H.A.A.U.G.

THE APPLE BARREL

*** CONTENTS ***

HAAUG APPLE BARREL

CLUB NOTES

MEETING SCHEDULE

The HOUSTON AREA APPLE USERS GROUP holds a general business meeting the second Thursday of each month in the rear chapel of Memorial Lutheran Church, 5800 Westheimer beginning at 6:30 P.M.   A meeting featuring tutorials, access to the HAAUG software library, and special interest group sessions is held beginning at noon the third Saturday of each month at the UT School of Public Health in the Medical Center at 6905 Bertner at Holcomb.

OFFICERS / EXECUTIVE BOARD

President	Steve Knouse
Vice Pres	Clark Johnson
Treasurer	Ruth Hughes
Secretary	Ruth Dill
Software Lib.	Jim Good
Hardcopy Lib.	Larry Baumann
Membership	Lee Gilbreath
IAC Rep.	Robin Cox
Editor	Mike Kramer

APPLE HOTLINE
713-668-3102

The APPLE HOTLINE provides an easy means for the   general public to   learn of meeting topics, news, etc., and can also be used by members to obtain answers to Apple - related questions.   Leave your name, member number (see Apple Barrel label), date, and time. You should get a return call within 48 hours.

MEMBERSHIP INFORMATION

New memberships are $30 and include the HAAUG starter kit. Renewals are $20 per year. Make checks payable to Houston Area Apple Users Group and mail to Lee Gilbreath, 3609 Glen- meadow, Rosenberg,TX 77471.   RENEWALS SHOULD NOT BE SENT TO THE APPLE BARREL.

CALL FOR ARTICLES

Articles and program   listings should be submitted in draft hardcopy form and on disk in Applewriter][or ///, Apple DOS or SOS text, Wordstar, Palantir][OR ///, or Pascal files, or via modem (358-6687). Files should not contain imbedded escape sequences or control characters and should be printed to disk fill justified if possible. Articles must be free of typing or spelling errors and should be grammatically correct as   they cannot be retyped.   Diskettes will be returned to the author provided his name and address are on them. Text should be printed 45 columns wide, listings 40 columns wide. Authors of published articles will receive two blank diskettes per printed page as compensation. The Apple Barrel reserves the sole right to determine which articles are used. Submit articles to:

Houston Area Apple Users Group
Apple Barrel
P.O.Box 610150
Houston, TX, 77208

APPLE BARREL SCHEDULE

The following schedule will be followed for preparation and mailing of the Apple Barrel.
Ads and articles due by 1st of month
Paste ups to printer by the 5th
Pick up from printer by the 15th
Mail by 20th
Received in Houston by 25th
Received in outlying areas by 1st

APPLE BARREL REPRINT POLICY

Unless otherwise noted within the program or article,   any original material published herein may be reprinted without permission by any non-profit Apple club, newsletter, or group, if proper credit is given to the Apple Barrel and the author.

ADVERTISING RATES

AD COST = $0.06 * MULTIPLIER * CIRCULATION

MULTIPLIERS

FULL PAGE	HALF PAGE	QTR PAGE	6TH PAGE	8th PAGE
1.00	0.60	0.35	0.20	0.10

Advertisements should be submitted in camera ready form to H.A.A.U.G., 2218 Running Springs, Kingwood, TX, 77339, by the 1ST of the month. Charges will be billed and accompanied by a copy of the Apple Barrel containing the ad.

EDITOR'S CORNER

This issue of the Apple Barrel features Rick Oshlo's article entitled IRR CALCULATOR. The program included should help you in evaluating your business investments. Don Johnson's A REVIEW OF DICK SPEED CHECKERS will help clear up some of the confusion as to which of the many disk speed checkers are accurate. Clark Johnson's DEALING WITH DOS column is the first of two parts of a review of the fast DOS packages currently flooding the market. Dan Herron continues with the third installment of his series on checksums, this time covering the IBM Modulus 11 Algorithm. Bill Muhlhausen continues with his game reviews, covering Southern Command, U-Boat Command, and others. In Apple /// Peelings, Randall Swearingen reveals how to increase the Apple /// memory from 128K to 256K for $100. Your editor couldn't resist inserting an article he had lying around entitled DEBUGGING APPLESOFT PROGRAMS, which should help some of the newer Apple owners with their BASIC programming.

Unless you are a brand new member you probably noticed that the February/March Apple Barrel looked a little different. Although the new cover design has not yet been completed, we elected to begin putting a cover on our club's magazine (no longer is it a newsletter). Although we intended to have the first magazine three-hole punched so you could put it in a notebook, it didn't get done. We'll try again with this issue. Soon we will try to persuade the stores to sell the Apple Barrel for us.

We've been getting a lot of favorable remarks about the Apple Barrel, both from our members and from the other clubs. It's all been made possible by the participation of an increasing number of writers. It's been particularly gratifying to see some of our newer members, such as Rick Oshlo and Gary Tubb, begin submitting articles.

I'd like to take this opportunity to remind you that three ways have been established for communication with HAAUG. These are the HOT LINE (668-3102), the Post Office Box (P.O.Box 610150, Houston, 77208), and the Business Box at the Saturday and Thursday Meetings. You should no longer use 2218 Running Springs, Kingwood, as this will likely result in a significant delay in your letter reaching the intended destination. When you communicate with HAAUG, try to indicate on the envelope

whether it should go to the membership chairman, the librarian, the Apple Barrel, etc. Membership renewals should be directed to the membership chairman, address changes to the mailing list chairman, inquiries about why you didn't get your disks to the librarian, complaints to Steve Knouse, etc. Inquiries as to why you didn't get your Apple Barrel should be directed to the membership chairman. All the Apple Barrel Editor (me) is responsible for is cutting, pasting, printing, having the labels printed by the mailing list chairman stuck on, and mailing. Your cooperation will help make things run a little more smoothly.

We get frequent requests for back issues of the Apple Barrel. Due to the cost involved, there are normally only a few extras printed. There are only two archival sets of the Apple Barrel that belong to HAAUG. One is in the hard copy library and the other is kept by the Apple Barrel Editor. Obviously it is not practical for copies to be made on a request basis. I don't know if the Apple Barrels in the hard copy library can be checked out, but the Apple Barrel copy is definitely not available to individuals. We've tossed around the idea of printing a Best of the Apple Barrel, but need volunteers to do it. Anyone interested?

ADVERTISERS

SUPPORTING STORES

The following stores support H.A.A.U.G. by offering discounts to members. Be sure to show your appreciation by patronizing them.

Computer Galleries,
        11538 NW Freeway, 956-0900.
        2493 S. Braeswood, 661-0055

Computer Wares, 12839 Gulf Freeway, 481-5600

CTI, 2802 Louisiana, 526-9666.

Micro Solutions, 9949 Harwin #E, 789-5443.

Moore Business Center, 1120 Smith, 237-9063

Softec, Inc., 10875 Katy Freeway, 468-2407

Supertec, FM1960 Bypass at Eastex Freeway, Humble, 446-9770.

MEETING MINUTES
FEBRUARY 10, 1983

Steve Knouse, President, called the meeting to order and welcomed the new members.

Ruth Hughes, Treasurer, reported the following for January:

Income	$3142.00
Cost of Sales	246.19
Expenses	612.68
	Net
Income	$2283.13
Total Assets	$23.498.42

Mike Kramer reported that the February/March issue of the Apple Barrel is now at the printers. Advertisements and articles for publication should be submitted no later than the first of each month in order for the newsletter to be at the printers by the fifth of the month. Everyone should have a copy by the first of the month.

The Apple Barrel is now a "magazine". It is stitched and is 3-hole punched for insertion in a notebook and has a heavy bond cover. Suggested cover designs were passed around for comment. Constructive comments are encouraged. Mike has enough articles for the next issue. Articles for publication in the Apple Barrel should be submitted on disk with a printed copy for review. All articles that have been submitted thus far have been published. Changes of address should be sent to Mike in order to be assured of receiving the newsletter.

The next issue of the newsletter will contain an article by Gary Tubb regarding a program contest. Gary explained that there would be five problems for kids to solve in two hours. The best solution to be published in the Apple Barrel will receive a box of diskettes.

Volunteers are needed to work on a committee to go through the back issues of the Apple Barrel for a special publication of the "Best of the Apple Barrel."

Steve reported that membership as of the 15th of last month was 966.

HAAUG now has a permanent Post Office Box for receipt of all club mail.

P.O. Box 610150, Houston, TX 77208

Hotline Number - 668-3102

We now have a "club business box" which will be in the back of the room at the Thursday meeting and at the Apple Answers desk at the Saturday meeting. This is the place to deposit questions, offers to help, want/don't want ads, help wanted ads, suggestions, etc. You can also use it to submit newsletter articles, membership renewals, member survey forms, etc. All mail sent to the P.O.Box or dropped in the club business box should be marked to the attention of the person you want to receive it; i.e. Apple Barrel, Membership, etc., so that mail does not have to be opened and remailed.

There will be an Apple Answers desk at a central location for technical questions and club business at the Saturday meetings from 1 to 3:30. It will be closed during the general meeting from 2 to 2:30. Someone will be there to answer technical questions.

There will also be a bulletin board for want/don't want ads, help wanted ads, SIG meeting announcements, etc.

Because of the time requirements for copying disks, some change will probably be made in software library procedure. At the present time, club members furnish their own disks, four sides for $1, one time through the line. Orders will be taken from 1 till 4 to be filled from 1:30 to 4:30. Sheets were circulated for volunteers to sign up to donate time or machine for use in copying at the Saturday meeting.

Volunteers are also needed for:

Membership committee - to make starter kits and to distribute club flyers to area computer stores.

Calling committee - to call members about special announcements, meetings, etc.

Contact Glenna Payne, the Volunteer Committee Chairperson, via Hotline, Business Box, or P.O. Box.

We are still in need of donations for a reference library for the Apple Answers desk and for the Hotline. Apple manuals, both hardware and software, and peripheral hardware manuals are especially needed. Please drop any of these manuals you may have into the Business Box with your name, address, and an estimate of reasonable value, and the treasurer will mail you a receipt, which you may deduct on your income tax return (provided the IRS feels we are charitable enough.)

As our membership has grown it has become nearly impossible to maintain the Hardcopy Library. We need new ideas on a way to maintain a transient library for 1000 people. Anyone having suggestions are requested to contact Larry Baumann.

At the Saturday meetings we have use of the first and second floors and the elevators. In the past there has been trouble with children riding up and down the elevators. There was also an instance when the elevators were haphazardly filled with chairs. Members are urgently requested to supervise their children. We cannot afford to lose this meeting location.

Clark Johnson made the following announcements:

Robin Cox will definitely start a new Beginner's Assembly SIG.

The Pascal SIG will talk about 4.1

Check the Bulletin Board at the Saturday meeting for SIG meeting times:

Mike Kramer announced that, beginning with this issue, the Apple Barrel will contain a brief list of those who have Source Account numbers. Those members who would like to have their Source account numbers listed should write him at the P.O. Box or drop a note in the business box.

Phil Lauter requested that anyone interested in Geneology should contact him. At the next Saturday meeting, February 19th, we will have a swap meet. Everyone should bring hardware and software (original copies only) which they no longer need.

Steve will play a video tape about the new Lisa computer throughout the Saturday meeting.

The meeting was turned over to Clark Johnson who presented the question and answer program.

HOUSTON AREA APPLE USERS GROUP
EXECUTIVE COMMITTEE MEETING
MINUTES - February 10, 1983

Steve Knouse, President, requested approval of the Executive Committee to change the address for mailing in membership applications and renewals from Lee Gilbreath's home to the new P.O. Box, except for the IAC mail, which must be sent to the IAC representative. Approved.

