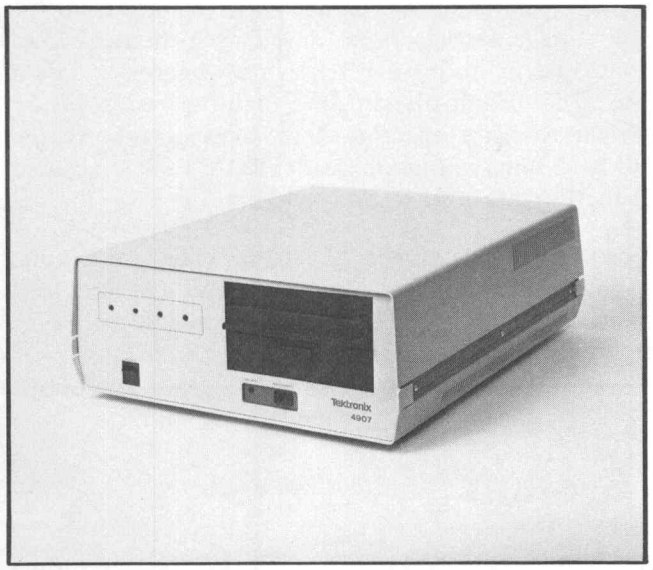
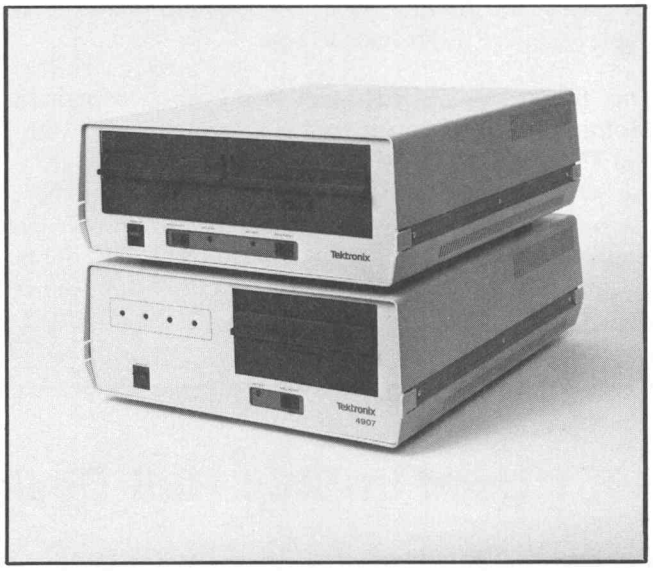
Teknigues

The 4051 Applications Library Newsletter Vol. 1 No. 9



The compact 4907 File Manager brings direct access flexible disc storage to the 4051.



The 4907, Option 31, triple disc drive enables up to 1.89 million bytes of storage.

Debut of the 4907, an Intelligent Disc Unit for the 4051

by Gary P. Laroff

The 4907 FILE MANAGER is a high capability flexible disc mass storage peripheral that makes a 4051-based computation system even more powerful than before. Dual and Triple disc drive systems are available as options. The 4907 is smart. It's built around a microprocessor to handle file management and directory chores. A new ROM Pack, the 4051 FILE MANAGER ROM Pack, is included with the 4907 as a standard accessory. It allows easy access of the 4907's file handling capabilities by the 4051 from a BASIC language program.

Briefly, the 4907 is a direct access, flexible disc device, utilizing a double density read/write feature that enables up to a 630,000 byte capacity per disc. That comes to 1.89 million bytes of storage for the triple drive configuration.

Programming the 4907 is convenient using the new ROM Pack. A major plus is that the file management and directories are handled within the 4907 unit. None of the 4051 read/write memory is required by the 4907 or ROM Pack. In fact, most programs that now do sophisticated file manipulation with either the 4051 internal tape and/or the 4924 Digital Cartridge Tape Unit can generally be rewritten for the 4907 using a much shorter program. The 4907 can support nine files open simultaneously. Therefore, data transfer, sorts and merges can be accomplished with a minimum of programming. As with the tape drives, status can be monitored on each file by using commands such as "ON EOF (3) THEN", "TYPE (6)" and "CLOSE 4". File access is made easy and quick with an advanced multiple level file-by-name system (each name up to 10 characters plus a 3 character extension). This library structure includes a directory that maintains user files, dates and times of access, passwords and available space.

The 4907 FILE MANAGER offers many features available only on large computer systems in addition to those standard features commonly available on calculators and "mini" systems. A real time clock

In This Issue

providing date and time of file activities, multiple file accesses, file security (passwording and execute only) and dynamic allocation of file space are examples of these enhancements.

