THE TARGET

JULY/DECEMBER 1981 -- an AIM65 newsletter

VIDEO

Yes, an issue of Target!!! After a long delay Target is available. On top of that, it is a big three issue bonus. Special emphasis has been placed on Video displays. Among other things, we have a review of the Video-1, an application program exemplifying the versatility of the Visible Memory, and a construction article to allow you to build your own video display. I want to thank all the contributers to this issue and especially to Dick Buchen. You may remember that Dick also provided the EPROM Programmer that was included in the September/October 1980 issue of Target.

AIM 65 EXPANSION

In the three years of Target I have occasionally touched on the topic of AIM 65 expansion. Most comments have been superficial. So at this time I will try to go a little deeper.

I define AIM expansion as adding hardware or software to add more features, capabilities, or otherwise make the AIM more than it was before. One way to expand the AIM is through software. In software we may add languages, utility programs, and of course application programs. The primary means for adding languages lies in adding ROMs to the AIM. Utility progroms may come through newsletters such as Target or from vendors. Application programs must essentially be created by the AIM user. Any further details on software expansion is better left to another time.

Now, let's consider hardware expansion. For hardware we may add RAM, EPROMs, I/O, or A/D or D/A boards. What physical method should be used to accomplish this goal?

The methods available are buying ready made products or building your own. You may also want to include a third method which would include a little of both of the above methods. This last method is the most effective. The end result will yield only the features that are needed. Nothing more and nothing less. Building your own will be the cheapest although not the easiest. The use of existing products will yield just the opposite.

At this time you may want to reference the March/April 1981 issue of Target for a list of suppliers that supply ready-made products for the AIM.

In the beginning there was the KIM-1 and the KIM-4. Since the KIM-4 was perhaps the first implementation of a bus for the 6502 processor and because it was also made by MOS Technology, the KIM-4 thus became a standard bus. A number of suppliers chose to use this bus in their own designs. However, some of them chose to change some of the signal definitions. In addition some chose to alter the card size or otherwise ignore the standard. Among the suppliers that are similiar to the KIM-4 bus include: Computerist, Hudson Digital Electronics, Micro Technology Unlimited, and Seawell Microsystems.

Another approach was taken by Rockwell which tended to support Motorola's Exorcisor bus. Applied Business Computers and Rockwell support this bus. A third bus is supported by Forethought Products which uses the S-100 (not used by the AIM-Mate series).

There is also a remaining assortment of suppliers which primarily revolve around the AIM 65 expansion connector.

As you can see there is quite an assortment of implementations. The one item that is common in most cases in the 22/44 pin connector. (Continued on page 35)

Video-1 Review

John Wahlquist
1643 N.Formosa Ave. Apt#4
Los Angeles,CA 90046

The VIDEO-1/1A is a multi-function expansion board designed specifically for the AIM-65. Its major functions include a color video interface, 28K of memory expansion, eight channels of analog-to-digital (A/D) conversion, sixteen channels of digital-to-analog (D/A) conversion, and a light pen interface. All of these functions are built onto a board the same width as the AIM and just slightly longer, a board in fact, that is designed to mount in any enclosure currently being made for the AIM.

Now you may ask, (and deservably so) why two numbers for this expansion board; and since you ask, and the answer pertains to the review, I'll tell you. The VIDEO-1 is the brainchild of Steve Rines and the Cedar Valley Computer Association, (a Rockwell company computer club). The VIDEO-1 was originally sold as a kit (parts, board, and, documentation) by the Cedar Valley Computer Association to members of various Rockwell computer clubs across the nation.

The rights to this board were then sold to the current supplier of the board, Microprocessor Products,Ltd. Some minor modifications (and one not so minor) were made to the board and it is now available as the VIDEO-1A. The major difference between the two boards is that the VIDEO-1 had a switching supply built in on the board to convert +24Volts to the ± 12 volts and -5 volts required by the A/D,D/A, and light pen sections, while the VIDEO-1A requires ± 12 volts(regulated). This only affects the three modules mentioned as all other functions require only ± 5 volts to operate.

A quick look at the VIDEO-1/1A reveals that the video section of the board is controlled by the Motorola MC6847 Video Controller chip. This chip is connected such that it can utilize up to 6K of memory (dedicated locations $8000 to $97FF) as video memory, outputing the contents of this memory to the RF interface where it is encoded into a signal compatible with television channels 3-6 (your choice).

There is a choice of 11 modes of operation; an Alphanumeric mode; 4 Semigraphics modes and 6 Graphics modes with resolutions ranging from 64X64 to 256X192, using up to 4 colors on the screen at once (up to eight colors for the semigraphics modes). The video signal is output as a standard RF signal or as a monitor compatible signal (you choose the wiring option).

The memory expansion section will hold up to 28K of type 2114 static RAM. RAM addresses start at $1000 and go to $7FFF. This allows for a 32K system with the video or a 38K system without the video (as VIDEO-1/1A RAM can also be used as system RAM). In addition to the RAM there is a spot for a 2716/2532 EPROM at $9800 to $9FFF.

The A/D section has eight channels of input to a digital converter with programmable gain/attenuation and offset. The sixteen lines of digital to analog output stem from two D/A converters, one unipolar and one bipolar, which when cascaded can yield up to 65,536 discrete steps. The light pen is capable of functioning in any of eleven video modes.

Now, how does it all work? In a word, "GREAT"! I built mine from a kit in my spare time over the course of about 3 weeks, and have been happily attached to a video terminal ever since. If you would like more information on the VIDEO-1A contact:

MicroProcessor Products,Ltd
2916 East Court Street
Iowa City, Iowa 55240
(319) 351-7587

Prices start at $159.00 for a semi-kit (board, documentation, and hard to get parts) and go up from there.

VIDEO-1 GAIM GRAPHICS SOFTWARE

JOHN WAHLQUIST

The GAIM Software Package is a graphics interpreter designed with the AIM-65/VIDEO-1 (multifunction video board for the AIM) system. The software comes burned into a 2716 EPROM designed to reside in the EPROM socket of the VIDEO-1 board ($9800 to $9FFF) and provides the user with 46 new graphics commands that are accessible from BASIC (thus making it an interpreter). GAIM is available from MicroProcessor Products,Ltd. for $30.00 postage paid.

Installation of the package is as simple as plugging the chip into the proper socket, but read the instructions anyway. In order to operate properly, the software needs RAM at locations $7E00 to $7F43, so be certain to install the top 1K of RAM on the VIDEO-1. Once this has been done then you can power-up to one of the best graphics packages on the market for an AIM based video board.

All of the graphics commands in the GAIM software package are executed via the USR function. A typical command would look like:

```
10 A = USR (30)
```

where A is a dummy variable (required to meet BASIC syntax requirements) and the number "30" is the function code to clear the screen. As an aside, it should be noted that all graphics commands are available for either immediate execution or as a part of a user program and the 46 commands will function in all sisteen graphics modes supported by the VIDEO-1 (16X46 = over 700 graphics commands).

The package functions entirely in machine language so the speed available is quite high. All variables needed by the machine language routines are taken directly from the BASIC variables thus eliminating the need to POKE data into machine language routines. If a function is called and there is a conflict of mode or a missing parameter (s), the package contains error sensing routines that will notify you of the fact (but beware, the error doesn't always halt your program and problems can occur with some commands as a result of this).