There will be an Executive Council meeting on March 8th. Steve will contact members with further details.

Steve turned the floor over the Mike Kramer who presented his recommendations on new Software Library rules. After a very heated discussion and no concensus, the topic was tabled and the meeting adjourned.

There will be a presentation on The Home Accountant at the Saturday meeting.

We need chairmen for the Medical and Home Special Interest Groups. Anyone interested in chairing or helping to coordinate these groups may indicate their interest by dropping a note in the business box.

Robin Cox, International Apple Corps Representative, announced he had current issues of the Apple Orchard for sale at a $.75 discount. He also has old issues to give away. He has subscription forms for anyone wishing them. The IAC has announced an insurance plan which will cover hardware and software for members of member clubs. Anyone interested can obtain applications from Robin. The IAC also has some new software which will be appearing in the Software Library soon.

MUSIC SIG

Those interested in starting a Music SIG or swapping ideas and/or songs should call Johann Matteson at 686-6532 or Lloyd Ernstes at 495-1463.

SPECIAL INTEREST GROUPS

Members who share interests are encouraged to join or form Special Interest Groups (SIGs). Although some of these groups meet separately from the regular meetings, most meet at the regular Saturday session at the times listed below. If you would like to become involved in a SIG, show up at the appropriate meeting room at the Saturday session or call the HOTLINE for meeting time and location if the SIG is not listed on the schedule.

HAAUG SATURDAY SESSION
SIG ROOM ASSIGNMENTS

TIME	AUDIT	MAIN	RM204	RM208	RM228
NOON	_____	BASIC__	CP/M___	_____	EDUC___
1230	_____	BASIC__	CP/M___	_____	EDUC___
1:00	_____	BASIC__	CP/M___	STOCK__	EDUC___
1:30	NEW MEM	BASIC__	CP/M___	STOCK__	EDUC___
2:00	GEN MTG	_____	_____	_____	_____
2:30	SPECIAL	SOFTWAR	_____	PASCAL_	BUSNESS
3:00	SPECIAL	SOFTWAR	ADVNCED	PASCAL_	BUSNESS
3:30	SPECIAL	SOFTWAR	ADVNCED	PASCAL_	BUSNESS
4:00	_____	SOFTWAR	ADVNCED	GAMES__	APPL///
4:30	_____	SOFTWAR	ASSMBLR	GAMES__	APPL///
5:00	_____	SOFTWAR	ASSMBLR	GAMES__	FORTH__
5:30	_____	SOFTWAR	ASSMBLR	_____	FORTH__

SIG CHAIRMEN: CALL GUS AT 481-5329 THE WEEK BEFORE MEETING TO CONFIRM NEED FOR ROOM OR FOR TEMPORARY ROOM ASSIGNMENT.

SIG CHAIRMEN

Business	Rudge Allen
Pascal	Jon Stevens
Statistics	Lindsay Reed
Education	Brian Whaley
Assembler	Robin Cox
Adv.Topics	Tom Murdock
CP/M	Jim Huck
Sci/Engg	Mike Conway
Stocks	George Marsden
FORTH	Steve Knouse
BASIC	Glenna Payne
Games	Bill Muhlhausen
Apple ///	Mike Kramer

NORTH SIDE SIG

Anyone on the north or northwest side of Houston interested in getting together to exchange information and discuss problems should call S.T.Whatley at 931-4217 or Chuck Schrader at 443-1050. Schrader and Whatley should call each other to coordinate the effort.

dBASE II SIG

Those interested in a dBASE II SIG should call Mike Flinn at 667-1869.

IRR CALCULATOR

An Applesoft Program to Calculate
Internal Rates of Return and Net Present Values
by Rick Oshlo

Background

Two of the more common techniques used to evaluate business investments are the Internal Rate of Return (IRR) calculation and its close cousin the Net Present Value (NPV) calculation. The net present value of a cash flow stream is the easier of the two to calculate since a discount rate is assumed and a single equation is applied to each cash flow. VisiCalc has a built in function to accomplish this task. To utilize the VisiCalc @NPV command you must select a discount rate and identify the location of the cash flow stream starting with the cash flow occurring at the <u>end</u> of the first period. The initial investment (the year 0 cash flow) is then added to the @NPV result to obtain the true Net Present Value.

Calculating the internal rate of return is more difficult since it cannot be determined directly. With VisiCalc, the user must repeatedly assume discount rates and run the NPV calculation until the discount rate that causes the NPV to equal zero is identified. That discount rate is defined as the Internal Rate of Return. Although VisiCalc will not solve for the IRR on its own, the calculation is one that is ideal for computer application in a separate program since it requires an iterative process for solution.

The VisiCalc @NPV routine is somewhat restrictive from several standpoints. First, it assumes that the individual cash outflow or inflow occurs as a lump sum at the end of each period. While this is a very common assumption, it is not necessarily the most accurate. In a case where annual periods are assumed and the cash flow stream is received steadily throughout the year, a more "accurate" result is obtained by using mid year discount factors. Mid year factors assume that the individual cash flow occurs as a lump in the <u>middle</u> of the period. Admittedly, the difference is usually small, but it can be significant if you are evaluating alternatives that have different cash flow timings.

A second restriction in VisiCalc is the inability to choose the placement of the final cash flow. The VisiCalc routine assumes that the last cash flow is in the period after the one preceding it. This is a valid assumption in many cases. However, in the case where the last cash flow represents the proceeds from salvage or final asset disposition, it may be more appropriate to recognize that cash flow as a lump at the end of the preceding period. This is particularly true when the "ongoing cash flows" are handled with mid year discounting.

Obviously if the subject of mid year discounting is of no interest, then the precise placement of the final cash flow is unimportant.

A third enhancement to the basic IRR evaluation technique involves calculating a Modified Rate of Return. The traditional IRR technique assumes that all positive cash flows are reinvested at the calculated IRR to earn compound interest over the entire analysis period. It also assumes that future negative cash flows can be invested today to earn interest at the calculated IRR until they are needed to cover the forecasted outlays. Neither of these assumptions is necessarily realistic or valid. By using the Modified IRR technique, it is possible to compensate by using real market interest rates to discount forecasted cash outlays at a "safe" rate that will ensure liquidity when the funds are needed; and to compound the cash inflows at a realistic reinvestment rate to the end of the analysis period.

Using the Modified IRR technique makes the biggest difference when the standard IRR is substantially larger or smaller than market interest rates. As an example consider an investment that has the following cash flow profile:

year 0	-5000	year 8	5000
year 1	-2000	year 9	5000
year 2	-2000	year 10	5000
year 3	-2000	year 11	5000
year 4	-2000	year 12	5000
year 5	5000	year 13	5000
year 6	5000	year 14	5000
year 7	5000	year 15	10000

A standard IRR calculation would yield a rate of return of 20.91%. However assumming a "safe" rate of 8.5% and a "reinvestment" rate of 10.5% will yield a Modified ROR of 15.49%. In many situations this difference could alter an investment decision.

The IRR CALCULATOR Program

This program was written to provide the flexibility that I found lacking in the basic VisiCalc routines and to provide a way to calculate a Modified IRR. <u>IRR Calculator</u> will calculate the net present value, internal rate of return or modified rate of return with up to 40 cash flows. End of year or mid year discount factors and final cash flow placement may be selected for each calculation. The individual cash flows can be in the range of ± $9,999.00 if dollars and cents are needed, ± $9,999,999 if whole dollars are used, or larger figures so long as the data string does not require more

than 7 display columns. The restriction on the number and size of the cash flows was established so that all the cash flows and the calculation results could be displayed at the same time on a single 40 column Apple screen. Either of these numbers could be easily increased with minor program changes to utilize an 80 column card or a scrolling cash flow display.

Data entry is from a previously saved IRR Calculator sequential file, from a VisiCalc file or from the keyboard. Initially, the program asks if the data is coming from a previously saved file. If it is, then you will be prompted for the file name. In the case of an IRR Calculator file, do not include the "IRR." in your response. In order to transfer data from VisiCalc, the cash flow stream first must be saved into a DIF file from the VisiCalc program. Save into the DIF file only that portion of the row or column that contains the actual cash flow data. DO NOT save the entire row or column including labels or extraneous data!

If the answer to the first question is "no" then the program assumes new data will be entered from the keyboard. First you will be asked for the number of cash flows to be entered. Respond with the number excluding CF0. You will then be prompted for each cash flow in the form of "CF0" for the initial (or year 0) cash flow, "CF1" for the cash flow occurring during year 1, and so forth until the specified number of items have been entered.

Once the data has been entered or loaded from a file, you will be given an opportunity to make any corrections, additions or deletions. Once the data is acceptable, you will be asked to select the calculation desired, discount method to be used and the placement of the final cash flow. For NPV and Modified IRR calculations you will also be prompted for the necessary interest and discount rates. Run time for the Applesoft version is approximately 5 - 10 seconds depending on the number of data points and their profile. The Expediter][compiled version executes in about ¼-½ the time required for the Applesoft version.

After the results are displayed, you may edit the data and run a new calculation, dump the results, data and parameters to a printer, save the data to disk or quit the program.

Program Internals

The program is not put together in a particularly "elegant" fashion. That's why it's free. In the accompanying listing, the

"X" next to some lines in the line number column indicates that the line is conditional and controlled by the preceding "IF" statement.

Lines 10 - 460 initialize the program and then either accept data from the keyboard and right justify it or read data from an existing file and display it.

Lines 470 - 730 contain the data editing routine.

Lines 740 - 950 accept input on the calculation method etc. and send the program off to the correct routine.

Lines 960 - 1110 contain the routine to calculate a standard IRR. It uses the Newton-Raphson method to converge on the answer rather than the more common technique of setting outside limits and successively cutting the interval in half. My experiments indicate that the Newton-Raphson method will converge in approximately ½ the time and number of iterations as are required for the other method.

Lines 1210 - 1360 contain the printer dump routine. It is designed for an interface card in Slot 1 that utilizes a Control-I "80N" to lock the screen.

Lines 1390 - 1570 save the data to a new or existing IRR Calculator file if you so choose.

Lines 1610 - 1780 contain the data display subroutine.

Lines 1790 - 1980 calculate the Modified IRR. All the negative data is discounted back to the present at the specified rate; all the positive data is compounded forward to the last period at its specified rate; and a compounded interest rate is calculated between the two figures.

Lines 1990 - 2080 calculate a standard net present value.

Lines 2090 - 2530 read a VisiCalc DIF file and verify its data format.

Line 2550 contains an error message that is utilized in several places throughout the program.