Many file activities are performed by the 4907 FILE MANAGER which normally require applications programs to perform; file compression, dates and duplication are just a few of these.

The FILE MANAGER and ROM Pack contain a Motorola microprocessor system utilizing 36K of ROM and 8K of RAM. The 4907 instructions are passed from the 4051 via the GPIB in an internal format. These machine level commands are generated by the system firmware interpretation, e.g.: CLOSE, INIT, CREATE, etc., or by the FILE MANAGER ROM Pack subroutines e.g.: CALL "MOUNT", CALL "REWIND", CALL "FORMAT", etc. No system software bootstrapping or special software package is required. As with the internal tape drive, all of the commands may be used within a program and parameters can be generated by the program.

Program overlaying, chaining, and appending are performed in a fraction of the time required when using the tape storage. Directory and file management capabilities allow ease in storing and retrieving programs and data files. The default file structure for programs is binary format. A binary capability similar to the operation of the 4051R05 Binary Program Loader ROM Pack is included in the File Manager ROM Pack.

You'll be seeing a number of 4907 programs for sorting, handling of extensive graphic data bases and data base management here, in TEKniques, in issues to come.

TEKniques

4051 Used in Ship Hull Design System

In the last issue of TEKniques the book *Mathematical Elements For Computer Graphics*, by Doctors David F. Rogers and James Alan Adams, was featured. One of the authors, Dr. David F. Rogers of the U.S. Naval Academy's Department of Aerospace Engineering, spoke at Tektronix recently. His visit gave further insight into the many applications for 4051 Graphic Systems at the U.S. Naval Academy.

Dr. Rogers emphasized that the U.S. Naval Academy is a university, principally an engineering school. Its purpose remains "to prepare midshipmen morally, mentally, and physically to be professional officers in the Naval Service." Fulfilling that purpose has brought increasingly complex subjects into the midshipman's curriculum. Dr. Rogers (and the Academy) are committed to computer graphics to teach these subjects. Computer graphics compresses time for the students, allowing more learning about inherently graphic concepts in less time. Forty 4051 Graphic Systems allow students to directly exercise their programming and other skills to find their own answers.

One of the applications in the school of engineering is ship hull design. Naval Academy students are using computer graphics to design ship hull shapes. Computer graphics in combination with numerically controlled machining is then used to generate models for testing in a towing tank.

Using the 4051 with the Tektronix 4956 Graphic Tablet, the lines which describe the ship hull can be created directly and stored on the timesharing computer system. (Alternately students can create the lines plan using traditional methods. In that case the lines plan is digitized using a Tektronix 4014-1 equipped with a Tektronix 4954 30" x 40" Graphic Tablet. The digitized points are stored on the timesharing computer system.)

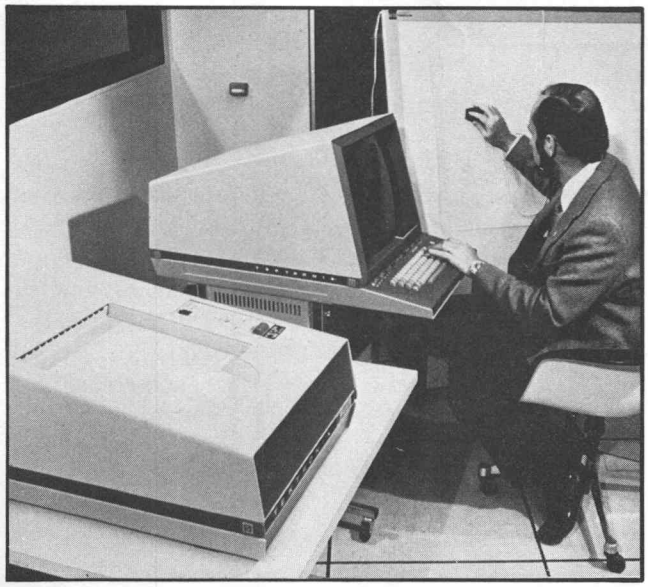
Preliminary fairing of the lines is accomplished using interactive computer graphics and a variety of fairing techniques, such as cubic splines, parabolically blended curves, Bezier curves or B-spline curves. Once the preliminary fairing is acceptable, waterlines and buttock lines are calculated.

Upon completion of the preliminary fairing the lines are transferred via a high speed data link to the DEC PDP 11/45 computer. The lines are then viewed on the Evans and Sutherland Picture System, using an interactive computer graphics program to perform further modifications. The student designer can then view the ship hull in three dimensions while using the hardware features of the Picture System to perform real-time transformations, such as rotation on the picture. At any time, any view of the picture can be transferred to magnetic tape for off-line plotting on the 57" x 89" Xynetics flatbed plotter.