Due to the versatility of this package, a summary of all 46 commands would entail more writing than my fingers can take, therefore, below I have summarized the commands by the action they perform.

Now that you have somewhat of an overview you are probably interested in a user's comments. Well, here goes. I find the package easy to use and very useful. The documentation, in the form of a user's manual, is excellent (if all you want to do is use the software. If you want to know what makes it "tick" don't look for the information there, because its not given) and gives many examples of how to use each function separately and in conjunction with the others. I would strongly recommend this package to anyone who owns a VIDEO-1 board and is the least bit interested in graphics.

GAIM GRAPHICS COMMANDS

(1) POINTS

There are a total of 13 point plotting commands. They enable you to plot points in either color or on-off graphics modes. Additionally, points can be plotted conditionally depending on the present state (on-off or color) of the point to be plotted, erased or erased conditionally on the current state of the position, or examined and a flag set if the position is in a prescribed state.

(2) LINES

Lines can be drawn absolutely, with color specifications, or with color protection and erased similiarly. There are six commands controlling line functions.

(3) BOXES

The five functions available for drawing boxes behave very similiarly to the line drawing commands.

(4) REGIONS

Five commands are available for flooding a region (or erasing a region) with color either as an absolute command or as a conditional command dependant upon existing colors.

(5) SPECIAL FUNCTIONS

Six functions are provided to set the graphics mode, erase the screen, check for allowable "X" & "Y" values, and manipulate the "X" & "Y" variables used by the program.

(6) TEXT FUNCTIONS

There are six functions allowing Alpha/Numerics to be printed on the screen. Options available include horizontal or vertical writing, standard or reversed field, and the color options common to the previous functions.

(7) CHARACTER FUNCTIONS

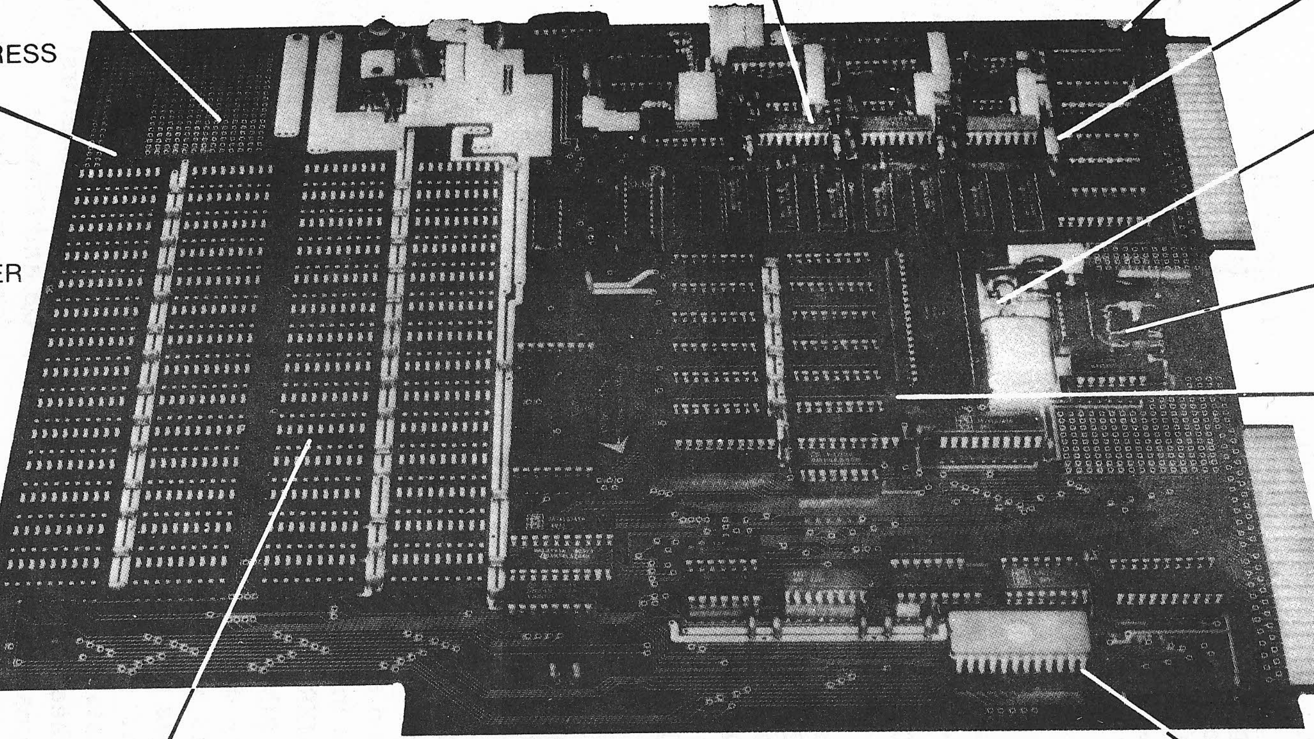
Design your own special characters and display them on the screen (Space Invaders here we come). Space in memory is allocated for up to 32 user characters which can be displayed either standard or reversed field, in color, and with the support of the color options. This mode is supported by five commands.

VARIABLES USED BY THE 46 COMMANDS

X0%,Y0%,X1%,Y1% - X & Y Positions
P0%,P1%,P2%,P3% - Flags, Colors, Mode, Font and Style depending upon function called.

COLOR GRAPHICS FOR THE AIM-65

Includes Software for CRT Handler!



Micro Processor Products
2916 East Court St.
Iowa City, Iowa 52240
319-351-7587
VISA / MASTERCARD ACCEPTED

GAIM SOFTWARE $30.00

AIM-65
MEMORY EXPANSION
BOARD $12.50

		Price
Video 1A-1	**Board, VDG and RF chips, 2 address ROMs, RF tuning inductor and cap, terminal strip for power, documentation. This is NOT a complete kit; other parts are required.**	**159.00**
Video 1A-2	**Assembled & tested video section, 1K RAM (sockets for 6K), EPROM socket, memory expansion logic, documentation.**	**279.00**
Video 1A-3	**Fully assembled and tested Video 1A Board with 6K Video RAM installed. Does not include extended memory – order below.**	**489.00**

3D Display

Osamu Asakawa
2-4-1 Taito, Taito-ku
Tokyo, Japan

It is difficult to imagine the graph of the function Z=f(x,y). This program displays the function upon Visible Memory using Keyword Graphics/Text software from MTU.

I. <u>To display the function</u>

(1) Define the function z=f(x,y) by using "DEF FNZ(X)=...." at the top of the program.

Example 1 Standardized two dimensional normal distribution

$$f(x,y)=\frac{1}{2\ \ 1-r^2}\ e^{-\frac{Q}{2}}$$

where $Q=(x^2-2rxy+y^2)/(1-r^2)$ and

0=r=1

will be defined as

```
R=0: REM R IS CORRELATION COEFFICIENT
DEF FNQ(X)=(X*X-2*R*X*Y+Y*Y)/(1-R*R)
DEF FNZ(X)=EXP(-FNQ(X)/2)/(6.283185*
                        SQR(1-R*R)).
```

See lines 1070 and 1090 of the program.

NOTE: The parameter C in line 1070 is the scale factor of Z axis. It is recommended that C=1 (not using the scale parameter) at the first trial. Modify C by the result of the first trial if required. Optionally if C=-1, the function will be displayed symetrically about x-y plane.

