra gate shown above to invert the line. Read your manual on your interrupt board to make sure.

There is not enough room here to explain how to use the interrupt system, but the following lines show how to enable and disable the two interrupt bits on this board.

enable receiver interrupt: MVI A,1 OUT 6EH
enable transmitter interrupt: MVI A,2 OUT 6EH
enable both interrupts: MVI A,3 OUT 6EH
disable both interrupts: MVI A,0 OUT 6EH
We have purchased the introductory variable-speed PHI-DECK package, which includes the deck, control electronics, power supply, and control box. We have been experimenting with an adapter which connects between the Tarbell Cassette Interface, the read-write heads, and the control electronics. This adapter provides start-stop, forward, rewind, and fast-forward control for the PHI-DECK from the Tarbell Cassette Interface. It also includes read-write electronics for one channel. It is still in the experimental stage, but speeds of 1000 bytes per second have been attained. The information is being provided here for those who would like to experiment further along these lines. The circuit is not guaranteed to work for your application, and will probably not be exactly what we end up with for our use. As progress is made, further refinements will be published, including more supporting software.

The simple program below is handy for experimenting. Flip sense switch 2 up momentarily to pulse the control line.

```
LOOP IN FFH DB FF Read Sense Switches.
OUT 6EH D3 6E Write to Control Port.
IN 6EH DB 6E Read Status Lines.
OUT FFH D3 FF Write To Display Lights.
JMP LOOP C3 00 00 Do it all over again.
```

Control Table

<table>
<thead>
<tr>
<th>Sense Switch</th>
<th>Control Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>RUN</td>
</tr>
<tr>
<td>01</td>
<td>STOP</td>
</tr>
<tr>
<td>10</td>
<td>FF</td>
</tr>
<tr>
<td>11</td>
<td>REW</td>
</tr>
<tr>
<td>12</td>
<td>FF</td>
</tr>
<tr>
<td>13</td>
<td>REW</td>
</tr>
<tr>
<td>14</td>
<td>FF</td>
</tr>
<tr>
<td>15</td>
<td>REW</td>
</tr>
<tr>
<td>16</td>
<td>FF</td>
</tr>
</tbody>
</table>

Sense switch 3 should be up for read, down for write.
74LS73/7473 J-K FLIP-FLOP
WITH CLEAR

74LS74 D-TYPE FLIP-FLOP
WITH PRESET AND CLEAR
**74LS5 4-BIT BISTABLE LATCH**

**74L86 EXCLUSIVE-OR**
SER - SERIAL DATA INPUT. DATA IS MOVED FROM SERIAL INPUT TO QA AFTER THE FIRST CLOCK.

A-B - PARALLEL DATA INPUT. DATA IS PRESENT ON ASSOCIATED OUTPUT (A→QA) AFTER CLOCK. IF MODE IS SHIFTED FROM PARALLEL, DATA WILL MOVE TOWARDS QE ON EACH FOLLOWING CLOCK.

ENB P - ENABLE PARALLEL MODE. WHEN SET, REGISTER IS IN PARALLEL MODE. WHEN RESET, REGISTER IS IN SERIAL MODE.

CK - CLOCK. NO CHANGE IN OUTPUT UNTIL RISING EDGE OF CLOCK.

CLR - CLEAR. WHEN RESET ALL OUTPUTS ARE FORCED TO TO RESET WITHOUT REGARDS TO ANY OTHER INPUT.

QA-QE • OUTPUTS. ALL OUTPUTS USED FOR PARALLEL OUTPUT. FOR SERIAL OUTPUT, ONE OF THE OUTPUTS IS USED DEPENDING ON THE LENGTH DESIRED.

7496 5-BIT SHIFT REGISTER

DM8131 6-BIT COMPARATOR
A, B - SERIAL DATA INPUTS. ONE OF THE TWO MUST BE SET TO ENABLE THE OTHER. DATA IS PRESENT AT QA AFTER ONE CLOCK AND MOVES TOWARD QH ON EACH FOLLOWING CLOCK.