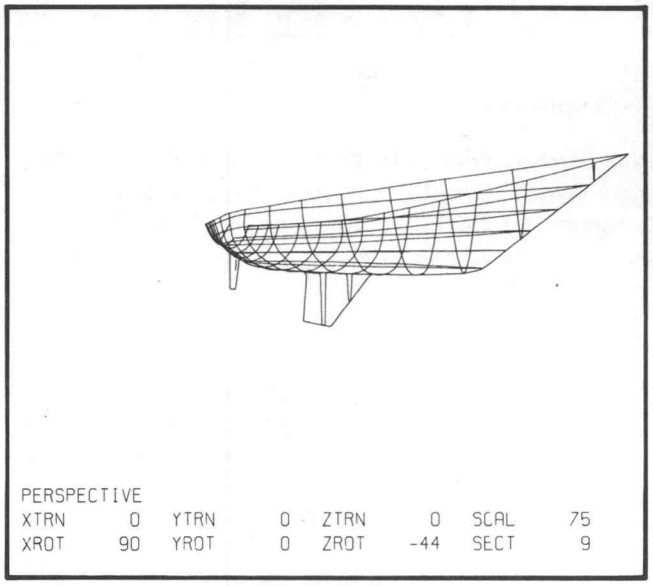
TEKniques, the 4051 Applications Library Newsletter, is published by the Information Display Group of Tektronix, Inc., P.O. Box 500, Group 451, Beaverton, Oregon 97077. It is distributed to TEKTRONIX 4051 users and members of the 4051 Application Library.

Publisher	Ken Cramer
Managing Editor	Patricia Kelley
Editor	Terence Davis
Graphic Designer	Mark Woods

Volume 1 Number 9.



Dr. David F. Rogers uses a Tektronix 4014-1 Computer Display Terminal and a 4954 Graphic Tablet to digitize a set of ship's lines.



The full power of this graphics system is demonstrated in the design of ship hull shapes. Using the 4051 and a graphic tablet, the student designer describes the ship hull contour lines which are automatically digitized and transferred to the computer for storage. Once the preliminary fairing is acceptable, waterlines and buttock lines are calculated.

At any stage in the design process the lines can be used to calculate the stability, shaft horsepower, etc. of the ship. These results are plotted using the 4051 Graphic System and 4631 Hard Copy Unit. Several iterations are generally required before the student designer is satisfied. designer is satisfied.

Once the ship hull is finalized the lines are returned to the DEC PDP 11/45, viewed and checked for fairness on the Xynetics flatbed plotter. Once satisfied, the designer uses the program to create a data file for the numerically-controlled milling machine. The file consists of approximately 8-10 waterlines per inch of model depth sorted in order from the keel to the deck. This waterline data base is then transferred to the internal magnetic tape cartridge of a 4051 Graphic System.

A 4051 Graphic System has been interfaced through its integral IEEE interface (the GPIB) to a Texas Instruments 960B minicomputer. This computer is used as a standard software CNC controller for a Pratt & Whitney TriMac XV numerically controlled machining center. A simple in-house designed interface is used to make the TI 960 think that the 4051 is the TI 960B paper tape reader. Unplugging the 4051's IEEE bus automatically restores normal operation of the TriMac XV's controller.

A BASIC program running in the 4051 Graphic System assists the machine operator with set up and control of the machine, accesses the lines data base on the internal magnetic tape, calculates the necessary tool offsets, and directly generates the necessary computer codes. It then sends them to the TriMac XV controller to drive the machine tool. It simultaneously provides a graphic display for the operator which is one step ahead of the actual cutting operation. This allows the operator to abort the program if any unforeseen interference will occur between the machine and the model or holding clamps.

Upon completion of the model, minimal hand finishing and painting prepares the model for rigging and testing in the tow tank.

This system, developed by Dr. David F. Rogers, the CADIG staff and the Naval Architecture faculty, allows midshipmen Naval Architecture students to easily carry a design through from concept to tests in one semester. This is only one of the many applications of computer graphics that are presently underway at the Naval Academy. It gives an idea of the power, appeal, and flexibility of interactive graphics when used in an integrated system. The 4051 fits ideally in this integrated system because it is flexible and cost effective. The accessibility of a number of 4051 Graphic Systems and the ease of BASIC programming provides hands-on experience for engineering students at the Naval Academy.