Example 2 $f(x,y)=x^2-y^2$ will be defined as

```
DEF FNZ(X)=X*X-Y*Y
```

II. <u>Speed Hint</u>

It takes 149.80 and 63.45 seconds to display Figures 2 and 3 respectively. (This is the time for GOSUB 4000 and GOSUB 5000.) We can make the program more than 20% faster by inserting

```
1142:
1144 K=0:K1=0:K2=0:XP=0:
     YP=0:DH=0:XD=0:XU=0
1146 XC=0: YC=0:C1=0:C2=0
     X=0:Y=0:XL=0:YL=0
1148 Z=0:MIN=0: MAX=0:X1=0:Y1=0
```

between lines 1140 and 1150.

III. <u>Adjustment</u>

Determine your FX in the line 1170. FX is the ration of X scale and Y scale to display the graph with correct scale.

(1) DEGREE OF Y AXIS 90
(2) FRAME (Y/N) Y
(3) measure domain on the display and correct FX.

ASMXRF

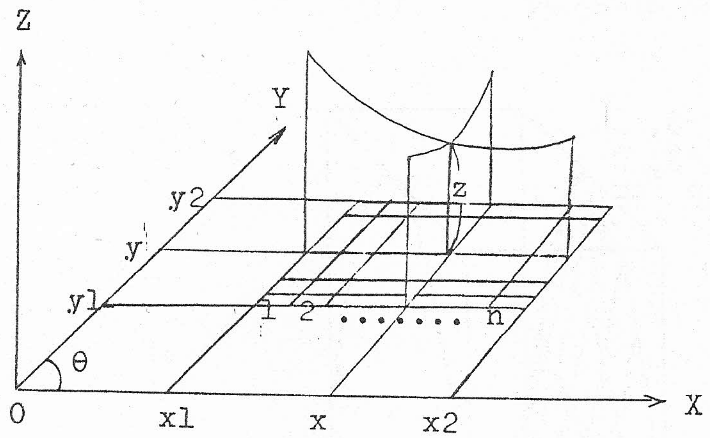
If you have a printer or terminal that can print 60 or more characters per line you can get full sized assembly listings just like the one in the monitor listing manual. ASMXRF reformats the LIST output from the AIM ROM assembler into a full size assembly listing. ASMXRF numbers all lines and compiles a complete cross reference of all labels in the program. ASMXRF also picks titles from .PAGE statements and puts them into the page headings. You can also enter an additional 30 characters to be printed in the page headings. You can select the line length and page size to match your printer. ASMXRF can even simulate paging on a TTY or other serial hardcopy terminal. For a sample of the listing produced by ASMXRF see the May/June 1981 issue of Target. If you are tired of reading AIM's 20 column assembly listing then try ASMXRF. ASMXRF runs on a 4K or greater AIM. It loads at \$200-\$AF9. Instructions included with either package:

ASMXRF object on cassette..........\$25
ASMXRF commented assembly listing. \$25

Prices include first class postage in the US and Canada. Add \$3 per order for foreign airmail. Check or money order US funds only. Order from:

Nehalem Bay Software
25730 Beach Dr.
Rockaway, OR 97136
(503) 355-8032

Message	Type in / Select	
DOMAIN (X1,X2)	x1,x2	See Fig-1
DOMAIN (Y1,Y2)	y1,y2	
DEGREE OF Y AXIS	θ	
NO OF MESH	n	
PRINT (L/H/N)	L...Hard copy using QKPRNT	
	H...Hard copy using QLPRNT	
	N...No print (default)	
FRAME (Y/N)	Y...Display frame	
	N...No display frame (default)	
DRAW (L/D/M)	L...Draw with line	
	D...Draw with dot	
	M...Draw with mesh (default)	
DETAIL (Y/N)	Y...Detail display (default)	
	N...Exit program	



$0 \leqq \theta \leqq 180$
$1 < n$

Fig-1

II. Source list and outputs of Ex.1 and Ex.2:
Both outputs are the result of the first trial.

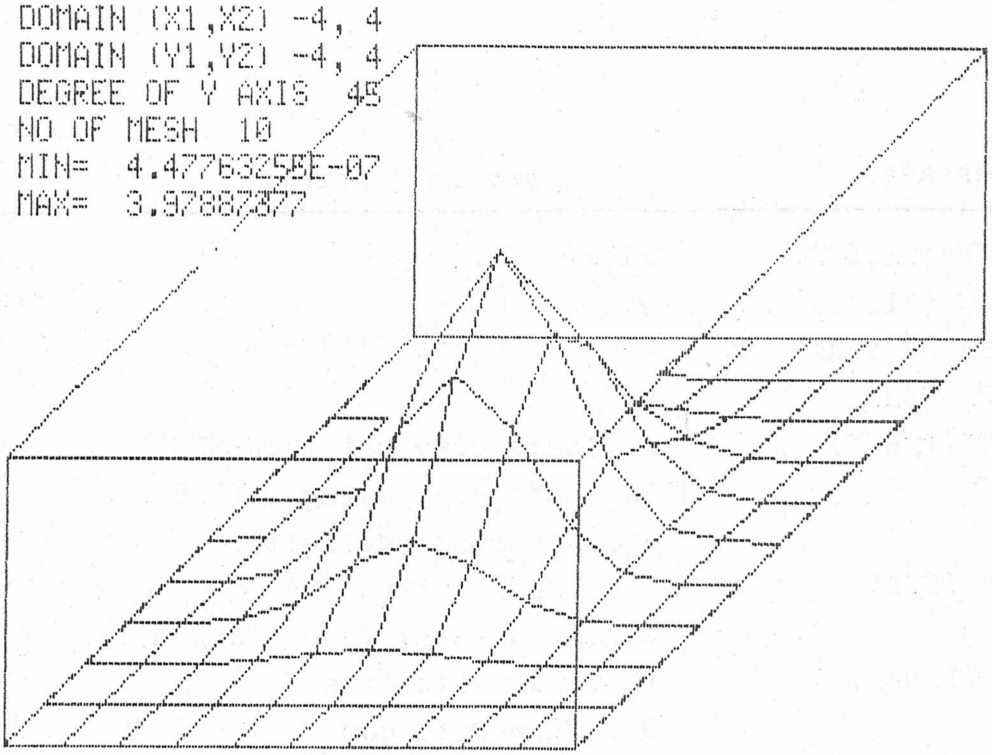


Fig-2 The first trial of Ex.1

```
LIST 1040-1100

 1040 :
 1050 REM      DEFINE FUNCTION Z=F(X,Y)
 1060 :
 1070 DEF FNZ(X)=X*X-Y*Y
 1100 :
```