For those that want to tinker with the program but hate typing, it has been submitted to the HAAUG software library in both its Applesoft and compiled versions (a dollar to the club is cheap to save the typing).

```
10      REM XXXXXXXXXXXXXXXXXXXXXXXX
20      REM X IRR & NPV CALCULATOR X
30      REM X        WRITTEN BY      X
40      REM X        RICK OSHLO      X
50      REM XXXXXXXXXXXXXXXXXXXXXXXX
60      DIM CF(40)
      : D$ = CHR$ (4)
70      TEXT
      : HOME
80      HTAB (3)
      : INVERSE
      : PRINT " INTERNAL RATE OF RETURN
             CALCULATOR "
      : NORMAL
      : POKE 34,2
90      VTAB 5
      : PRINT SPC( 2)"THIS PROGRAM CALC
             ULATES THE INTERNAL      RATE
             OF RETURN OR NET PRESENT VAL
             UE                OF UP TO 40 CAS
             H FLOWS."
95      PRINT
      : PRINT SPC( 2)"CASH FLOW VALUES
             FROM $-9,999,999 TO      $9,99
             9,999 IN WHOLE DOLLARS OR FR
             OM      $-9,999.99 TO $9,999.
             99 IN DOLLARS &           C
             ENTS ARE ACCEPTED.
100     VTAB 15
      : PRINT SPC( 5);"USING PREVIOUSLY
             SAVED FILE? ";
      : GET A$
      : IF A$ < > "Y" AND A$ < > "N"
             THEN PRINT CHR$ (7)
  X   : GOTO 100
110     PRINT A$
120     IF A$ = "Y" THEN 340
130     HOME
140     PRINT SPC( 2);"INPUT CASH OUTFLO
             WS AS NEGATIVE (-)"
150     PRINT SPC( 2);"INPUT CASH INFLOW
             S AS POSITIVE (+)"
160     PRINT SPC( 5);"INITIAL INVESTMEN
             T IS CF(0)"
170     INPUT "NUMBER OF CASH FLOWS EXCL
             UDING CF0? ";N
180     IF N < 1 OR N > 39 THEN VTAB (10
             )
  X   : PRINT CHR$ (7)"NUMBER OF CASH F
             LOWS MUST BE"
  X   : PRINT "BETWEEN 1 AND 39 EXCLUDI
             NG CF0"
  X   : FOR I = 1 TO 2000
  X   : NEXT
  X   : GOTO 130
190     VTAB 8
      : INPUT "CF0 ";CF(0)
      : IF LEN ( STR$ ( ABS (CF(0)))) >
             7 THEN GOSUB 2550

  X   : VTAB 8
  X   : CALL - 868
  X   : GOTO 190
200     VTAB 8
      : PRINT "CF0"; SPC( 9 - LEN (
             STR$ (CF(0))));CF(0)
210     FOR I = 1 TO N
220        FX = 0
      :    REM  FLAG TO PRINT OR INPUT A
             CASH FLOW. FX=0 FOR INPUT
             CF(I) & FX=1 TO PRINT CF(
             I) RIGHT JUSTIFIED
```

```
230        IF I = 1 OR I = 14 OR I = 27
             THEN VT = 8
  X   :    REM  DISPLAY DATA IN THREE CO
             LUMNS OF 13 ENTRIES EACH
240        VT = VT + 1
250        VTAB (VT)
260        IF I > 13 THEN HTAB 14
270        IF I > 26 THEN HTAB 27
280        IF FX = 1 THEN PRINT "CF";I;
             SPC( 10 - LEN ( STR$ (I))
             - LEN ( STR$ (CF(I))));CF(
             I)
  X   :    GOTO 320
290        PRINT "CF";I;
300        CALL - 868
      :    INPUT "";CF(I)
      :    IF LEN ( STR$ ( ABS (CF(I))))
             > 7 THEN GOSUB 2550
  X   :    GOTO 250
310        FX = 1
      :    GOTO 250
320     NEXT
330     GOTO 470
340     VTAB 20
      : PRINT "IRR CALCULTOR OR"
      : PRINT " VISICALC DIF FILE? (I/D
             ) ";
      : GET A$
      : IF A$ < > "I" AND A$ < > "D"
             THEN PRINT CHR$ (7)
  X   : GOTO 340
350     PRINT
      : VTAB 20
      : CALL - 958
360     IF A$ = "D" THEN 2090
  X   : REM  GOTO SUBROUTINE TO  READ A
             VISICALC DIF FILE
370     INPUT "IRR CALCULATOR FILE NAME?
             ";F$
380     F$ = "IRR." + F$
390     PRINT D$;"READ";F$
400     INPUT N
410     FOR I = 0 TO N
      :    INPUT CF(I)
      : NEXT
420     INPUT IR$
      : REM  DETERMINE IF STORED DATA I
             S FOR IRR OR NPV
430     IF IR$ = "I" THEN INPUT R
  X   : GOTO 450
440     INPUT IR
      : INPUT X
450     PRINT D$;"CLOSE";F$
460     GOSUB 1610
      : REM  GO TO DISPLAY FORMATTING R
             OUTINE
470     VTAB 23
      : CALL - 958
480     PRINT "ALL CASH FLOWS OK? ";
      : GET A$
      : IF A$ < > "Y" AND A$ < > "N"
             THEN PRINT CHR$ (7)
  X   : GOTO 470
490     PRINT
500     IF A$ = "Y" THEN 740
520     VTAB 23
      : CALL - 958
530     INPUT "CHANGE WHICH CASH FLOW? "
             ;A
540     IF A > N THEN N = A
  X   : REM  IF A>N THEN MUST BE ADDING
             NEW CASH FLOWS.  SET N=A
```
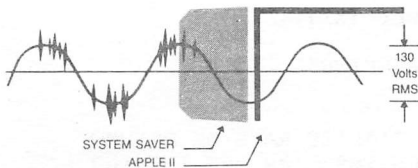
System Saver™

The most important peripheral for your Apple® II.
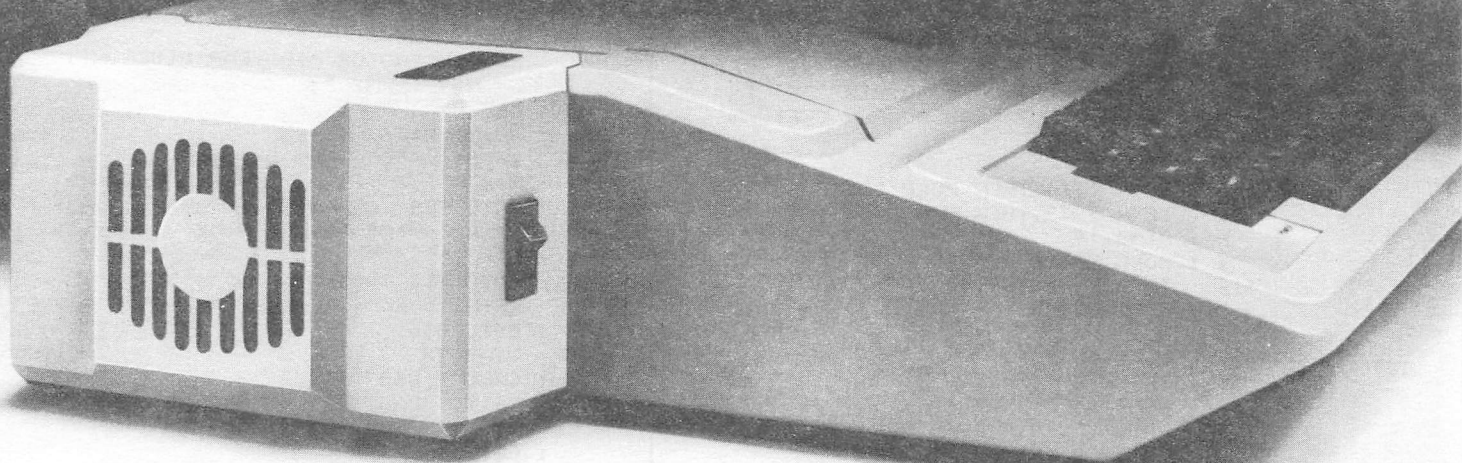
For Line Surge Suppression

The SYSTEM SAVER provides essential protection to hardware and data from dangerous power surges and spikes.

By connecting the Apple II power input through the SYSTEM SAVER, power is controlled in two ways: 1) Dangerous voltage spikes are clipped off at a safe 130 Volts RMS/175 Volts dc level. 2) High frequency noise is smoothed out before reaching the Apple II. A PI type filter attenuates common mode noise signals by a minimum of 30 dB from 600 khz to 20 mhz, with a maximum attenuation of 50 dB.

For Cooling

As soon as you move to 64K RAM or 80 columns on your Apple II you need SYSTEM SAVER.

Today's advanced peripheral cards generate more heat. In addition, the cards block any natural air flow through the Apple II creating high temperature conditions that substantially reduce the life of the cards and the computer itself.
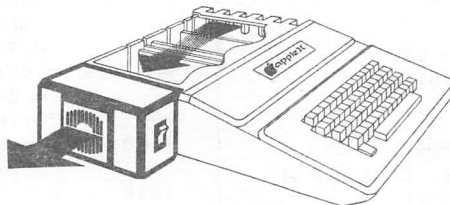
SYSTEM SAVER provides correct cooling. An efficient, quiet fan draws fresh air across the mother board, over the power supply and out the side ventilation slots.

For Operating Efficiency

SYSTEM SAVER contains two switched power outlets. As shown in the diagram, the SYSTEM SAVER efficiently organizes your system so that one convenient, front mounted power switch controls SYSTEM SAVER, Apple II, monitor and printer.
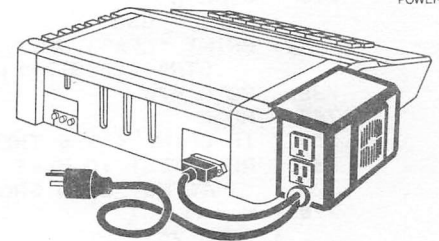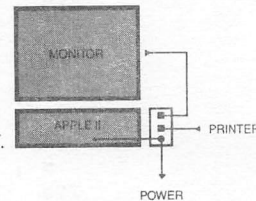
The heavy duty switch has a pilot light to alert when system is on. You'll never use the Apple power switch again!