Applications in fracture mechanics, wind tunnel testing, gear, cam and machine design, classical stress-strain relationships and a physics simulation involving ray paths through a lens are described in an applications note entitled "Computer Graphics at the Naval Academy." This publication will be available in mid-November and may be secured by writing: Institutional Marketing; Tektronix, Inc.; Del. Sta. 60-371; P.O. Box 500; Beaverton, OR 97077. Tekniques

Speed It Up with Strings

By Leland C. Sheppard
Sunnyvale, CA

The Flowchart Program (51/00-8005/0) has recently been revised to speed up its execution time for any given program (see the Editor's Note in this issue). The string technique described here saved considerable time (and memory too) using that revision. Some of you may want to use it as well.

The 4051's trace facility was used during the revision to determine where time was being spent. Most of the time was spent in several FOR/NEXT loops. These loops were used to build and scan the program's branch table, and to scan each incoming statement for control characters.

The results were significant when the string technique was applied to Flowchart for those functions. Times were reduced two-thirds to three-quarters. The technique is straight-forward, fast, and uses less memory than a strictly numeric approach. The increase in speed results from scanning a list of numbers (in a string format) with one POS statement instead of an entire FOR/NEXT loop. Even with the extra statements required to set up the string, the time savings are significant still.

The memory saving results from the fact that only 1 byte per digit is required to store the number in a string where 8 bytes are required for the equivalent numeric variable. The technique can be applied in any case where many integer numbers must be referenced or saved, and where the numbers are relatively small (say 5 digits or less). There is one restriction on this technique. All the numbers must be stored as the same length (by padding with blanks if necessary) to facilitate the search/scan. Otherwise, 100 cannot readily be differentiated from 1000, and so forth.

The method used to build the string and scan it are shown here.

1. To define the string: DIM A$(B*4) where B is the number of numbers to be included (in this case the length is restricted to 4 digits); A$="" to initialize the string to zero length. DIM B$(4),C$(4) as 2 working strings for building and extracting from the list.

2. To build entries in the string: In the Flowchart example, this is used to extract statement numbers from program statements and store them. P=POS(D$," ",1) which will yield the length+1 of the statement number; C$=SEG(D$,1,P-1) to extract the statement number from the program statement; B$=" " to initialize this string to its full length (4 bytes); B$=REP(C$,1,LEN(C$)) to left-align the number (while retaining excess blanks for padding) in B$; A$=A$&B$ to tack the number onto the string of numbers in A$. D$ is assumed to have been defined to hold 1 program statement.

If the numbers must be ordered other than the way they are read A$ can be initialized to its full length with blanks and the REP function can be used to insert individual numbers into it. However, if a choice exists the concatenation technique used above is much faster than REP.

3. To scan for a particular number: (B$ is assumed to contain a left aligned value 4 digits or less, padded with blanks at the right if necessary) LET Y=1 to establish the search start; P=POS(A$,B$,Y) to get the starting location of B$ if it exists; IF P=0 THEN.....B$ is not contained in A$; LET Y=P+1 to establish the point at which to assume the search. A non-zero result does not guarantee we've found B$ unless the value of P is an integral boundary in A$. For example, IF (P+3)/4 ≠ INT ((P+3)/4) then go back and try again, we've overlapped 2 numbers and have not found B$. An example of this would be where A$ contained 3 numbers, 1420, 1500 and 2015, A$ would appear as "142015002015". The POS function would yield P=3 but would fail the integral boundary test if B$ were "2015".

One other problem exists when using a FOR/NEXT loop instead of the string technique for this kind of an application. When you have found what you are looking for, you will presumably exit the loop before exhausting the loop count. That is, you will branch out of the middle of it. This loses 26 bytes of memory every time you make such an exit; given enough such exits you'll run out of memory! There is a Programming Tip in this issue to restore the 26 lost bytes. TEKniques

*Editor's Note

This issue of TEKniques includes an abstract which describes a revision of the Flowchart Program for 4051 Basic Programs. The revised program (Abstract No. 51/00-8005/1) runs faster and requires less memory than the previous version. Take a look!

Does anyone have a small program for Christmas cards or decorations using the 4051 and 4662 Plotter or 4631 Hard Copy Unit? Send it in for the December issue and receive a free listing of the program of your choice. The best program will receive a free tape also. (Deadline is November 30.)

Coming soon . . . a contest for the best program in a specified area. Watch for the rules and prizes in the next issue.