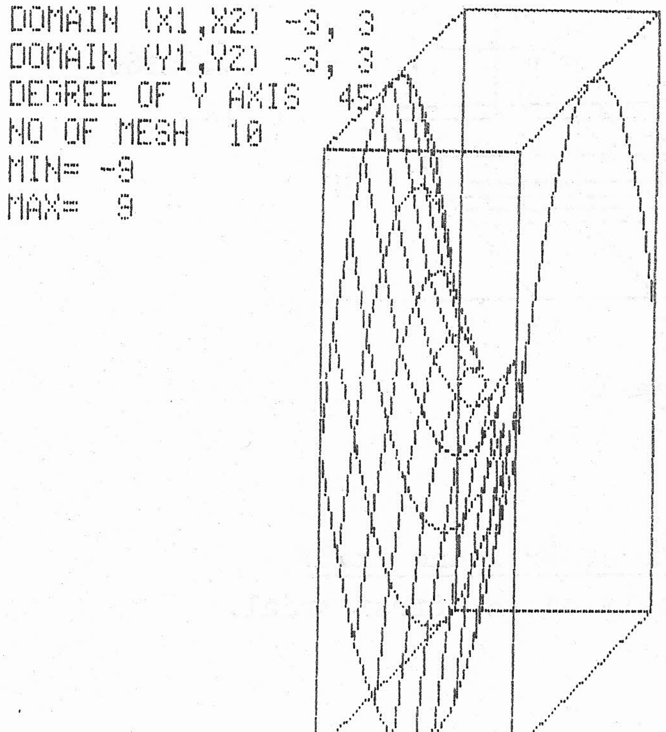


Fig-3 The first trial of Ex.2

```
1000 REM     3 DIMENSIONAL GRAPHICS
1010:
1020 REM     OSAMU ASAKAWA
1030 REM     JUNE 1, 1981
1040:
1050 REM     DEFINE FUNCTION Z=F(X,Y)
1060:
1070 R=0: C=25
1080 DEF FNQ(X)=(X*X-2*R*X*Y+Y*Y)/(1-R*R)
1090 DEF FNZ(X)=C*EXP(-FNQ(X)/2)/(6.283185*SQR(1-R*R))
1100:
1110 REM     INITIALIZATION
1120:
1130 ]X: NRMDSP: GMODE 1
1140 SETWIN 0,0,0,319,199: WINDOW 0: WCLR
1150:
1160 DIM HL(319),DM(3,1),PL(1,3,1)
1170 WW=319: WH=199: FX=1.2
1180 DG=45: MS=10
1190 SP=32: CR=13: DEL=127
1200:
1210 REM     MAIN ROUTINE
1220:
1230 AUTEXT "DOMAIN (X1,X2)";SP
1240   GOSUB 2000: X1=VAL(CH$)
1250   GOSUB 2000: X2=VAL(CH$)
1260   IF X2<X1 THEN GOSUB 3000: GOTO 1230
1270 AUTEXT CR
1280 AUTEXT "DOMAIN (Y1,Y2)";SP
1290   GOSUB 2000: Y1=VAL(CH$)
1300   GOSUB 2000: Y2=VAL(CH$)
1310   IF Y2<Y1 THEN GOSUB 3000: GOTO 1280
1320 AUTEXT CR
1330 AUTEXT "DEGREE OF Y AXIS";SP;STR$(DG);CR
1340 AUTEXT "NO OF MESH";SP;STR$(MS);CR
1350 RDCSR CX,CY
1360:
1370 INIT=1: FRAME$="Y": GOSUB 4000
1380:
1390 MOVE CX,CY
1400 AUTEXT "MIN=";SP;STR$(MIN);CR
1410 AUTEXT "MAX=";SP;STR$(MAX);CR
1420 RDCSR CX,CY
1430:
1440 DR$="M": GOSUB 5000
1450:
1460 MOVE CX,CY
1470 AUTEXT "PRINT (L/H/N)";SP: GOSUB 2000
1480   IF CH$<>"L" AND CH$<>"H" THEN 1510
1490     IF CH$="L" THEN GOSUB 3000: @KPRNT
1500     IF CH$="H" THEN GOSUB 3000: @LPRNT
1505     GOTO 1520
1510 AUTEXT CR
1520 AUTEXT "DETAIL (Y/N)";SP: GOSUB 2000
1530   IF CH$="N" THEN END
1540 AUTEXT CR
1550 AUTEXT "DEGREE OF Y AXIS";SP: GOSUB 2000
1560   DG=VAL(CH$)
1570   IF DG<0 OR DG>180 THEN GOSUB 3000: GOTO 1550
1580 AUTEXT CR
1590 AUTEXT "NO OF MESH";SP: GOSUB 2000
1600   MS=VAL(CH$)
1610   IF MS<1 THEN GOSUB 3000: GOTO 1590
1620 AUTEXT CR;"FRAME (Y/N)";SP: GOSUB 2000
1630   FRAME$=CH$
1640 AUTEXT CR;"DRAW (L/D/M)";SP: GOSUB 2000
1650   DR$=CH$
1660:
1670 WCLR
1680 AUTEXT "(X1,X2)";SP
1690 AUTEXT STR$(X1);",";STR$(X2);CR
1700 AUTEXT "(Y1,Y2)";SP
1710 AUTEXT STR$(Y1);",";STR$(Y2);CR
1720 AUTEXT "DEGREE OF Y AXIS";SP;STR$(DG);CR
1730 AUTEXT "NO OF MESH";SP;STR$(MS);CR
1740 RDCSR CX,CY
1750:
1760 GOSUB 4000
1770 GOSUB 5000
1780:
1790 GOTO 1460
1800:
2000 REM     INPUT PARAMETER
2010:
2020 CH$="": N=0
2030 GET C$: IF C$="" THEN 2030
2040 IF ASC(C$)=CR THEN AUTEXT SP: RETURN
2050   IF C$="," THEN AUTEXT C$;32: RETURN
2060     IF ASC(C$)=DEL THEN 2100
2070       N=N+1: CH$=CH$+C$: AUTEXT C$
2080 GOTO 2030
2090:
2100 IF N=0 THEN 2030
2110   N=N-1: CH$=LEFT$(CH$,N): AUTEXT C$
2120 GOTO 2030
2130:
3000 REM     DELETE MESSAGE
3010:
3020 RDCSR X,Y
3030 IF X>2 THEN AUTEXT DEL: GOTO 3020
3040 RETURN
3050:
4000 REM     DISPLAY PROFILE
4010:
4020 GRSHRT
4030 IF INIT=1 THEN GOSUB 4100
4040 GOSUB 4300
4050 IF FRAME$="Y" THEN GOSUB 4600
4060 ]X
4070 RETURN
4080:
4100 REM     PROFILE OF GRAPHICS
4110:
4120 MIN=1E38: MAX=-1E38
4130 DX=(X2-X1)/MS: DY=(Y2-Y1)/MS
4140 FOR I=0 TO MS: Y=Y1+I*DY
4150   FOR J=0 TO MS: X=X1+J*DX
4160     Z=FNZ(X)
4170     IF Z>MAX THEN MAX=Z
4180     IF Z<MIN THEN MIN=Z
4190   NEXT J
4200 NEXT I
```

```
4210:
4220 DM(0,0)=X1: DM(0,1)=Y1
4230 DM(1,0)=X2: DM(1,1)=Y1
4240 DM(2,0)=X2: DM(2,1)=Y2
4250 DM(3,0)=X1: DM(3,1)=Y2
4260 Z(0)=MIN: Z(1)=MAX
4270 INIT=0
4280 RETURN
4290:
4300 REM     SETUP PROFILE
4310:
4320 A=3.14159265*DG/180
4330 C1=COS(A): C2=SIN(A)
4340:
4350 X=X1: Y=Y1: Z=MIN: GOSUB 4900
4360 XL=XC: YL=YC
4370:
4380 X=X2: Y=Y2: Z=MAX: GOSUB 4900
4390 XH=XC: YH=YC
4400:
4410 W=XH-XL: H=YH-YL
4420 IF DG>90 THEN W=W-2*(Y2-Y1)*C1*FX
4430 IF W<=0 OR H<=0 THEN JA CR;"NON 3 DIMENSION": STOP
4440:
4450 FR=WW/W
4460 IF H/W>WH/WW THEN FR=WH/H
4470:
4480 XO=(WW-W*FR)/2+(320-WW)/2
4490 IF DG>90 THEN XO=XO-(Y2-Y1)*C1*FX*FR
4500 YO=(200-WH)/2
4510 RETURN
4520:
4600 REM     DISPLAY FRAME
4610:
4620 FOR I=0 TO 1
4630   FOR J=0 TO 3
4640     XC=(DM(J,0)+DM(J,1)*C1)*FX
4650     YC=Z(I)+DM(J,1)*C2
4660     PL(I,J,0)=(XC-XL)*FR+XO
4670     PL(I,J,1)=(YC-YL)*FR+YO
4680   NEXT J
4690:
4700   JM PL(I,3,0),PL(I,3,1)
4710   FOR J=0 TO 3
4720     JD PL(I,J,0),PL(I,J,1)
4730   NEXT J
4740 NEXT I
4750:
4760 FOR J=0 TO 3
4770   JL PL(0,J,0),PL(0,J,1),PL(1,J,0),PL(1,J,1)
4780 NEXT J
4790 RETURN
4800:
4900 REM     3-DIM TO 2-DIM CONVERSION
4910:
4920 XC=(X+Y*C1)*FX
4930 YC=Z+Y*C2
4940 RETURN
4950:
5000 REM     DISPLAY GRAPHICS
5010:
5020 NOCHK: GRSHRT
5030 FOR I=0 TO 319
5040   HL(I)=0
5050 NEXT I
5060:
5070 DX=(X2-X1)/MS
5080 DY=(Y2-Y1)/MS
5090 DH=DX*FX*FR/2+0.5
5100 XD=(320-WW)/2
5110 XU=319-XD
5120 A=3.14159265*DG/180
5130 C1=COS(A): C2=SIN(A)
5140:
5150 IF DR$<>"D" AND DR$<>"L" THEN DR$="M"
5160 IF DR$="D" OR DR$="L" THEN GOSUB 5300
5170 IF DR$="M" THEN GOSUB 5500
5180 JX
5190 RETURN
5200:
5300 REM     DRAW WITH DOT OR LINE
5310:
5320 FOR I=0 TO MS: Y=Y1+I*DY
5330   PT=0
5340   FOR J=0 TO MS: X=X1+J*DX
5350     XC=(X+Y*C1)*FX
5360     YC=FNZ(X)+Y*C2
5370     XP=(XC-XL)*FR+XO
5380     YP=(YC-YL)*FR+YO
5390     IF YP<HL(XP) THEN PT=0: GOTO 5440
5400     IF DR$="D" THEN JM XP,YP: JW: GOTO 5430
5410       IF PT=0 THEN JM XP,YP
5420       JD XP,YP: PT=1
5430     GOSUB 5800
5440   NEXT J
5450 NEXT I
5460 RETURN
5470:
5500 REM     DRAW WITH MESH
5510:
5520 DR$="L": GOSUB 5300
5530:
5540 FOR I=0 TO 319
5550   HL(I)=0
5560 NEXT I
5570:
5580 M1=X2: DM=-DX
5590 IF DG>90 THEN M1=X1: DM=DX
```