Easy Installation

Just clips on. No mounting or hardware required. Color matched to Apple II.
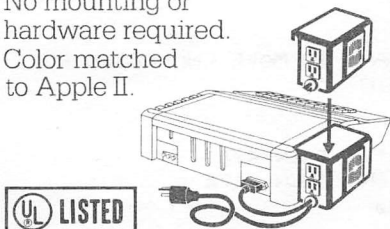
Compatible with Apple Stand
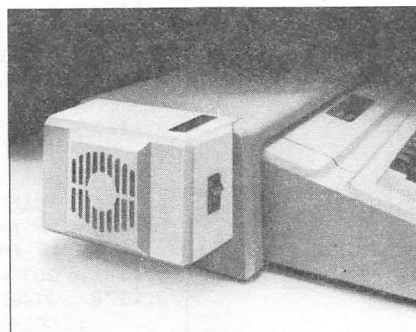
```
550   B = A
   : REM   B WILL BE USED TO SET VTAB
         POSITION.
560   IF A = 0 THEN 710
570   REM   POKES IN 590,600 & 690 SET
         WINDOW WIDTH SO REST OF LIN
         E IS NOT ERASED UPON <RETURN
         >
580   FX = 0
590   IF A < 14 THEN POKE 33,13
600   IF A > 13 THEN B = A - 13
 X  : HTAB 14
 X  : POKE 33,26
610   IF A > 26 THEN B = A - 26
 X  : POKE 33,40
 X  : HTAB 27
620   VTAB (B + 8)
630   PRINT SPC( 12)
640   HTAB ( POS (0) - 11)
650   PRINT "CF";A; SPC( 2 - LEN (
         STR$ (A)));
660   IF FX = 1 THEN FX = 0
 X  : PRINT SPC( 8 - LEN ( STR$ (CF(A
         ))));CF(A)
 X  : GOTO 690
670   CALL - 868
   : INPUT "";CF(A)
   : IF LEN ( STR$ ( ABS (CF(A)))) >
         7 THEN GOSUB 2550
 X  : GOTO 590
680   FX = 1
   : GOTO 600
690   POKE 33,40
700   GOTO 470
710   VTAB 8
   : CALL - 868
   : INPUT "CF0 ";CF(0)
   : IF LEN ( STR$ ( ABS (CF(0)))) >
         7 THEN GOSUB 2550
 X  : GOTO 710
720   VTAB 8
   : CALL - 868
   : PRINT "CF0"; SPC( 9 - LEN (
         STR$ (CF(0))));CF(0)
730   GOTO 470
740   HOME
750   IF CF(N) < > 0 THEN 780
760   REM  TEST TO MAKE SURE CF LIST H
         AS NOT BEEN SHORTENED
770   N = N - 1
   : GOTO 750
780   VTAB 7
   : PRINT SPC( 5);"CALCULATE NET PR
         ESENT VALUE OR"
   : PRINT SPC( 5);"INTERNAL RATE OF
         RETURN? (N/I) ";
   : GET IR$
   : PRINT
790   IF IR$ = "N" THEN HOME
 X  : PRINT SPC( 5);"NET PRESENT VALU
         E CALCULATION"
 X  : GOTO 850
800   IF IR$ < > "I" THEN PRINT CHR$ (
         7)
 X  : GOTO 780
810   HOME
   : PRINT SPC( 3);"INTERNAL RATE OF
         RETURN CALCULATION"
820   VTAB 9
   : CALL - 868
830   PRINT "STANDARD OR MODIFIED IRR?
         (S/M) ";
   : GET SM$
   : IF SM$ < > "S" AND SM$ < > "M"
         THEN PRINT CHR$ (7)
 X  : GOTO 820
```

```
840   PRINT SM$
850   VTAB 11
   : CALL - 868
860   PRINT "END OR MID YEAR DISCOUNTI
         NG? (E/M) ";
   : GET B$
   : PRINT B$
870   IF B$ = "E" THEN D = 0
 X  : M = 0
 X  : GOTO 900
880   IF B$ = "M" THEN D = .5
 X  : M = .5
 X  : GOTO 900
890   PRINT CHR$ (7)
   : GOTO 850
900   VTAB 13
   : CALL - 868
910   PRINT "LAST CF IN SAME YR AS THE
         ONE BEFORE? ";
   : GET C$
   : IF C$ < > "Y" AND C$ < > "N"
         THEN PRINT CHR$ (7)
 X  : GOTO 900
920   PRINT C$
930   IF C$ = "Y" THEN M = 1
940   IF IR$ = "N" THEN 1990
 X  : REM  GO TO NET PRESENT VALUE CA
         LCULATION
950   IF SM$ = "M" THEN 1790
 X  : REM  GO TO MODIFIED ROR CALCULA
         TION
960   VTAB 19
   : PRINT SPC( 16);
   : FLASH
   : PRINT "WORKING"
   : NORMAL
970   R = 0
   : REM   INITIALIZE RATE OF RETURN.
         LINES 980-1100 ARE THE IRR
         CALCULATION ROUTINE. IT USES
         THE NEWTON-RAPHSON TECHNIQU
         E TO CONVERGE ON THE IRR.
980   X = CF(0)
   : X1 = 0
990   FOR I = 1 TO (N - 1)
1000    T = CF(I) X EXP ( - (I - D) X
         R)
1010    X = X + T
1020    X1 = X1 + ( - (I - D) X T)
1030   NEXT
1040   X = X + (CF(N) X EXP ( - (N - M)
         X R))
1050   X1 = X1 - (N - M) X CF(N) X EXP
         ( - (N - M) X R)
1060   R1 = R + ( - X / X1)
1070   IF ABS (R1 - R) < .00005 THEN 11
         00
 X  : REM   CALCULATE IRR TO AN ACCURA
         CY OF .005%
1080   R = R1
1090   GOTO 980
1100   R = INT (( EXP (R) - 1) X 10000
         + .5) X .01
1110   GOSUB 1610
1120   VTAB 23
   : CALL - 958
1130   PRINT "WANT TO MAKE CHANGES & RU
         N AGAIN? ";
   : GET A$
   : PRINT
1140   IF A$ = "Y" THEN 470
1150   IF A$ = "N" THEN 1170
1160   GOTO 1120
1170   VTAB 23
   : CALL - 958
```

```
1180   PRINT "WANT A PRINTOUT? ";
       : GET A$
       : IF A$ < > "Y" AND A$ < > "N"
             THEN PRINT CHR$ (7)
    X  : GOTO 1170
1190   PRINT A$
1200   IF A$ = "N" THEN 1380
1210   PRINT D$;"PR#1"
1220   PRINT CHR$ (9);"80N"
1230   PRINT
       : PRINT
       : PRINT SPC( 3);"INTERNAL RATE OF
             RETURN CALCULATOR"
1240   PRINT
1250   GOSUB 1620
1260   IF SM$ = "S" THEN 1300
1270   PRINT
       : PRINT "MODIFIED RATE OF RETURN
             METHOD USED"
1280   PRINT "NEGATIVE CF DISCOUNT RATE
             IS ";NP;"%"
1290   PRINT "POSITIVE CF DISCOUNT RATE
             IS ";PP;"%"
1300   PRINT
       : PRINT
1310   IF B$ = "M" THEN PRINT "MID YEAR
             DISCOUNTING USED"
    X  : GOTO 1330
1320   PRINT "END OF YEAR DISCOUNTING U
             SED"
1330   IF C$ = "Y" THEN PRINT "CF";N;"
             IS AT END OF PERIOD ";(N - 1
             )
    X  : GOTO 1350
1340   PRINT "CF";N;" IS IN PERIOD ";N
1350   PRINT
       : PRINT
       : PRINT
       : PRINT
       : PRINT
       : PRINT
1360   PRINT D$;"PR#0"
1370   GOTO 1120
1380   VTAB 23
       : CALL - 958
1390   PRINT "WANT TO SAVE THE DATA? ";
       : GET A$
       : IF A$ < > "Y" AND A$ < > "N"
             THEN PRINT CHR$ (7)
    X  : GOTO 1380
1400   PRINT
       : IF A$ = "N" THEN 1580
1410   IF F$ = "" THEN 1450
1420   VTAB 23
       : CALL - 958
1430   PRINT "USE " CHR$ (34) RIGHT$ (F
             $,( LEN (F$) - 4)) CHR$ (34)
             " AS FILE NAME? ";
       : GET A$
       : IF A$ < > "Y" AND A$ < > "N"
             THEN PRINT CHR$ (7)
    X  : GOTO 1420
1440   PRINT A$
       : IF A$ = "Y" THEN 1480
1450   VTAB 23
       : CALL - 958
1460   INPUT "NEW FILE NAME? ";F$
1470   F$ = "IRR." + F$
1480   PRINT D$;"OPEN";F$
1490   PRINT D$;"DELETE";F$
1500   PRINT D$;"OPEN";F$
1510   PRINT D$;"WRITE";F$
1520   PRINT N
```

```
1530   FOR I = 0 TO N
       :   PRINT CF(I)
       : NEXT
1540   PRINT IR$
1550   IF IR$ = "I" THEN PRINT R
    X  : GOTO 1570
1560   PRINT IR
       : PRINT X
1570   PRINT D$;"CLOSE";F$
1580   VTAB 22
       : CALL - 958
1590   PRINT SPC( 10);"ALL DONE -- BYE!
1600   POKE 34,0
       : END
1610   HOME
1620   IF IR$ = "N" THEN L = LEN ( STR$
             (IR) + STR$ (X)) X .5
    X  : PRINT SPC( 7 - L)"NET PRESENT V
             ALUE AT "IR"% IS $"X
    X  : GOTO 1640
1625   IF S$ < > "" THEN S$ = ""
    X  : PRINT
    X  : GOTO 1640
    X  : REM  SKIP AROUND PRINTOUT OF IR
             R IF DATA WAS READ FROM A DI
             F FILE AND IRR HAS NOT YET B
             EEN CALC.
1630   L = LEN ( STR$ (R)) X .5
       : PRINT SPC( 6 - L)"INTERNAL RATE
             OF RETURN IS "R"%"
1640   CF = 0
1650   FOR XX = 0 TO N
       :   CF = CF + CF(XX)
       : NEXT XX
       : CF = INT (CF X 100 + .5) X .01
1660   CL = LEN ( STR$ (CF)) X .5
1670   PRINT
       : PRINT SPC( 7 - CL)"CASH FLOWS A
             S SHOWN TOTAL $"CF
1680   PRINT
       : PRINT
       : PRINT "CF0"; SPC( 9 - LEN (
             STR$ (CF(0))));CF(0)
1690   FOR I = 1 TO 13
1700     LI = LEN ( STR$ (I))
1710     LCF = LEN ( STR$ (CF(I)))
1720     PRINT "CF";I; SPC( 10 - LI - L
             CF);CF(I);
1730     LCF = LEN ( STR$ (CF(I + 13)))
1740     PRINT " CF";(I + 13); SPC( 8 -
             LCF);CF(I + 13);
1750     LCF = LEN ( STR$ (CF(I + 26)))
1760     PRINT " CF";(I + 26); SPC( 8 -
             LCF);CF(I + 26)
1770   NEXT
1780   RETURN
1785   REM  MODIFIED ROR CALCULATION RO
             UTINE
1790   PRINT
       : INPUT "NEG CF DISCOUNT RATE? (%
             ) ";NP
1800   PRINT
       : INPUT "POS CF DISCOUNT RATE? (%
             ) ";PP
1810   VTAB 20
       : PRINT SPC( 16);
       : FLASH
       : PRINT "WORKING"
       : NORMAL
1820   NR = 1 + NP X .01
       : PR = 1 + PP X .01
1830   R = 0
       : X = 0
       : X1 = 0
```

```
1840   IF CF(0) < 0 THEN X = CF(0)
1850   IF CF(0) > 0 THEN X1 = CF(0)
1860   NX = N
1870   IF C$ = "Y" THEN NX = NX - 1
1880   FOR I = 1 TO (N - 1)
1890      IF CF(I) < 0 THEN X = X + CF(I
            ) / (NR ^ (I - D))
1900      IF CF(I) > 0 THEN X1 = X1 + CF
            (I) % PR ^ (NX - I + D)
1910   NEXT
1920   IF CF(N) < 0 THEN X = X + CF(N)
          / NR ^ (N - M)
1930   MX = 0
1940   IF C$ = "N" AND B$ = "M" THEN MX
          = .5
1950   IF CF(N) > 0 THEN X1 = X1 + CF(N
          ) % PR ^ (MX)
1960   R = (X1 / - X) ^ (1 / NX) - 1
1970   R = INT (R % 10000 + .5) % .01
1980   GOTO 1110
1985   REM   NET PRESENT VALUE CALCULATI
          ON ROUTINE
1990   VTAB 15
        : INPUT "NPV DISCOUNT RATE? (%) "
;IR
2000   VTAB 20
        : PRINT SPC( 16);
        : FLASH
        : PRINT "WORKING"
        : NORMAL
2010   NR = 1 + IR % .01
2020   X = CF(0)
2030   FOR I = 1 TO (N - 1)
2040      X = X + CF(I) / (NR ^ (I - D))
2050   NEXT
2060   X = X + CF(N) / (NR ^ (N - M))
2070   X = INT (X + .5)
2080   GOTO 1110
2090   REM   THIS SUBROUTINE READS A VIS
          ICALC DIF FILE
2110   GOSUB 2240
2120   GOSUB 2300
2130   CM = 0
2140   FOR I = 0 TO N
2150      GOSUB 2420
2160      IF T1 = 0 THEN CF(I) = V1
2170      IF T1 = 1 THEN CF(I) = 0
2180      CL = LEN ( STR$ ( INT (CF(I)))
            )
        :   IF CL > CM THEN CM = CL
2190   NEXT I
2200   PRINT CHR$ (4)"CLOSE"FD$
2210   IF CM < 5 THEN 2230
2220   FOR I = 0 TO N
        :   CF(I) = INT (CF(I) + .5)
        : NEXT
        : GOTO 460
2230   FOR I = 0 TO N
        :   CF(I) = INT (CF(I) % 100 + .5
            ) % .01
        : NEXT
        : GOTO 460
2240   PRINT
        : INPUT "DIF FILE NAME? ";FD$
2260   PRINT CHR$ (4)"OPEN"FD$
2270   NV = 0
2280   NT = 0
2290   RETURN
2300   PRINT CHR$ (4)"READ"FD$
2310   INPUT T$
2320   INPUT S,ND
2330   INPUT S$
2340   IF T$ = "VECTORS" THEN 2380
2350   IF T$ = "TUPLES" THEN N = ND - 1
     % : GOTO 2310
```

```
2360   IF T$ = "DATA" THEN RETURN
2370   GOTO 2310
2380   IF ND = 1 THEN 2310
2390   VTAB 18
        : CALL - 958
        : PRINT "MORE THAN 1 ROW OR COLUM
            N WAS SAVED IN  THE DIF FILE
.   SAVE ONLY THAT PART OF
            THE SINGLE ROW OR COLUMN CON
            TAINING THE ACTUAL CASH FLOW
            DATA.  TYPE " CHR$ (34)"RUN
            " CHR$ (34)" TO    RESTART IR
            R CALCULATOR."
2400   PRINT CHR$ (4)"CLOSE"
2410   POKE 34,0
        : END
2420   GOSUB 2480
2430   IF T1 < > - 1 THEN 2510
2440   IF S$ < > "BOT" THEN 2510
2450   GOSUB 2480
2460   IF T1 = - 1 THEN 2510
2470   RETURN
2480   INPUT T1,V1
2490   INPUT S$
2500   RETURN
2510   PRINT
        : PRINT "ERROR IN DIF FILE FORMAT
            ."
2520   PRINT CHR$ (4)"CLOSE"
2530   POKE 34,0
        : END
2550   POKE 33,40
        : VTAB 23
        : CALL - 958
        : PRINT CHR$ (7)"CASH FLOW > SEVE
            N PLACES IN LENGTH."
        : FOR IZ = 1 TO 2000
        : NEXT IZ
        : VTAB 23
        : CALL - 958
        : RETURN
```