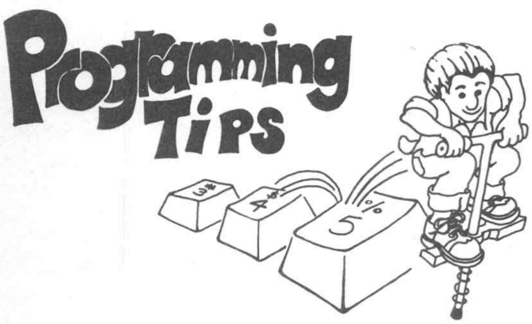
Opportunity To Receive Recognition

By Robert J. Bublitz

The Instructional Media Laboratory at the University of Wisconsin-Milwaukee is updating and expanding its INDEX TO COMPUTER BASED LEARNING (previously entitled INDEX TO COMPUTER ASSISTED INSTRUCTION).

The 1976 Edition of the INDEX contains 1837 computer based learning (CBL) programs available from 222 sources worldwide in more than 160 subject areas. The forthcoming 1978 Edition will cover current CBL activities in computer assisted instruction, computer managed instruction, computer generated tests, and others.

This index presents an excellent opportunity for authors of 4051 educational and medical software to receive recognition. Contact Anastasia Wang, Instructional Media Laboratory, P.O. Box 413, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, by November 30.



Centering Text Strings

by Ken Cramer

Information is often conveyed best when the data is presented graphically. Often, however, the graph alone doesn't tell the whole story. Alphanumerics such as graph titles and axis labels make a graphic presentation more understandable. One of the frequently encountered needs is text that is centered about a vertical axis. One method is to move the cursor to the center point, count the number of characters, divide by two, backspace and print the string. The following BASIC statements perform text centering:

```
100 REM THERE ARE 37 BACKSPACES IN THE SEG FUNCTION
110 REM A$ IS THE STRING TO BE CENTERED
120 MOVE X,Y
130 B$=SEG("HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH",1,LEN(A$)/2)
140 PRINT B$,A$
```

There is also a way to center text using a graphic move rather than backspace characters. This can result in a speed advantage, especially when using the plotter. The BASIC statements which perform this are as follows:

```
100 REM X,Y IS THE LOCATION WHICH THE TEXT
110 REM WILL BE CENTERED AROUND IN USER DATA UNITS
120 REM D IS THE DEVICE ADDRESS
130 REM INP@D,24: X1,Y1 RETURNS THE SCREEN
140 REM LOCATION IN TERMS OF GDU'S.
150 REM A$ IS THE STRING TO BE CENTERED
160 REM S IS THE X-AXIS ALPHASCALE FACTOR FOR THE PLOTTER
170 MOVE @D:X,Y
180 INPUT @D,24:X1,Y1
190 PRINT @D,21:X1-(LEN(A$)-0.5)*(0.896*(D=32)+0.5*S*(D<>32)),Y1
200 PRINT @D:A$
```

Saving Memory FOR/NEXT Loop Exit

**by Leland C. Sheppard
Sunnyvale, CA**

Anytime you exit a FOR/NEXT loop before the loop count is exhausted, i.e., branch out of the middle of it, 26 bytes of memory are lost. Another 26 bytes are lost each time the exit is taken, so it is possible to run out of memory. To prevent this from occurring, bracket the FOR/NEXT you are exiting with a dummy FOR/NEXT and branch to that dummy.

Example:

Bad Loop:

```
  FOR I=1 TO 10
  IF J(I)=0 THEN 5
  NEXT I
5 J(I)=J(I)+1,  etc.
```

Corrected (memory saving) Loop:

```
  FOR A0=1 TO 1                (dummy)
  FOR I=1 TO 10
  IF J(I)=0 THEN 5
  NEXT I
5 NEXT A0                      (end of dummy)
  J(I)=J(I)+1, etc.
```

The 26 bytes are used by the 4051 software to keep track of the beginning of that loop (each loop) and frees the memory when the loop count is exhausted. If you branch out of the middle, the count is never exhausted and therefore the memory is never freed. When the end of the dummy loop is reached (NEXT A0) the software thinks that both loops have been satisfied and will free the 26 bytes used for each; no memory is lost this way.

Numeric Quicker Sort on 4907 Disc Unit

by Gary P. Laroff

The following program is an adaptation of the QUICKER SORT program from General Utilities Programs, Volume 1, PLOT 50: 4050A08. Modifications allow using its powerful sort subroutine on data files stored on the 4907 FILE MANAGER.