EPROM PROGRAMMER

The MDA-65 from Thorson Engineering will:

* Program both 2716 and 2532 type EPROMS. Transfers any part of AIM-65 memory to EPROM.
* Cause your AIM-65 to operate as a full duplex terminal. Allows communication with a time-sharing system or development system at speeds up to 2400 baud.
* Allow transfer of object files or BASIC programs to or from another computer via the terminal simulator.

The programmer circuit card, software in EPROM, and program listing all for only $135.00 check or money order. Terminal simulator software separately is $50.00. Washington residents add 5.3% sales tax.

Thorson Engineering Company
6225 76th St. SE
Snohomish, WA. 98290
(206) 334-4214

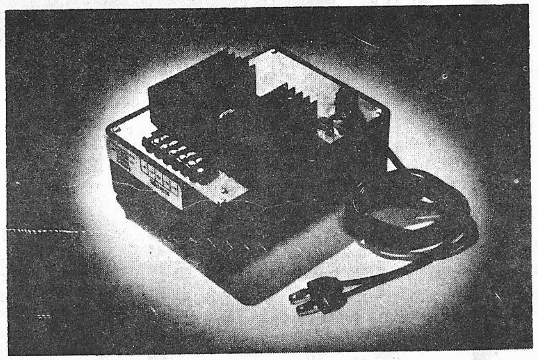
```
5600 FOR I=0 TO MS: X=M1+I*DM
5610   PT=0
5620   FOR J=0 TO MS: Y=Y1+J*DY
5630     XC=(X+Y*C1)*FX
5640     YC=FNZ(X)+Y*C2
5650     XP=(XC-XL)*FR+X0
5660     YP=(YC-YL)*FR+Y0
5670     IF YP<HL(XP) THEN PT=0: GOTO 5710
5680     IF PT=0 THEN JM XP,YP
5690       JD XP,YP: PT=1
5700     GOSUB 5800
5710   NEXT J
5720 NEXT I
5730 RETURN
5740:
5800 REM     UPDATE HIDDEN LINE CONTROL
5810:
5820 K1=XP-DH: IF K1<XD THEN K1=XD
5830 K2=XP+DH: IF K2>XU THEN K2=XU
5840 FOR K=K1 TO K2
5850   IF HL(K)<YP THEN HL(K)=YP
5860 NEXT K
5870 RETURN
```

Brochure A-1
Summer 1981

Micro Technology Unlimited
2806 Hillsborough Street
P.O. Box 12106
Raleigh, NC 27605, U.S.A.
(919) 833-1458

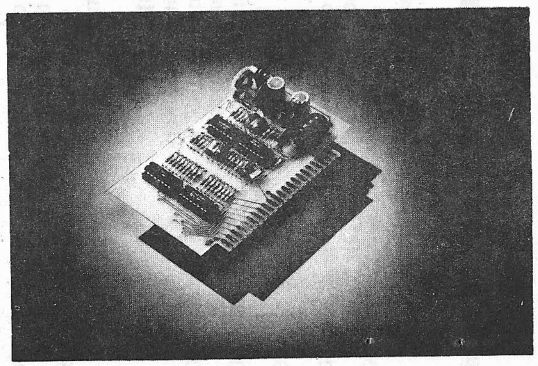
MTU EXPANSION PRODUCTS FOR THE AIM-65

K-1000-5 5 POWER SUPPLY $65.00



This power supply is designed specifically for the AIM-65 system. It is designed for table top operation and comes fully enclosed with a line cord, fused primary circuit and a terminal strip for connection to the DC power. The power outputs are: regulated +5 Volts at 3 Amps maximum (no other loads) 2 Amps with all other voltages loaded, regulated +24 Volts at 2.5 Amp maximum 0.5 Amp continuous draw, unregulated +8 Volts nominal up to 0.75 Amp, unregulated +16 Volts nominal up to 0.25 Amp. This supply will drive the AIM-65 and any three (typically four) of our bus expansion products.