CK - CLOCK. NO CHANGE IN OUTPUT UNTIL RISING EDGE OF CLOCK

CLR - CLEAR. WHEN RESET, ALL OUTPUTS ARE RESET WITHOUT REGARDS TO OTHER INPUTS.

QA-QH - OUTPUTS. DATA IS MOVED FROM A AND B TO QA ON EACH CLOCK AND THE DATA AT QA IS MOVED TOWARDS QH.

74L164 8-BIT PARALLEL-OUT SHIFT REGISTER

**Diagram**

- **Comparitor**
- **Flip-Flop**
- **Output Stage**
- **Trigger**
- **Discharge**
- **Vcc**
- **Reset**

**NE555 Timer**
8720 Bidirectional One-Shot

INPUT SIGNAL

REFERENCE

LIMITER OUTPUT A

LIMITER OUTPUT A

ONE-SHOT OUTPUT Q

ONE-SHOT OUTPUT Q

ONE-SHOT OUTPUT Q

ONE-SHOT "ON" TIME IS DETERMINED BY TIMING COMPONENTS CONNECTED TO PINS 12, 14, AND 15.

8720 Bidirectional One-Shot Input

AND OUTPUT WAVEFORMS
DEAR CUSTOMER,

THANK YOU VERY MUCH FOR PURCHASING A CASSETTE INTERFACE.
I AM INTERESTED TO KNOW WHAT KIND OF PROGRESS YOU HAVE MADE
WITH YOUR TARBEll CASSETTE INTERFACE. I WOULD REALLY APPRCIATE
IT IF YOU WOULD TAKE TIME TO FILL THIS QUESTIONNAIRE OUT. THIS
WILL HELP ME TO PROVIDE YOU WITH BETTER SERVICE IN THE FUTURE.

DID YOUR INTERFACE ARRIVE IN A REASONABLE LENGTH OF TIME? YES NO
WERE ANY OF THE ITEMS DAMAGED IN SHIPMENT? YES NO
WERE ANY OF THE ITEMS MISSING? IF SO, WHAT? YES NO
WAS THE QUALITY OF WORKMANSHIP ON THE BOARD REASONABLE? YES NO
HAVE YOU STARTED CONSTRUCTION YET? YES NO
HAVE YOU HAD ANY PROBLEM UNDERSTANDING THE MANUAL? WHERE? YES NO
WERE THE COMPONENTS OF REASONABLE QUALITY? WHAT WASN'T? YES NO
HAVE YOU COMPLETED THE CONSTRUCTION YET? YES NO
HAVE YOU TESTED THE INTERFACE YET? YES NO
WHAT IS THE MAKE AND MODEL OF YOUR RECORDER?
HAVE YOU ENCOUNTERED ANY PROBLEMS? WHAT? YES NO
IF THERE WERE PROBLEMS, ARE THEY FIXED? IN WHAT WAY? YES NO
HAVE YOU FOUND ANY OF THE ADJUSTMENTS CRITICAL? WHICH? YES NO
ARE YOU DISSATISFIED IN ANY WAY WITH THE UNIT? HOW? YES NO
DO YOU HAVE ANY SUGGESTIONS FOR IMPROVEMENT OF THE DESIGN YES NO
WHAT DO YOU FEEL IS THE MAJOR DISADVANTAGE OF THIS UNIT?
HAVE YOU DEVELOPED ANY SOFTWARE OR HARDWARE RELATIVE TO THE
INTERFACE THAT MIGHT BE OF USE TO OTHER PEOPLE? WHAT?
HAVE YOU TRIED PACKING DENSITIES HIGHER THAN THE STANDARD
800 BITS/INCH? WHAT DENSITY? WITH WHAT RESULTS?
WHAT IS YOUR NAME, ADDRESS, AND PHONE NUMBER?
DO YOU MIND IF I GIVE OUT YOUR NAME TO OTHER CASSETTE USERS?

PLEASE USE THIS SPACE, AND THE REVERSE SIDE, IF NECESSARY, TO
MAKE ANY COMMENTS ON THE INTERFACE THAT MAY BE HELPFUL.

SINCERELY,

DONALD E. TARBELL
144 MIRALESTE DRIVE #106
MIRALESTE, CALIF. 90732