DISK SPEED CHECKERS

by
Don Johnson

The rotational speed of a diskette is an important factor in the performance characteristics (minimal read/write errors) of your Apple disk drive. The purpose of this article is two-fold: first, the importance of disk speed is discussed and then a review of the programs which can assist you in maintaining accurate speeds is provided.

Sequences of pulses representing bits are recorded onto concentric rings (tracks) on a diskette by switching on and off the voltage applied to the write head. Apple has specified the number of information bits that are written on each track. If the bits are not written quickly enough, the specified number of bits may not fit into one rotation of the diskette. A very complete and informative description of the encoding of bits into nibbles and the format of nibbles on a track is contained in Chapter 3 of "Beneath Apple DOS".

WILDCARD™

MAKES BACK-UP COPIES OF PROTECTED SOFTWARE QUICKLY, EASILY, WITH JUST A PUSH OF A BUTTON.

New software locking schemes have rendered even the latest generation of copy programs virtually unusable. Locksmith™, Nibbles Away™ and other "Nibble copiers" require complicated parameter settings, much patience and great effort to use. More often than not, the results are disappointing. WILD-CARD is different. Rather than copying disks track by track, WILDCARD ignores the disk and any copy protection encrypted on it. Instead, WILDCARD takes a snapshot of memory in your Apple® II. Now you can make back-up copies of protected software with the push of a button.
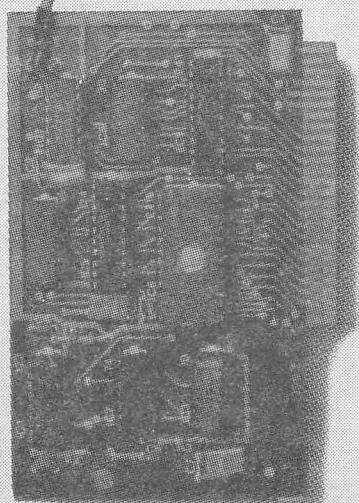
Features
- Hardware copying device...push button operation.
- Copies ALL* 48K memory resident software, most 64K software.
- No Parameters are necessary.
- WILDCARD lives in any slot.
- WILDCARD is undetectable by software.
- Produces autobooting disk in 2 minutes.
- Copies become accessible for alterations.
- Copies are DOS 3.3 compatible.
- Software and utilities included.

System requirements: Apple II Plus with 64K and DOS 3.3.
* Wildcard does not operate with CP/M® or other microprocessor based software.

$129.95 direct from East Side Software Co., 344 E. 63 St., Suite 14-A, New York City 10021, 212/355-2860. Please include $3.00 for handling. Mail and phone orders may be charged to MasterCard and VISA. N.Y. State residents add sales tax. Dealer inquiries welcome.

IMPORTANT NOTICE: The WILDCARD is offered for the purpose of enabling you to make archival copies only. Under the Copyright Law you, as the owner of a copy of a computer program, are entitled to make a new copy for archival purposes only and the WILDCARD will enable you to do so. The WILDCARD is offered for no other purpose and you are not permitted to utilize it for any other use, other than that specified.

Apple II is a registered trademark of Apple Computer, Inc. CP/M is a registered trademark of Digital Research, Inc. Omega Microware, Inc. Nibbles Away—trademark of Computer: applications.

The rate at which bits are written onto the diskette is NOT determined by the rotational speed of the diskette. Rather, this rate is determined by a machine language routine contained in DOS. In DOS, this routine is known as RWTS (read/write track/sector). In this routine, a very carefully written program (written to insure proper timing) determines the rate at which nibbles are presented to the disk controller. The bits contained in the nibbles are then written on the diskette at a rate determined by the Apple clock. Thus, if the disk speed is too high, the nibbles may not fit into the allotted space. If the speed is too low, more bits are crammed into less space on the track. Write errors can occur in either situation.

In contrast, the rate at which bits are read from the diskette is at least partially determined by the rotational speed of the diskette. A special-purpose chip in the Apple drive expects the bits read from the read head to occur at a rate that must meet fairly strict tolerances. If these tolerances are not met, both read errors and write errors (the correctness of the write cannot be verified) are likely to occur.

One element missing from this description is how the bit write rate is determined. If you inspect a standard 5 1/4" diskette for an Apple drive, a small hole about 1/4" in diameter can be found near the large central hole. If you carefully rotate the magnetic medium itself, you will discover that it too has a hole that is only visible when it is aligned with the hole in the cover. This is a timing hole: the speed at which the diskette turns in the drive can be controlled by noting how often the holes are aligned and adjusting the speed accordingly. Apple, Inc. does NOT use this timing hole. The rotational speed is determined solely by the amplitude of a voltage applied to the motor in the disk drive. For the technically inclined, this situation is termed an *open-loop system*; for the technically disinclined, the speed cannot be controlled automatically. One must adjust the voltage applied to the motor to obtain the proper diskette rotation speed (300 rpm).

Consequently, the only aspect of this rather complicated story that the typical user can monitor and adjust is the disk speed. If the speed is maintained too high (a so-called "hot" disk), the bits are not written at a higher speed and the controller may have trouble reading a diskette written in this way on the same drive! Because of the uncontrolled nature of disk speed, maintaining proper speed is entirely up to you.

Adjusting the speed is easy, but requires you to open the disk drive. If the drive is new, this procedure will void the warranty on the drive. In this case, it is probably best just to check the speed with a program and report any important misalignments to your dealer. If your drive is past the warranty stage, with the power off, turn your drive upside down and remove the four obvious screws. These screws hold the disk cover. Carefully turn the drive right-side up and slip the cover forward about half an inch. The cover will resist moving somewhat; this is normal. Look at the opening thus created on the right side of the drive. What looks to be the head of a small flat-head screw can be found mounted in the bottom of this opening. Turning this screw head with the proper screwdriver adjusts the resistance of a potentiometer, which determines the disk drive speed. Turning this screw a slight amount makes a significant change in the speed. One can now turn the power on and adjust the speed. Many Apple users drill a hole in the cover at the appropriate location to simplify access to this screw. The question remains — how does one know what is the proper speed?

There are two basic ways of determining the proper speed — one easy but potentially inaccurate while the other is harder but is guaranteed to be accurate. The latter method involves removing both the cover and the bottom from the drive. After removing the four screws described earlier, take the cover off entirely while the drive is upside-down by moving the cover toward the rear. Put the drive on its left side. Four more screws are revealed on the bottom of the drive which must also be removed. These screws hold on the bottom which once removed reveals the presence of a drive belt running between two wheels. The large central one has a 60 Hz timing wheel glued onto it. Shining a fluorescent light on this wheel while the drive is rotating a diskette will result in a pattern of light and dark lines. If this pattern is stationary (it no longer rotates), the disk speed is correct. The speed can thus be adjusted by turning the speed adjustment screw while the drive is on until a stationary pattern is obtained. The power company may not deliver the proper voltage to your home, but it makes very sure that the frequency of the voltage is accurately maintained at 60 Hz. Therefore, this method of speed adjustment is the most accurate, but requires you to open the drive completely. It therefore behooves you to be extremely careful if you use this technique. I would point out that food and/or drink do not mix well with an open drive.

The easier approach is to use a speed checking program. Such programs write a pattern of bits onto a track and, through software, determine the time it takes for the pattern to turn one revolution. Thus, one needs only to have access to the speed adjustment screw to set the proper speed. However, the software approach to timing is tricky and can be inaccurate. As we shall see, many of the speed checkers reviewed here are inaccurate and should be avoided.

When any of these speed checkers are used, one finds that they display in some fashion how the measured speed varies with time. Even if you are not turning the speed adjustment screw, a variation of speed will be reported by the speed checker. This is normal but should be small; the speed of a drive will vary with subtle mechanical and thermal variations (remember the open-loop control mentioned earlier). Consequently, adjust the speed so that the average is equal to the correct speed. If you encounter a large variation in speed, there may be a problem with your drive.

To assess the accuracy of speed checkers, I used the "hard" method to determine the proper speed while the programs were displaying their idea of what the proper speed should be. The results shown below were identical for both of my drives. In the descriptions that follow P means program name, A means what the program reported when the speed was indeed correct, H means what the program reported when the speed was at the largest allowable value, L when the speed was at the lowest allowed value, and C means a comment.

P: APPLE Cillin II
A: -11 units
H: +11 <-> +13 units
L: -42 <-> -3D units
C: This is a diagnostic?

P: Nibbles Away II
A: -6 <-> -B units
H: +1B <-> +1F units
L: -3A <-> -34 units
C: THE worst

P: Locksmith 4.1
A: +5 units on the hi-res screen
H: +20 units on the hi-res screen
L: -20 units on the hi-res screen
C: These are guesses as the program does not provide a scale.

P: Ultra Disk Pack I
A: 0 <-> +2 units
H: +20 <-> +22 units
L: -22 <-> -20 units
C: *REQUIRES* a initialized disk; tells you which way to turn the screw!

P: Disk II Speed Check
A: Right on!
H: -18 <-> -20 units
L: +22 <-> +24 units
C: from Apple, Inc.; *VERY* difficult to use.

P: A2 Utility Diskette
A: 199.83-200.04 msec
H: 198.40-198.57 msec
L: 201.45-201.60 msec
C: 200 +- 1.5 msec is the spec!