K-1002 MUSIC HARDWARE BOARD: $49.00



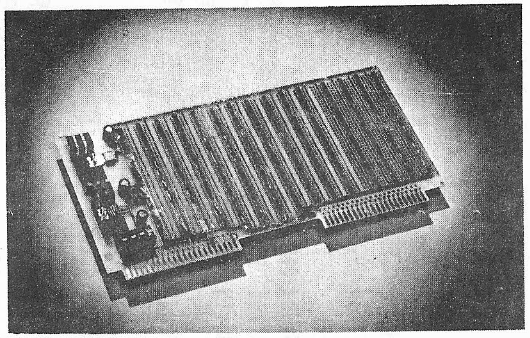
This board is a fully integrated digital audio system. An 8 bit digital-to-analog converter combined with a quality 6-pole low-pass filter and half-watt audio amplifier with volume control means that only a speaker is needed for rich, 4-part harmony with either of our music programs listed below. The board itself is approximately 5.5" wide by 3" deep and plugs into a 44 pin edge socket (not supplied) that may be mounted in a K-1005 card file. Only +5 Volts is needed for power and any 8 bit parallel output port is suitable as the data source. Don't confuse the K-1002 with cheap imitations, it is designed and built by MTU, the originators of digital music for microcomputers!

CODOS DISK OPERATING SYSTEM (supplied with K-1013-3D only) **K-1013-3M** $25.00

The AIM-65 version of the CODOS disk operating system provides features not available even on much larger microcomputer systems. CODOS provides an ideal environment for the assembly or BASIC language program developer and system implementer. Twelve character named files and nearly ½ megabyte storage on a single sided 8-inch diskette makes the manipulation of large amounts of data simple. Fast transfer rate of 19.6K bytes per second average makes the loading of massive programs or data files into memory nearly instantaneous. The channel-oriented I/O structure means that user programs need not distinguish between devices and files when performing sequential I/O. The built-in debugging monitor completely replaces the AIM monitor and adds many new functions as well. There is even a Visible Memory text/graphics driver routine provided free of charge to allow the inexpensive K-1008 Visible Memory to be used as a display terminal with 22 lines of 53 characters rather than the single line 20 character AIM display.

CODOS is fully integrated into the AIM's I/O structure. The AIM Editor, AIM Assembler, AIM BASIC, and even the AIM monitor may use disk files in place of cassette tape or memory files simply by specifying U as the device code when the "IN=" or "OUT=" prompt is printed by the AIM. The user then enters the up to 12 character file name and CODOS takes over from there. At this time BASIC support is limited to the program save and load functions built into the AIM BASIC ROM. The Supervisor Call facility provided by CODOS makes it possible for AIM assembly language programs to be transportable to other CODOS equipped 6502 systems. The CODOS Supervisor Calls provide for all device and file I/O as well as providing 16 bit arithmetic and number conversion functions needed by assembly language programs. Standard CODOS is available at 8000-9FFF. For optional address 6000-7FFF add $10.00.

K-1020 PROTOTYPE BOARD: $42.00



This board allows construction of custom circuits and comes assembled with on-board +5 and +12 Volt regulators, and bypass and input filter capacitors. It mounts in a K-1005-A Card File. Both bus Expansion and Application edge fingers are provided with gold plating and pads for connection. The board hole pattern allows 8, 14, 16, 18, 20, 22, 24, 28, 40, 64 pin IC's to be used, with a maximum capacity of seventy-five 14 and 16 pin chips. A heavy ground plane covers most of the board bottom for noise suppression while +5V and +12V planes intermesh on the top side. The board is constructed of standard glass epoxy material with all holes plated through. A documentation manual is provided containing example schematics for bus interface, address decoding, and form pages for your documentation.

K-1032-1 THE BANKER 32K RAM ROM I/O BOARD: $450 K-1032-1M $10.00

This is the 32K Memory Board with a difference. Not only does it have 32K of low-power dynamic RAM individually addressable in 4K blocks, it also has 4 ROM sockets for up to 16K of ROM, and 4 parallel I/O ports implemented with 6522 VIA's. It even has a EPROM programmer for 5 volt 2716 or 2732 EPROM's. The most significant feature though is **TWO** kinds of bankswitching right on-board which allows one to expand an AIM well beyond its normal 40K limit. The RAM and the ROM are divided into 4 blocks each (8 total) and an Enable Register determines which blocks respond thus allowing two or more blocks to be at the SAME address. Jumpers select which blocks are enabled by Reset. The Banker also recognizes 18 address bits which make it usable in future 18 bit address bus 6502 based systems.

K-1032 THE BANKER 16K RAM BOARD: $295.00

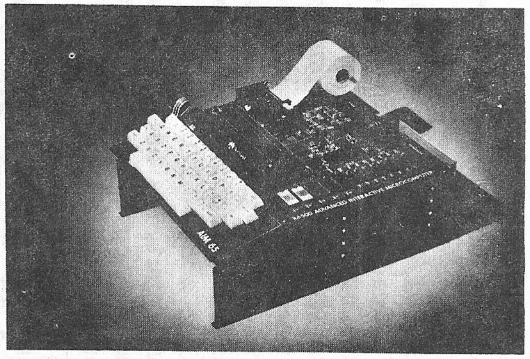
This is the same as the K-1032-1 Banker memory described above but with only 16K RAM installed and none of the ROM or I/O circuitry. It replaces our earlier K-1016 16K RAM board.

K-1002-8C INSTRUMENT SYNTHESIS MUSIC SOFTWARE: $49.00

This is by far the most powerful and flexible real-time 4 voice digital music synthesis program ever implemented on a microcomputer. Fundamentally this program is similar to the K-1002-5C music software except that the ability to specify an independent amplitude envelope for **EACH HARMONIC** of the tones has been added. Thus the harmonic structure of the tones **ACTUALLY VARIES** during individual notes in much the same way that conventional musical instrument sounds do. When coded instrument specifications are based on published analyses of common musical instruments, the resulting sounds will indeed closely resemble the analyzed instrument. The greatest power however comes from the ability to define original instrument sounds, and with the flexibility offered, quite a variety is possible. The manual supplied with the program contains an extensive discussion of sound synthesis principles and a library of over 20 different instrument specifications. The cassette contains the music program and three example songs, one for 8K, one for 16K and one for 32K machines.

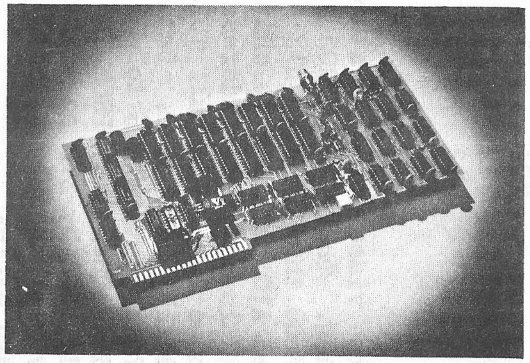
K-1005-A ASSEMBLED CARD FILE: $85.00, **K-1005M** $5.00

The K-1005-A Card File was designed to answer the need for system expansion of an AIM-65. The structure mounts the AIM above the frame with 5 expansion slots below it, 4 serviced by the backplane and the bottom slot with no connectors. Brackets extend the basic frame to mount the AIM keyboard at the 11.5 degree typewriter angle, in front of the display. A Bus Expansion Motherboard is supplied with gold plated connectors and a barrier terminal strip for power connections. This card file structure and expansion bus is what MTU has adopted as **THE STANDARD** and will continue to support as the system approach for the 6502 processor as the S-100 structure is for the Z-80. All MTU bus expansion products will work in this card file, allowing an AIM-65 to be upgraded to a truly powerful system. An application motherboard is optionally available.