One of the first necessary general purpose routines required for a disc-based system is a sort routine that does not require having all of the data in the 4051 at one time. Here we are providing a subroutine, accessible from a user program to sort a linear array of numeric data into increasing order by the fastest general purpose sorting method available. The direct access capability of the 4907 allows READing and WRIting data into file records in

any order. Thus, the 4051 need read only two numbers into its memory at any time to perform the sort. Data READ times from the 4907 disc are further increased by the large 4907 data buffer arranged in 15 caches of 256 bytes each. Data, requested by the 4051, that resides in one of the buffers is automatically sent by the 4907 without wasteful rereading from the disc. The 4907 knows the content of its buffer.

The subroutine is R.S. Scowen's (Scowen, R.S., "Algorithm 271, Quickersort", *Communications ACM*, Vol. 8, #11, Nov., 1965, pp 669-670) method based on bisection and recursions. See this reference for the sort methodology.

The Quickersort subroutine begins with line 2000. The sort is performed on any file that has been opened as logical unit number one. Lines 100-130 are initialization. Lines 1000-1460 provide an example of a program that uses the subroutine. This example program generates a random number array that is sorted with the Quicker Sort subroutine.

DISC OPERATIONS

Line Number	Function Performed
110	Set time and date.
120	Address and enable device 0.
1120	Define user library "GPL".
1130	Delete files in library.
1140	Define file "SORTDATA" in library "GPL" to hold the data to be sorted: "U" public file "C" contiguous records, for speed "A0" number of records, equals number of random numbers to sort "10" size of sequential record, one number
1150	Assign file "GPL/SORTDATA" "I" logical unit number, compatible with sorting subroutine "F" read and write access Q$ status

To run the program, just use any formatted diskette. To use the sorting subroutine in one of your programs, delete lines 100-1460 and use only lines 2000-2680.

```
100 INIT
110 CALL "SETTIM","28-SEP-77 21:00:00"
120 CALL "MOUNT",0,""
130 PRINT @32,26:2

1000 REM
1010 REM  ******START OF ONE FILE QUICK SORT DRIVER ROUTINE******
1020 REM
1030 PRINT "LI** ONE FILE SORT **JJ"
1040 PRINT "THE SORT ALGORITHM ITSELF IS IN A SUBROUTINE STARTING AT";
1050 PRINT " LINE #2000.ILINES #1000-1460 ARE AN EXAMPLE OF HOW TO ";
1060 PRINT "USE THE SUBROUTINE:JJ"
1070 PRINT "HOW LARGE A RANDOM ARRAY WOULD YOU LIKE FOR A TEST SORT = ";
1080 INPUT A0
1090 REM
1100 REM *** DEFINE USER LIBRARY "GPL" WITH DATA FILE "SORTDATA"
1110 REM
1120 CALL "USERLIB","GPL"
1130 KILL "SORTDATA"
1140 CREATE "SORTDATA","UC";A0,10
1150 OPEN "SORTDATA"!1,"F",Q$
1160 REM
1170 REM *** GENERATE RANDOM NUMBERS TO SORT
1180 REM
1190 PRINT "JINUMBERS TO BE SORTEDJ"
1200 FOR I=1 TO A0
1210 A9=INT(1000*A0*RND(-2)+1)
1220 WRITE #1,I:A9
1230 PRINT A9;"I";
1240 NEXT I
1250 PRINT
1260 PRINT "JISORTING NUMBERS"
1270 REM
1280 REM  *** GO TO QUICKERSORT SORTING SUBROUTINE
1290 REM
1300 GOSUB 2010
1310 IF A0<=32 THEN 1370
1320 REM
1330 REM  *** OUTPUT LIST OF SORTED NUMBERS
1340 REM
1350 PAGE
1360 PRINT
1370 PRINT "JISORTED NUMBERSJ"
1380 FOR I=1 TO A0
1390 READ #1,I:A6
1400 PRINT A6;"I";
1410 NEXT I
1420 PRINT @32,26:0
1430 END
1440 REM          ******END OF ONE FILE SORT DRIVER ROUTINE******
1450 REM
1460 REM *-----------------------------------------------------------*

2000 REM
2010 REM              ******START OF ONE FILE SORT******
2020 REM
2030 DELETE B3
2040 DIM B3(24)
2050 IF A0<=1 THEN 2650
2060 B4=0
2070 B1=1
2080 B2=A0
2090 IF B2<=B1 THEN 2600
2100 A5=0
2110 IF B2-B1<1 THEN 2550
2120 IF B2-B1>1 THEN 2200
2130 READ #1,B1:G1
2140 READ #1,B2:G2
2150 IF G1<=G2 THEN 2550
2160 A2=G1
2170 WRITE #1,B1:G2
2180 WRITE #1,B2:A2
2190 GO TO 2550
2200 A5=1
2210 B5=INT(0.5*(B1+B2))
2220 READ #1,B5:G5
2230 A3=G5
2240 READ #1,B1:G1
2250 WRITE #1,B5:G1
2260 B6=B2
2270 A4=B1
2280 A4=A4+1
2290 IF A4>B6 THEN 2440
2300 READ #1,A4:H4
2310 IF H4<=A3 THEN 2280
2320 IF B6<A4 THEN 2440
2330 READ #1,B6:G6
2340 IF G6<A3 THEN 2370
2350 B6=B6-1
2360 GO TO 2320
2370 READ #1,A4:H4
2380 A2=H4
2390 READ #1,B6:G6
2400 WRITE #1,A4:G6
2410 WRITE #1,B6:A2
2420 B6=B6-1
2430 GO TO 2280
2440 READ #1,B6:G6
2450 WRITE #1,B1:G6
2460 WRITE #1,B6:A3
2470 IF B6+B6<=B1+B2 THEN 2520
2480 B7=B1
2490 B8=B6-1
2500 B1=B6+1
2510 GO TO 2550
2520 B7=B6+1
2530 B8=B2
2540 B2=B6-1
2550 IF A5<=0 THEN 2600
2560 B4=B4+2
2570 B3(B4-1)=B7
2580 B3(B4)=B8
2590 GO TO 2090
2600 IF B4<=0 THEN 2650
2610 B1=B3(B4-1)
2620 B2=B3(B4)
2630 B4=B4-2
2640 GO TO 2090
2650 RETURN
2660 REM
2670 REM              ******END OF ONE FILE SORT******
2680 REM
```