P: Disk Speed Test
A: -2 <-> 0 units
H: +18 <-> +20 units
L: -21 <-> -23 units
C: from the HAAUG Library (Volume 1A1)

An interesting aspect of these results is that the commercial programs are the most inaccurate. One of the most accurate speed checkers can be found in the HAAUG library. A notable exception among the commercial programs is the program from the A2 Utility Diskette. The speed checker (appropriately named 'SPEED') on this diskette is probably unfamiliar to most Apple users; it is provided by Micro-Sci when one purchases its Apple-equivalent disk drive. This speed checker works on Apple drives as well and is my personal favorite. A comment on all of the speed checkers except this one: the dimensions of the "units" are unspecified. The user has no idea if his speed is GREATLY off when the speed checker says it is off. The Micro-Sci utility clearly specifies the units of its measurement (milliseconds) and informs the user of the allowable variation of disk speed.

Because most speed checkers do not inform the user what they are measuring (speed or time for one revolution), I have included the readings from the various programs when the drive speed is set at its limits. My favorite (SPEED) and the Disk II Speed Check measure the time for one revolution (i.e., a greater number means a *slower* speed); the others measure speed. What the programs considered an accurate speed lay within the specified limits. I would prefer to use a checker that I knew was accurate.

GAME REVIEWS

By

Bill Muhlhausen

SOUTHERN COMMAND
PRODUCT OF STRATEGIC SIMULATIONS INC.

This game is a recreation of the Yom Kippur War crossing of the Suez Canal. The instruction book is very detailed and will require 30 to 60 minutes study before attempting to play the game.

SSI has presented the people who enjoy detailed board type war games a very entertaining program. The game may involve either 10 or 20 turns depending on the preference of the player or players. It is a one or two player game. In solitaire play you have four options as to level of difficulty. A solitaire game is faster to play than the two player game. In playing the game in both modes my associates and I found that we could not expect to finish a game in less than 5 to 8 hours. In fact 10 to 12 hour games were not unusual. The program does provide the participants the ability to save a game and come back to it. Our experience with this feature was erratic. Some games were easily saved while others were not. Though we were able to save the games for future play we were not able to restart them. Four attempts were made to restart a saved game but none of these were successful.

Some good features of this game are that terrain affects movement and combat effectiveness. Hidden movement is possible, hardened defenses for the Arab side are available, indirect artillery fire and ambushes are possible, and airstrikes may be made by the Israeli side. The Arabs can commit their airforce twice during the game to nullify a air strike. Units can be put into attack, defense, transport, river, or fire modes. You are able to reorganize battle weakened units. Artillery units generally may not be reorganized. Units are easily moved around the map. The map only displays a limited area around the unit to be moved. The map can be scrolled along the horizontal and vertical axis. This will give players a idea of the exact type and number of units in that area. However, you may not have a over all detail map of what kind and where all units you or your opponent have units in the area is available to the players. All graphics are in color and are well done. The game is played with keyboard commands only so no joystick is needed.

In play testing this game we found that it was hard to win playing the computer solitaire style. The Israeli player also has a hard time winning when playing another human player. To us this made the game well worth the price tag of $59.95 because it is not likely to become stale and boring to play after a short period of time. I feel that this is one of the best board simulation war games now on the market. Gamers that enjoy board type war games and those who think they might like to try this type of computer game will derive a great deal of fun playing SOUTHERN COMMAND.

U-BOAT COMMAND
PRODUCT OF SYNERGISTIC SOFTWARE

The object of this game is to sink the computer's navy. You have the option of naming your opponent and the country you are fighting for or allowing the computer to do so. Another option is the ability to name the enemy aircraft carrier and your sub or leaving it to the computer. There are four levels of difficulty. At each of these levels you have the option to play with or without enemy aircraft attacking you. To accomplish your mission you must first sink the enemy carrier. If the carrier escapes you lose. The enemy always has armed freighters and destroyers to intercept you as well as an aircraft option. Although you can miss your target the computer never misses your sub. You may fire torpedos from longer range than the computer can shoot. You must be careful not to sink your supply ships. Under these conditions shooting first can sometimes foul you up. The program has fine color graphics.

To run the program you need a joystick. Both the joystick and the keyboard are used in steering and fighting your sub. You have a sonar and a compass read out on your control panel. If your instruments have not been damaged you have a map overview as to where

the carrier & your sub are located. Your other opposition is not shown on the overview nor are your supply ships. You have the ability to turn your engines on or off, dive under water & surface, and send up a oil slick. In under water mode be careful to note the battery and air status on the control panel.

In playing this game I found that it was easy to learn the control commands but somewhat more difficult to coordinate steering and firing. In my opinion this game strikes a excellent balance between playability and realism. I felt that I was really the captain of the sub during play. I enjoyed the simplicity of operation as well as the many options that have been written in this game. The game could possibly become stale after you have sunk all the computer navies at the varying levels with and without aircraft. Most gamers however will get their money's worth in fun and enjoyment before this feat is accomplished.

MINGS CHALLENGE
PRODUCT OF MICRO FUN

Mings Challange is a shoot them up game with a twist. Death ships as well as bombs are dropped on the player. The death ships may be freed by the cyclops. The graphics are only average but are in color.

The game has 10 achievement levels. The first person to reach tenth level gets a message and, if that person is the first to notify Micro Fun, there is a award of 10 free game programs. The twist in this program is that you can set up your own game parameters for each level of play. This is a nice advantage if you have small children or others who are not as skillful as you or other frequent players. The game may be played with either the keyboard or a joystick. The sound may be turned on or off as you choose.

In conclusion I personally was unimpressed by this piece of software and would not want to buy a copy for my own use. Since my personal tastes and preferences may not be the same as some of you it is possible you might enjoy this game, particularly if you like shootem up games.
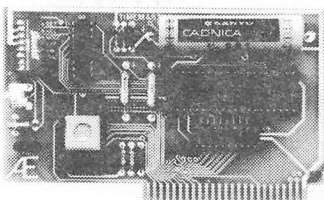

JIGSAW
PRODUCT OF MICROFUN

The product that all of you who enjoy puzzles and computers have been waiting for is now here. You can now solve jigsaw puzzles on your computer with this program. Not only can you solve pre-programmed pictures but you who have saved graphics pictures can use them for a puzzle of your own creation.

The instructions are fairly simple but it will take a little time to get used to using them. Those who have only a green screen or a black and white monitor will have a truly hard time completing a puzzle as the shapes

are not as distinct on the screen as they are in the cardboard version. Because of this colors are much more important in matching pieces than are shapes. The program creates a 180 piece puzzle which may be saved to disk and worked on in sessions, as your time allows.

The program seems to run well. This program is not recommended for those of you who do not enjoy puzzles. Those of you who do want something to do in your spare time will probably enjoy it.

The games reviewed above were made available for review by the Software Center of Houston at 2200 Southwest Freeway.

WANT/DON'T WANT ADS

For Sale: Mountain Hardware Introl X/10 home controller board. $75.

Format][Word Processor for Apple][, $125.

Found: Key ring and keys at January Saturday meeting.

Call Mike Kramer at 358-6687.

---===*===---

For Sale: Okidata Microline 82A Printer, 8 months old, perfect condition, $350 firm.

Call Chris Dede at 649-1569.

---===*===---

For Sale: Enhance your Apple][with Apple //e features. Videx Keyboard Enhancer][for $99 (Retail $149). Videx Videoterm Board for $199 (Retail $345). Videx Soft Video Switch for $19 (Retail $35). 16K RAM Board for only $49 (Retail $195). Excellent condition. Original cartons & documentation.

Call Bob Britton at 376-7525.

---===*===---

Wanted: Apple][owner's manual.

Call Buzz Sawyer at 960-1122

---===*===---

For Sale: Apple Graphics Tablet with manual and software.

Call Frank Schlesinger at 870-0168

---===*===---

For Sale: Apple][Silentype Printer, includes interface, documentation, and five rolls of paper. List price $413, sell for $275.

Heuristics Speech Lab, incoudes peripheral card, microphone, documentation, and 2 disks of programs. List price $200, sell for $125.

Call Dave Novak at 496-6996.

---===*===---

For Sale: Microsoft Z-80
Videx-compatible 80 Column Card
2-16K RAM Cards
Versacard (parallel,serial,clock)
More...Best offer.

Call Richard at 480-0981.

---===*===---

For Sale: Professional Easy Writer
Visicalc 3.3
PFS File
$50 each with documention.

Call Ruth Dill at 672-4317.

---===*===---

Wanted: Good used Disk][w/o controller.
Call Bill at 772-0323.

---===*===---

For Sale: Apple Pascal with manuals, virtually unused. $175 or offer.

Call Dale at 266-8360.

---===*===---

For Sale: Home Accountant, never used. $45.

Call Mitch Hanna at 790-0200.

---===*===---

For Sale: DB Master with Manuals.

Call Paul Brand at 464-4714.

---===*===---

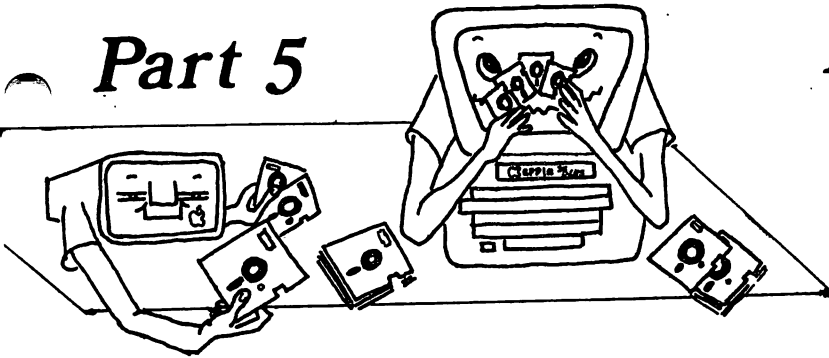For Sale: Apple][+, 4 months old. $800 firm.

Call John Stanley at 480-8987.

---===*===---

For Sale: Disk][Controller, DOS 3.3, $110 with manual, $100 without.

Call Ed Seegar at 723-6919.

---===*===---

Part 5

Dealing With DOS

By

Clark Johnson

For this issue we will deviate from the normal tutorial-type column on DOS and instead review some of the "fast DOS" packages being marketed today. These DOS enhancements could be very beneficial in a wide range of your computer applications. Not only should the reviews in this article be interesting to computer hackers like myself, but please note (!), several copy-protected business programs and home use programs can use a fast DOS to considerably speed up their operation. So if you are a businessman and interested only in applications, not utilities, read on.

Since I am rather wordy and could not condense this article down enough to fit in one APPLE BARREL issue, it will be broken up into two parts. This month we will cover how a fast DOS operates, which DOS's will be reviewed, and what are the criteria for selecting one particular DOS over another. Next month, the actual reviews and comparisons will be given.

It seems like almost overnight, there sprang up a wide selection of these fast DOS's, either significant re-writes of Apple DOS or just minor patches to the operating system. These programs had at least common objective — to speed up the time required to load a program from disk to memory.

Actually, the concept behind a fast DOS is fairly simple. Practically any "expert" Apple machine language programmer familiar with Apple DOS could write a basic fast DOS. It appears to have taken about three years before anyone came up with the revelation that Apple DOS is really very slow and could be sped up significantly. After that enlightenment, several authors wrote their own packages, although there are some appreciable differences in the value of their final products.

How does a fast DOS work? Basically, a fast DOS uses the same principle as the disk utility FID on the System Master disk. You will notice when using FID, that a long program can be copied from one disk to another in just a few seconds. If FID can perform its function so quickly — that of loading a program from disk to memory and then storing from memory to another disk — why does it take so long to simply load a program for a RUN or a BRUN?