K-1008 VISIBLE MEMORY BOARD: $240.00, **K-1008M** $10.00

This board is a unique concept in microcomputer memory systems. It is an 8K memory add-on to the AIM-65 which includes that circuitry to simultaneously display the memory contents as 320 by 200 dots on a CRT monitor. The board is designed for the AIM BUS and is merely connected in parallel with the expansion connector (some lines not connected). With this board, the processor runs at full speed with no wait states, and no software overhead or CPU time is required to refresh the display. There is no snow or other visible interference on the screen when the display memory is being accessed by the processor, or any other time.



K-1008-5C GRAPHICS/TEXT SOFTWARE $25.00

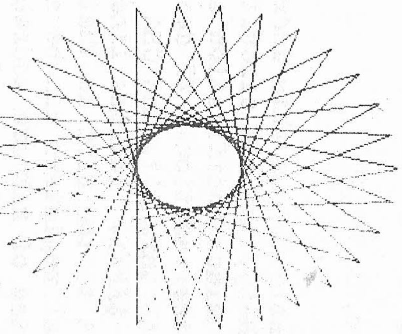
This is a package of several support programs for the Visible Memory for both machine language and BASIC programmers. The machine language subroutine library provides for plotting of points and lines given point coordinates or line endpoints as pairs of 16 bit integers. In addition, there are two text display routines for 22 or 18 lines of 53 characters each. An interface between AIM BASIC and the graphics routines is also provided so that the Visible Memory becomes BASIC's console with 53 character lines. Point and line plotting functions are called from BASIC programs by establishing values for X1%, Y1%, X2%, and Y2% variables and using the USR call facility. Text may be plotted by setting the cursor position with POKE's and then using normal PRINT statements. The console display functions are also usable by the AIM monitor and the AIM text editor. Links are provided for the K-1009-1C graphic print software which allows a printout of the Visible Memory screen contents at any time under BASIC control. The entire package is less than 2.5K and comes assembled for 4K and 20K AIM systems.

K-1008-9C KEYWORD GRAPHIC/TEXT SOFTWARE: $49.00

The K-1008-9C graphics software is optimized for versatility and ease of use by the BASIC programmer. When loaded into memory and enabled, over **40 COMMANDS** are added to AIM BASIC and the Visible Memory becomes a 22 line, 53 character console, which makes it much easier to write and edit AIM BASIC programs. A line may be plotted with a simple command such as: LINE 25, 85, 305, 167 and text with a command such as: CHAR "MARKET TREND". In addition, scaling and offsetting of coordinate values and clipping of offscreen data is supported. Up to 4 independent "windows" are maintained by the package with easy switching from window to window. The most significant new feature is "shape tables" similar to that on the Apple II computer but with the addition of relative vectors and subshape subroutining as well. There are even commands to cause a printout of the Visible Memory screen if the user has purchased the K-1009-1C Print Package. This software requires 7K and comes assembled for use with a 20K AIM.

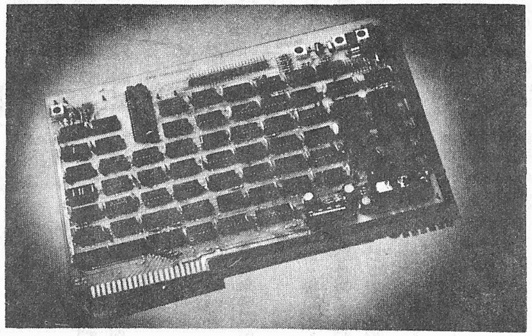
K-1009-1C GRAPHIC PRINT SOFTWARE: $35.00

This package consists of a Visible Memory screen print routine and a sideways text print routine. The screen print routine prints the 320 x 200 Visible Memory dot array DOT-FOR-DOT meaning that any kind of graphic image can be printed. The **QUICK PRINT** routine gives the entire image on one strip and the **QUALITY PRINT** routine gives it on two strips for proper proportioning. The **TEXT PRINT** routine can print the contents of the AIM text buffer (maintained by the AIM editor) SIDEWAYS in groups of 10 lines with line lengths up to 127 characters.



K-1013-3D 16K RAM FLOPPY DISK CONTROLLER WITH CODOS $595.00, **K-1013M** $10.00

This disk controller board, using the NEC uPD765 chip, was designed from the ground up for maximum speed and reliability in data transfer operations. Disk data transfers are done via direct memory access to the 16K of on-board memory without program intervention. Using our CODOS disk operating system, a 24K byte program can be located and loaded in 3 seconds, with a sustained data transfer rate of 19.6 KByte/sec. This allows speeds previously unattainable for date acquisition, data base management, etc.



The 16K of on-board read/write memory is split into two 8K blocks, independently addressable on any 4K boundary. One block is available for the user. The other 8K block can be write protected (under software control) and is normally used to hold the disk operating system software and disk data buffers.

The hardware allows operation with standard or mini-floppy drives, single or double density, single or double sided, and up to 4 drives with simultaneous seek. In addition, it contains 16K of on-board 2-port memory, initial program load ROM for turn-key systems, and complete MTU/KIM BUS compatibility in the MTU tradition. To use this product on the AIM-65 requires a power supply, K-1005-K Card File, 8" Floppy Disk drive, and disk cables.

The complete disk system including the CODOS operating software is designed to function with 8" double density, single or double sided Shugart model SA800 or SA850 8" or equivalent floppy disk drives. Use of mini-floppy or single density drives are not presently supported by MTU. The controller is sold with a diskette containing our CODOS, the K-1013-1M hardware manual and K-1013-3M CODOS manual. No drives or cables are supplied in this package.

AIM 65 Video

Dick Buchen 21 Alta Loma Drive Vallejo,CA 94590

INTRODUCTION

The AIM video board is used to interface the AIM-65 to a television receiver. The video board contains all the necessary hardware and software.

The video board can generate alphanumeric characters along with semigraphics, or can be used in one of eight full graphic modes. Depending on the mode selected, up to eight colors may be generated. A black and white or color television can be used.

Circuit Description

AIM/VIDEO INTERFACE

The video board interfaces to the AIM expansion connector J3. The connector numbers on the schematic correspond to the video board edge connector. All of the video signals are compatible with the AIM except the signal called ENB, which will be explained later. The following chart will show how to connect the video board to the expansion connector J3:

Name	J3	Name	J3	Name	J3	Name	J3
A0	A	A9	L	D0	15	CSA	20
A1	B	A10	M	D1	14	RES	7
A2	C	A11	N	D2	13	SYS Ø2	U
A3	D	A12	P	D3	12	SYS R/W	V
A4	E	A13	R	D4	11	SO	5
A5	F	A14	S	D5	10	GND	22
A6	G	A15	T	D6	9	VCC	21
A7	H			D7	8	+12	17
A8	K					-12	16

VIDEO EPROM

The program for the video board is stored in Z16, a 1K X 8 EPROM. The video EPROM is addressed from $7800 thru $7BFF. Z12 decodes address lines A10 thru A15 to chip select Z16 from $78XX thru $7BXX. Address lines A0 thru A9 connect to Z16 so all 1024 bytes can be individually addressed. Data bus lines D0 thru D7 connect to Z16 to obtain the program data when the EPROM is addressed. R/W is decoded by Z12 so that Z16, a read only device, can only be chip selected during a read cycle.