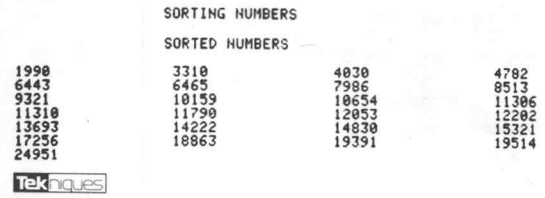
```
                ** ONE FILE SORT **

THE SORT ALGORITHM ITSELF IS IN A SUBROUTINE STARTING AT LINE #2000.
LINES #1000-1460 ARE AN EXAMPLE OF HOW TO USE THE SUBROUTINE:

HOW LARGE A RANDOM ARRAY WOULD YOU LIKE FOR A TEST SORT = 25

                NUMBERS TO BE SORTED

 6465            7986            9321            4782
 10654           19391           6443            4030
 11310           13693           10159           15321
 24951           11306           17256           11790
 3310            10863           14222           8513
 1990            12053           14830           19514
 12202
```



4051 Applications Library Program Abstracts

Documentation and program listings of these programs may be ordered for $15.00 each. Programs will be put on tape for an additional $2.00 handling charge per program and a $26.00 charge for the tape cartridge. (The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc. assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.)

Please use the Applications Library Order Form. Order forms are included in the Membership Packet and are available from your local Tektronix Sales Engineer.

ABSTRACT NUMBER: 51/00-9516/0

Title: **Advanced Media Graphics**
Author: Patrick Rafferty, Lyle Wallis
University of Missouri-Columbia
Memory Requirement: 24K
Peripherals: 4662, 4952
Statements: 744

Advanced Media Graphics (AMG) speeds preparation of classroom aids and program documentation. AMG draws diamonds, boxes, lines and circles; rotates text, draws titles, simplifies outlining, rotates arrow heads, and flowcharts with complete preview and delete capabilities using the 4952 Joystick.

AMG enables the user to create high quality graphics on paper, overhead projector transparencies and slides. An interactive approach has been taken to make it as easy as possible to use the full capabilities.

AMG can be broken down as follows:

1. Title and Text section.
2. Arc-Character section.
3. Box.
4. Circle.
5. Diamond.
6. Line.
7. Outline section allowing user to set tab and character scale variables to speed outlines; includes text input.
8. Flowchart — an independent code section that uses the command processor concept and the pointer command to allow the user to choose both the position and the type of figure with only one keystroke.
9. Arrowhead section.

ABSTRACT NUMBER: 51/00-6103/0

Title: **Publication Distribution**
Author: David A. Beginski
Memory Requirements: 16K (Option 20)
Peripherals: 4631 (Hard Copy Unit)
Statements: 393

The Publication Distribtuion program will accept 100 publications with 24 subscribers per publication. Selection of publications in any order or quantity may be made and printed on the 4051 screen. Changes may be made on any number of publications and/or subscribers. A continuous copy feature is available to print out subscribers' names on the publications selected.