The answer is in the way these two different loads are handled. FID loads entire sectors (256 bytes per sector) from a disk, while the DOS commands LOAD and BLOAD read in one byte at a time. FID is able to work in its manner because files on a disk always occupy whole sectors, even if the last sector is not completely filled with data. The fast DOS packages then follow this same principle by loading into memory, complete sectors (instead of individual bytes) until the last sector is reached. The last sector is then read into memory in the conventional manner, byte by byte, to avoid carrying "garbage data" into memory. (Garbage data would be zeroes or other worthless data in the unused portion of the last sector.)

Extra machine language programming inside of DOS is necessary to accomplish this task. There are several small "holes" or unused areas inside DOS. These areas have been used by practically all the authors who have written the fast DOS's. Usually, however, there areas are not large enough to hold all the required coding. Several authors have taken over the area where the INIT instructions were loaded, thereby disabling INIT. The limitations, if any, of the various fast DOS packages will be discussed in detail subsequently.

There is one other method to increase the speed of loading files that does not require the DOS to be modified. This involves changing the numerical order of sectors on all tracks of the disk. It has been established that in standard DOS 3.3, the order of sector numbering is inefficient. When DOS reads data from a sector on a disk, it needs a brief moment to digest that data before starting to read the next sector. Since the disk drive is constantly spinning the disk around, it is possible that the next sector needed by DOS is not close to the read head of the disk drive. This is the precise problem with the sectoring order used by DOS 3.3. By the time DOS is ready to read the next sector, that sector is "out of place". Therefore, DOS must wait until the revolution of the disk brings that sector back around to the disk drive read head. It is possible to change the order of sectors on a track so that by the time DOS needs the next sector, it will be close to the read head. At least one utility package (BAG OF

TRICKS by QUALITY SOFTWARE) has a program
(INIT) that will allow the disk to be
re-initialized with an improved order of
sector numbering and resulting improvement in
load times. This technique is mentioned in
this article since it does provide one
alternative to actually changing over to a
fast DOS.

When using a fast DOS, substantial speed
improvements can be realized. Many files can
be loaded or saved in 1/5 of the time
required by standard DOS 3.3. Speed
improvements for re-ordering the sectors are
not nearly as dramatic. This technique might
improve a load time to slightly faster than
60% of the normal time required. It should
be emphasized that the two different
techniques - fast DOS and sector re-ordering
- should not be used together. If used
simultaneously, the file transfer time will
be increased back to a longer time. This is
because the fast DOS's grab the data so
quickly that they would be slowed down if the
next sector were relocated elsewhere on the
track.

PACKAGES REVIEWED

The following fast DOS packages are
reviewed in this article. They represent
practically of the nationally - advertised
DOS enhancement packages. To my knowledge,
only one was omitted - TDE (The DOS EHANCER)
by S & H Software.

a) DAVID-DOS by David Data ($39.95)
   12021 Wilshire Blvd, Suite 212
   Los Angeles, CA 90025
   213-478-7865

b) DIVERSIDOS by DSR, Inc. ($30)
   5848 Cramptom Ct.
   Rockford, IL 61111
   815-877-1343

c) FAST DOS by Wytand P/L ($29.95)
   60 Gollan Ave
   Dundas, 2117, Australia

d) PRONTODOS by Beagle Bros. ($30)
   4315 Sierra Vista
   Santa Diego, CA 92103
   714-296-6400

e) SPEEDDOS by Softkey Publishing
   ($20)
   P.O. Box 44549
   Tacoma, WA 98444
   206-581-6038

f) TURBOCHARGER by Silicon Valley
   ($29.95))
   1625 El Camino Real
   Belmont, CA 94002
   415-593-4344

g) ULTRA FAST LOAD by Computer
   Advanced Ideas ($29.95))
   1442A Walnut Street
   Suite 341
   Berkely, CA 9470
   415-526-9100

CRITERIA FOR EVALUATION

As mentioned before, even though the
fast DOS's have common major objectives, they
have met these objectives with varying
degrees of success. Also, the packages
usually offer some bonus capabilities or
utilities in addition to the speed
enhancements.

These are the items I have selected as
criteria for evaluation.

1) Time required for a LOAD or BLOAD --
   This obviously also applies to a RUN or
   BRUN. This is the major goal of a fast
   DOS.

2) Time required for a SAVE or BSAVE --
   This is a completely different criterium
   from item (1) because some packages do
   not speed up "saves", and those that do
   have different speeds.

3) Time required for reading or writing Text
   files --

   Now we're really getting select. Only
   one package was able to speed up the
   normal reading and writing of Text files.

4) Compatibility with existing programs --
   Whenever you make modifications to DOS,
   you run the risk of making it
   incompatible with programs written for a
   standard DOS 3.3. Generally speaking,
   disk utility programs pose the greatest
   chance of incompatiblity.

5) Compatibility with copy - protected
   software --
   There are several protected programs that
   can be sped up by the use of a fast DOS,
   if it is compatible.

6) What are the methods of transferring the
   DOS to your disk? --
   The ease of transferring the fast DOS to
   your disks is important. The individual
   fast DOS's employ different means. There
   are three basic ways - a) creation from
   the master program disk, b) using the
   INIT command after any bootup on the fast
   DOS, and c) overlaying the DOS
   modifications on top of a standard DOS by
   running a specific file. Methods a) and
   b) are the most beneficial, especially
   when you may use either of the two for a
   particular DOS.

7) What bonus benefits are offered in the
   software package? --
   Besides the speed increase, what does the
   packages offer in other enhancements to
   DOS?

As stated, next month we will evaluate
each of the DOS packages considering the
criteria just stated. Summary conclusions
and recommendations will be given at that
time.

APPLE /// PEELINGS

By

Mike Kramer

CHANGING APPLE /// CP/M DRIVER FILE

INTRODUCTION

Although the procedure to add, change, or delete a driver under Apple /// CP/M is similar to that under SOS, it requires more steps and the steps required are not intuitively obvious. This discussion attempts to summarize those steps but is not intended as a substitute for the manuals.

Because of constraints of the CP/M system, the peripheral driver file is named DRIVER.SOS rather than SOS.DRIVER. As supplied by Apple, the CP/M system knows about the console, Silentype, and serial drivers but is totally ignorant of parallel devices. Your task is further complicated by the fact that CP/M only knows about four physical devices, consoles (CON:), tape readers (RDR:), tape punches (PUN:), and system printers (LST:). It permits up to four logical devices per physical device. Your driver file, however, can have as many drivers as you want within memory constraints. You just can't access all of them easily. A further complication is that apparently you can't look at the contents of the driver configuration blocks once the driver file is placed on the CP/M disk. You can see what drivers are included and which are assigned to the CP/M logical devices.

PREPARING THE DRIVER FILE

The first step in adding to or changing the CP/M DRIVER.SOS file is to boot the Apple /// SOS Utilities Disk. Since the procedure for preparing a driver file is well documented in Chapter 2 of the Standard Device Drivers Manual it will not be discussed here. The main point to remember is that you start the procedure under SOS, not CP/M and you should store the driver file on the CP/M Utilities Diskette under the name DRIVER.SOS. If you name it SOS.DRIVER you would overwrite the SOS.DRIVER file on the utilities diskette, which you may not want to do.

TRANSFERRING DRIVER FROM SOS TO CP/M

Insert the CP/M System diskette in the builtin drive (Drive A:) and the CP/M Utilities Diskette in the external drive (Drive B:) and boot the system. Next type SOSXFER to run the SOS to CP/M file translation program. After an asterisk appears type the following:

B:SOS.DRIVER=SOS.DRIVER

If you do not want to change the driver file on the CP/M System diskette, you will need to specify a different destination name on the left side of the "=" which would have to be used through the rest of the procedure described below.

After you specify the source and destination files and press RETURN the screen will then display:

    Insert SOS disk into drive A:
    Insert CP/M disk into drive B:
    Press RETURN to begin

When the transfer is complete, the message "Transfer complete" will be displayed.

For more detail on the use of SOSXFER see Page 24 of the Softcard /// Installation and Operation Manual.

CONFIGURING CP/M I/O

Replace the SOS disk in Drive A: with the CP/M System Diskette, press Control C to do a warm start, and run the configuration program by typing CONFIG. CONFIG will list the names of all the drivers in the file DRIVER.SOS along with the current CP/M device assignments. Likely the device assignments will show that CON:=CO1 and that CO1:=.CONSOLE. Also it will probably show that LST:=LS1 and that LS1:=.PRINTER. / If you named the parallel driver .PRINTER then you are all set. If you wanted a Silentype to be the system printer and it is listed as LS2: you would just type LST:=LS2:. When the device assignments are as you want them type END or a Control C. You will then be asked if the changes should be made permanent. If you answer YES they will be written back to the builtin drive.

See Page 52 of the Softcard /// Installation and Operation Manual for a sample display and detailed explanation of the use of the CONFIG utility.

TRANSFERRING DRIVER.SOS TO OTHER DISKS

Place any diskette you wish to have the new DRIVER.SOS file placed on into the builtin drive and leave the CP/M System disk containing the new DRIVER.SOS in the builtin drive. Press Control C to tell CP/M a disk was changed. If you don't, you will get an error when you try to write to it. Next erase the DRIVER.SOS file on the disk in the external drive with the following command:

    >ERA B:DRIVER.SOS

Next transfer the new DRIVER.SOS from the CP/M System disk with the following command:

    >PIP B:DRIVER.SOS=DRIVER.SOS

When this operation completes, you should have a disk with the desired I/O configuration, assuming that you did everything correctly. If you want the same I/O configuration on another disk, just PIP it.

UPGRADE TO 256K FOR $100

If your 128K Apple /// has two boxes on the bottom plate, one marked 96K and the other marked 128K, upgrade to 256K requires installation of a memory board at a cost about $900. If the blocks are labelled 128K and 256K, you can upgrade to 256K for approximately $100. An Apple /// of this sort has a 256K board with 128K of chips plugged in and 16 sockets for another 128K. The memory chips can be ordered from various sources who advertise in the personal computer magazines (Byte is a good bet) at about $6.25 each plus postage. Installation takes about 30 minutes and only involves plugging in the chips.

Submitted by Randall Swearingen

ON PREFIXES AND PATHNAMES

When the system is first booted, the diskette in the builtin drive becomes the default volume. This means that any disk access that does not include a drive identifier (.D1, .D2, etc.) or a volume name will look for the diskette that was in the builtin drive at boot-up. It can become very annoying to have to specify ".D2/" in front of every file name or in front of "CAT" when cataloging from Business BASIC. When the option is available to change the prefix, you should do it to preserve your sanity. Many of the commercial programs, such as Apple Writer ///, permit changing the prefix. This makes infinite sense, since you rarely will be saving files on the program disk which you booted. From Business BASIC, you can change the prefix by typing

PREFIX$ = /pathname

where "PREFIX$" is a modifiable reserved variable and "pathname" can be just the volume name of the diskette you want to be the default or can include one or more subdirectory levels. If you type

PREFIX$ = .D2

the system will read the name of the volume in drive 2 and set the prefix equal to that name. When you are in Business BASIC and you want to know what the prefix is, just type

PRINT PREFIX$

SPECIAL CONSOLE CONTROLS

In various places in the Apple /// manuals there is reference to the special console controls involving the control key and the 5, 6, 7, 8, and 9 keys on the numeric key pad. Volume 1 of the Business BASIC Manual suggests referring to the Apple /// Owner's Guide for a description of the functions. Have you tried to find where in the Owner's Guide? If it's there, I can't find it. The best description of these keys starts on Page 56 of the Standard Device Drivers Manual. To save you having to look them up I'll summarize them:

CONTROL 5 - Toggles the video output on and off. Data is still written to the console when the display is turned off, but it is not visible until CONTROL 5 is pressed again. The Apple /// will execute programs about 20% faster when it is not refreshing the video, so try CONTROL 5 next time you run Visicalc.