The Publication Distribution program will be used most effectively for magazine distribution in a company of 400 employees or less.

ABSTRACT NUMBER: 51/00-8005/1

Title: **Flowchart Program for 4051 Basic Programs**
Author: Han Klinkspoor, Datatek
Revised by Leland C. Sheppard
Sunnyvale, CA
Memory Requirements: 9K for the program; will run on 16K machine and chart programs with up to 170 branches. On 24K or 32K machines, it will chart programs with 700 or more branches.
Peripherals: 4631 Hard Copy Unit optional
Statements: 405

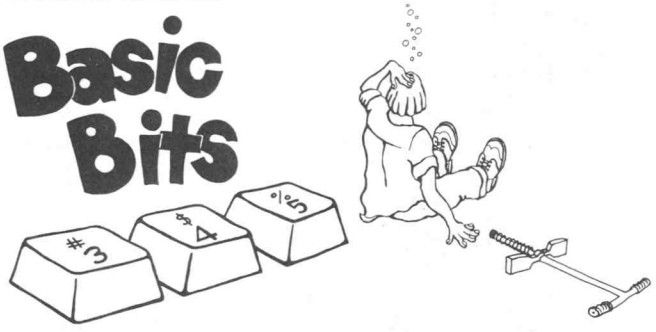
This is a revised version of 51/00-8005/0; the purpose of the revision was to increase the speed of the program while reducing its appetite for memory. Some compromises were made to allow the program to be run on a 16K machine; these are described in the section on restrictions. The overall flow of the program was not modified from the original.

This program will flowchart any 4051 BASIC program from a tape file. It does the job in the following way:

In the first pass, a map of the branches is made to enable "look ahead" in the second pass. In the second pass, program lines are processed one at a time. The line number is stripped off and the branch table is examined to draw incoming or outgoing branches, if any. As each entry in the branch table is processed, the page number on which that reference occurred is plugged back into the branch table for subsequent printing. As the program is charted the current page number and the starting and ending statement numbers shown on that page are printed on the bottom of the page.

Advantages: Runs about 3 times as fast in pass 2 as the old and requires about 1/2 the memory. It includes page number references in the branch table and the range of statement numbers shown on a page are printed with the page number at the bottom.

Restrictions: Limit of 4 character statement numbers to allow the program to run on a 16K machine. This may be modified to 5. Maximum of 20 FOR/NEXT loops unless modified to increase the limit. Page limit is 99 but may be modified. TEKniques



Clean and Cool

As a part of your regular maintenance program, don't neglect cleaning the cooling fan filter on the back panel of your 4051. This is particularly important in a dusty environment.

To avoid tape errors and excessive tape wear, clean the tape head surface with isopropyl alcohol and a swab. The head surface should be cleaned at least every 1,000 hours of average 4051 use. More frequent cleaning may be necessary in a non-office environment where dust and oil are present.

When fingerprints and dust accumulate on the surface of the 4051 display, remove them with glass cleaner and a soft cloth. It is easier and more effective to perform this cleaning when the display surface is cool. Glass cleaners are less likely to streak when applied to cool surfaces.

See Section 10 of the 4051 Graphic System Operator's Manual for more explicit instructions on the limited maintenance required by the 4051.

Software Notice

As a customer service TEKniques publishes Software Notices submitted to it for this purpose. No evaluation or endorsement by TEKTRONIX, INC. of software covered by these Software Notices is implied by such publication. TEKTRONIX, INC. expressly disclaims any obligation of warranty or support relating to software covered by Software Notices.

Hexsadecimetic

Hexsadecimetic (Hex-sa-de'si-ma-tic') is a copyrighted base 2-16 convertor/calculator. This program, written in TEK 4051 BASIC, will convert to or from decimal or perform arithmetic (4 functions) on any numbers to 70 digits, including fractions in any base 2 thru 16. Fractions are rounded to 3 decimal places: this limit can be changed with 1 statement. Booklet includes program listing, abstract, operating instructions, line-by-line commentary, variable and statement cross-references, etc. Required 5K of memory, including data areas. $5.00/copy. 10% discount to multi-4051 installations on total order. California residents add 6% sales tax. Send check or M.O. to: Leland C. Sheppard, Dept. T, P.O. Box 60051, Sunnyvale, California 94086.

Tektronix®
COMMITTED TO EXCELLENCE

TEKTRONIX, INC.
Information Display Group
Applications Library
Group 451
P.O. Box 500
Beaverton, Oregon 97005

ADDRESS CORRECTION REQUESTED