CONTROL 6 - Flushes all characters from the type ahead buffer. Lets you recover when your typing gets out of control.

CONTROL 7 - Suspends all screen output. If a program attempts to place new information on the screen, execution halts until CONTROL 7 is pressed again. This is used to pause catalog listings, screen printing in Apple Writer ///, listings of Business BASIC programs, etc.

CONTROL 8 - Causes control characters to be displayed as two letter abbreviations on the screen. Control codes sent to the screen will be ignored when in this mode. Potentially useful when trying to find out why the screen display gets blown away when running your latest whiz-bang program.

CONTROL 9 - Stops processing screen output without halting program execution.

MODULUS 11

CHECK DIGIT FORMULA

(A pro's programming technique, part 3.)

By
Dan Herron

In two past articles we have discovered why people use check digit mathematics on computer files. Check digits reduce errors and save time, especially when data is keyed during one session and processed at some other time against master files. The last article discussed Modulus 10 check digit formulas, and we discovered that your bankcard account number is a modulus 10 check digited number.

This article will discuss the Modulus 11 check digiting formula which is commonly known as the IBM Modulus 11 Algorithm. The reason to discuss this particular method is that keypunch machines have used it for many years, we have it working, and we are not about to change! This formula uses weight factors, product add and eleven complement. Remember the premise of check digiting: A base number has an extra digit, the check digit. This last digit is exactly predictable by using arithmetic on the base number.

CTI INFORMATION PROCESSING SUPPLIES, INC.
2802 LOUISIANA   526-9666

/ DATA/WORD PROCESSING
- SUPPLIES
- FURNITURE
- ACCESSORIES

CTI IS A HOUSTON BASED FIRM SPECIALIZING IN FULFILLING ALL YOUR WORD OR DATA PROCESSING NEEDS. **CTI** CAN SIMPLIFY YOUR ORDERING BY PROVIDING QUALITY WP/DP SUPPLIES, FURNITURE & ACCESSORY ITEMS FOR ALL TYPES OF SYSTEMS. ALL PRODUCTS ARE OF THE HIGHEST QUALITY AND FULLY GUARANTEED. FOR COMPATIBILITY QUESTIONS, OR TO ORDER CALL **526-9666.**

DISCOUNT PRICES

DATA PROCESSING SUPPLIES

- **Verbatim** DISKETTES
- CONTROL DATA DISK PACKS
- DISK CARTRIDGES
- PRINTER RIBBONS

- DISKETTE MAGAZINES
- MAG TAPE
- CONTINUOUS PAPER
- PRINTOUT BINDERS

WORD PROCESSING SUPPLIES

- **Verbatim** DISKETTES
- PRINTER RIBBONS
- PRINTWHEELS—METAL, PLASTIC
- HEAD CLEANING KITS & DISKS

- SPECIAL FORMAT DISKETTES
  (CPT, LANIER, LEXITRON, MICOM, NBI)
- MAG CARDS

▼IKING
SOUNDSHIELDS

ELIMINATE UP TO 90% OF OFFICE PRINTER NOISE. OVER 250 MODELS AVAILABLE.
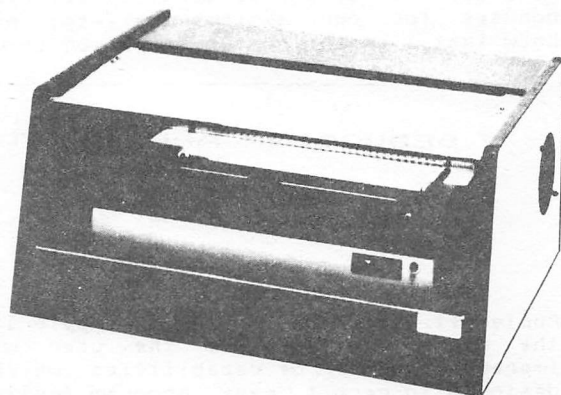
Let's explore an example of Mod 11 check digiting. Here are the steps...

1. Assign the weight factors to each digit position of the base number. The weight factors for Mod 11 are:
2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, 2, 3 ... starting with the units position of the number and progressing toward the high-order digit.
2. Multiply each digit by its weight factor.
3. Add these products.
4. Divide this sum by 11.
5. Subtract the remainder from 11.

Note: If the remainder in step 4 is zero, the check digit is zero.
If the remainder is one, the base number has no valid self-check digit; you must make sure that such base numbers are not used in the fields that you define as Modulus 11 self-check fields.

Here is a short example of Mod 11 check digit calculations.

For a base number of 1 2 3 4 5

Weight factor   x6 x5 x4 x3 x2

Multiply each digit   6 10 12 12 10
  by its wt.factor

Add these products   6 +10 +12 +12 +10 = 50

Divide sum by 11   50 / 11 = quot 4
                                rem 6
Subtract the remainder from 11: 11 - 6 = 5

This result is the check digit for the base number in the example. The entire "control field" for the number is 123455.

For a longer base number, assign the weights this way:
Base number   1 2 3 4 5 6 7 8 9
Weights   4 3 2 7 6 5 4 3 2.
Note that the sequence "repeats" beginning at 2.

I hope that this explanation of check digit formulas has been of use to you. In the real world where transactions may be keyed "offline" these techniques improve our accuracy and significantly reduce keying errors. The money we save with increased accuracy is of course spent on salaries and bonuses for our systems analysts; now you know that I am an innacurate person anyway.

DEBUGGING APPLESOFT

PROGRAMS

BY MIKE KRAMER

Applesoft BASIC* on both the Apple][+ and the Apple //e provides the user with an impressive number of capabilities which are designed to permit easy program testing and debugging. Unfortunately, there is no one place in the manuals that discusses the topic of debugging or how to use the tools provided. This article will discuss those capabilities and attempt to outline procedures which should help the relatively new Applesoft programmer to find program errors more quickly.

EXAMINING VARIABLES

One of the most valuable debugging aids provided by Applesoft is the ability to examine the last value of any variable in a program after its execution has been stopped for whatever reason. This permits checking the results of calculations, disk file inputs, or any other value or string used in your program. This is done by issuing (with program execution halted) an immediate command through the keyboard to print the desired variable or variables. If the variable does not exist in the program or it has not been assigned a value yet, it will have a value of zero if it is numeric or display no value at all if it is a string. This capability is actually a feature of the BASIC language, not Applesoft, so this technique will also work on your TRS-80.

As you will recall, changing the program in any way or issuing a RUN command will set all numeric variables equal to zero and will assign all strings a null value. This is great most of the time, since you can follow the poor programming practice of not initializing variables or including a CLEAR command at the beginning of your program. When you are debugging, however, it is absolutely essential that you do not add, modify, or delete a line in the program if you wish to look at the last value(s) of your variable(s).

At this point it is probably best to illustrate how to examine variables and attempt to show why this might be useful. First key in the following program and RUN it:

```
100 A = 10 : B = 5 : C = 2

110 A$ = "APPLE " : B$ = "COMPUTER"

120 D = A + B : E = B / C

130 C$ = A$ + B$

140 END
```

Next enter the following commands from the keyboard and see what happens:

```
] PRINT A,B,C,D,E

] PRINT A$,B$,C$
```

You should have seen a series of numbers representing the last values of the variables in the program. The results were fairly predictable, but it should be obvious that similar commands could be very useful in tracking down unexpected results in more complex programs.

Although the example above was fairly short, immediate commands are limited to 256 characters, the size of the keyboard input buffer and must not begin with a line number. In addition to PRINTing the last values of a variable, it is also possible to change the value of a variable from an immmediate command. The following example shows how all the values in an array can be checked and how a variable can be changed. First key in the following program and RUN it. Don't forget to clear out the earlier program by typing NEW.

```
100 DIM A(20)
110 I = 20
120 FOR J = 1 TO I
130 A(I) = I
140 PRINT "A(";I;") = ";A(I)
150 NEXT J
160 END
```

You should have seen a list of numbers on the screen starting with "A(1) = 1" and ending with "A(20) = 20".

Now type in the following immediate command:

```
] FOR K = 1 TO 20 : ? "A(";K;") =";A(K) :
NEXT K : I = 10 : GOTO 120
```

This should have resulted in listing of the 20 original values of A(I) followed by a short delay and a RUNning of the program with only ten values listed.

TRACE

The Applesoft TRACE command causes the line number of the line being executed to be written on the screen. This is most useful when a program seems to hang up an infinite loop or if the logic flow you intended does not seem to be followed. TRACE works without a problem if your program uses no DOS (PRINT D$...) commands. As pointed out on Page 44 of the DOS 3.3 manual, use of TRACE can cause problems when accessing a disk file. Try running a program that writes to a file with TRACE in effect! You should end up with program line numbers in the file. The DOS manual suggests that defining D$ = CHR$(13) + CHR$(4) will make most DOS commands work properly but doesn't elaborate.

If you follow the practice of stacking multiple program statements in a program line, remember that the line number will be written on the screen for each statement in the program line. If you find it difficult to read the line numbers as they flash past your eyes, you might want to use the SPEED= command from the keyboard to slow down program execution. A comfortable speed is about 150. Be sure to reset it back to SPEED=255 when through.

MONITORING DOS ACTIVITY

When you begin reading from or writing to disk files you will often wonder whether things are happening as you intended. You can easily watch the activity between the computer and the disk using the DOS MON command described on Page 42 of the DOS 3.3 manual. The MON command can be issued from the keyboard before running a program or can be issued from within the program in a PRINT D$ statement. You may optionally display on the screen the commands as they are issued, the data as it is written to the disk, the data as it is read from the disk, or any combination of the three. The early versions of Apple DOS defaulted to the MON state on bootup, so many programs such as FILE CABINET issue a NOMON command at the beginning. If you are curious about the structure of FILE CABINET files, change the NOMON to MON, set SPEED=150 and run the program. Many times it is useful to include a MONCIO statement in a program, as in my AUTOMATIC POKE MAKER in the February 1982 issue of the Apple Barrel.

PRINT STATEMENTS

A tried and proven method of debugging is the addition of PRINT statements to your program to print values of selected variables at various points in the program. In fact, adding WRITE statements is an essential debugging tool with compiled languages such as FORTRAN since they do not offer the tools you have with Applesoft BASIC. The use of diagnostic PRINTs can be effective but is time consuming, can introduce new errors, and requires surgery to remove after debugging is complete.

FLOW CHARTS

If you follow good programming practice, you probably flowchart your programs before writing or typing in any code, although it appears that some of the computer math teachers teach that flow charts are a waste of time. Unfortunately, it is easier to conceive an idea for a program and begin typing it in, so most of us tend to write a program and then wonder why it doesn't work. Often you can find the problem by taking the time to flow chart the faulty program. You don't have to be very fancy or follow any particular rules. Just draw blocks, label them to indicate the major function they represent, and connect them with arrows indicating the program flow. Don't attempt to write down every line of the program or you will quickly become tired of flowcharting.

SUMMARY

Software debugging is like any other troubleshooting skill, which is learned through experience. I have attempted to describe some of the tools and techniques and briefly how to use them. The rest is up to you.

_____
* All the techniques described above work with Apple /// Business BASIC except for the MON command.)

Houston Area Apple Users Group
APPLE BARREL
P.O.Box 610150
Houston, TX 77208

VOLUME 6, NO. 3, 1983

ADDRESS CORRECTION REQUESTED

THE H.A.A.U.G.