VIDEO RIOT

The AIM controls the operation of the video board using Z17, the video RIOT. RIOT stands for RAM,I/O, TIMER. The video RIOT is comprised of 128 X 8 static RAM, two software controlled 8 bit bi-directional data ports, a programmable timer, and a programmable edge detect circuit. The video RIOT is addressed from $AF00-$AFFF. To be able to use this address space on the AIM-65, the AIM circuit board will need a minor modification as described in another section. Z2 decodes address lines A8 thru A11, and CSA to chip select Z17 at $AFXX. Address lines A0 thru A7 connect to the RIOT so that RAM and internal registers can be addressed. The 128 bytes of RAM are addressed from $AF00-$AF7F. The addresses for other operations are as follows:

ADDRESS	FUNCTION
AF80	Read/Write Data Register A (DRA)
AF82	Read/Write Data Register B (DRB)
AF81	Read/Write Data Direction Register A(DDRA)
AF83	Read/Write Data Direction Register B(DDRB)
AF84	Write Neg. Edge Detect Ctrl Disable Int.
AF85	Write Pos. Edge Detect Ctrl Disable Int.
AF86	Write Neg. Edge Detect Ctrl Enable Int.
AF87	Write Pos. Edge Detect Ctrl Enable Int.
AF9C	Preset Timer Divide by 1 Enable Int.
AF9D	Preset Timer Divide by 8 Enable Int.
AF9E	Preset Timer Divide by 64 Enable Int.
AF9F	Preset Timer Divide by 1024 Enable Int.
AF94	Preset Timer Divide by 1 Disable Int.
AF95	Preset Timer Divide by 8 Disable Int.
AF96	Preset Timer Divide by 64 Disable Int.
AF97	Preset Timer Divide by 1024 Disable Int.
AF94	Read Timer
AF95	Read and Clear Interrupt Flags

RIOT I/O data ports PA0 thru PA5 are configured as outputs and are used to set up the various video modes of Z19. I/O ports PB3 thru PB7, configured as inputs, are used to read the on board dip switch. The video program reads the switch during initialization to configure Z19 to a particular mode. I/O port PA7, configured as an input, is used to read the field sync pulse coming from Z19. I/O port PA6, output, is used to control the operation of Z19's pin 31. When PA6 is low, Z19 pin 31 is low. When PA6 is high, data bus signal D7 controls Z19 pin 31. Other signals used by the video RIOT are R/W (pin 35), clock Ø2 (pin 39), and reset (pin 34).

VIDEO RAM

The video board can store up to 6K bytes of data . Twelve 1K X 4 static RAM IC's are used to accomplish this. The video RAM, Z21-Z32, is chip selected in pairs.

The odd numbered IC's connect to data bus lines D0-D3, the even numbered IC's to D4-D7. Each pair of IC's, for example Z21 and Z22, stores 1024 X 8 bits or 1K bytes of data.

Data to be displayed is strored in the video RAM by the video program. The AIM can read or write the video RAM, The VDG (Z19) can only read the RAM.

The video RAM is chip selected by either the AIM or the VDG. The video RAM is always being read by the VDG whenever the AIM is not reading or writing the RAM. Z13,Z10A, and Z7B decode address bus lines A11 thru A15 so that Z9E pin 10 goes low when the AIM addresses $6XXX thru $77XX. When Z14 pin 9 goes low, address lines A10 thru A12, connected to Z14's 'B' inputs, appears on its 'X' outputs. These address lines are decoded by Z15, a one of eight decoder, and one of Z15's outputs will go low to chip select a particular pair of video RAM ICs. Z9E pin 10 also goes to Z19 pin 12 to disable Z19's address lines A0 thru A12. Z9E pin 10 goes to Z5 pin1 and 15 to enable them so that the AIM's address lines A0 thru A9 drive the video RAM. When the AIM is not addressing the video RAM, Z9E pin 10 remains high, Z14 pin 9 will be high, and VDG address lines VDG10 thru VDG12 appear on Z14's "X' outputs to be decoded by Z15. Now the video RAM is chip selected by the VDG. Z19 pin 12 is high which enables Z19's address lines A0 thru A12. Z5 and Z6 are disabled so now the VDG addresses the video RAM.

The video RAM data is controlled by pin 10 on Z21 thru Z32. While the AIM is addressing the video RAM, Z7B pin 6 is high which enables Z11B. During a write operation Z11B pin 6 goes low when SYSTEM R/W goes low and SYSTEM Ø2 is high. During a read operation Z11B pin 6 goes high while SYSTEM R/W is high. When the AIM is not addressing the RAM,Z7B pin 6 remains low which causes Z11B pin 6 to remain high. The video RAM will now stay in the read mode irrelevant of SYSTEM R/W or SYSTEM Ø2.

ADDRESS AND DATA BUFFERS

The AIM address bus lines A0 thru A9 connect to Z16 and Z17 without buffering. A0 thru A9 are buffered by Z5 and Z6 before driving the video RAM. Address lines A8 thru A15 are connected to various decoders without buffering.

The AIM data bus, D0 thru D7, connects to two bi-directional data bus buffers, Z1 and Z3. Z1 is used to interface the video RAM with the AIM. Z1's buffers are enabled whenever the AIM addresses the video RAM (Z1 pin 19 goes low). The direction of data thru Z1 depends on the logic level at Z1 pin 1. During a write operation, Z1 pin 1 is low and data from the AIM is stored in the video RAM. A read operation brings Z1 pin 1 high and data from the video RAM is put onto the AIM data bus. Z3 is used to transfer data to and from Z16 and Z17. Z11C pin 8 goes high whenever the AIM addresses the video EPROM or the video RIOT. This in turn brings Z9B pin 4 low to enable Z3's buffers (Z3 pin 19). Z11C pin 8 also goes to Z7C where it is anded with SYSTEM R/W and SYSTEM Ø2. Z7C pin 8 goes high during a write operation which causes Z3's buffers to transfer data from the AIM to Z17. A read operation brings Z3 pin 1 low and data from Z16 or Z17 is put on the AIM bus.

VIDEO DISPLAY GENERATOR

Z19 is the Video Display Generator or VDG. This IC provides a means of interfacing the AIM to a color or black and white television receiver.

The VDG reads data from the video RAM when Z19 pin 12 is high. VDG's address pins A10 thru A12 are used to chip select the video RAM while address pins A0-A9 address the video RAM. The VDG's data lines D0 thru D7 read the data from the video RAM. Refer to the video RAM section for more details.

The VDG's television signals are generated on Z19 pins 9,10,11, and 28. These four analog outputs are used to transfer luminance and color information to a standard NTSC color television receiver. The VDG requires a 3.579545 MHz square wave on pin 33 to generate these television signals.

The synchronizing outputs from the VDG are on pins 37 and 38. Pin 37 has a high to low transition which coincides with the end of active display area, and a low to high transition which coincides with the trailing edge of the vertical synchronization pulse. During the time pin 37 is low, the AIM may have total access to the video RAM without causing undesired flicker on the screen. Pin 38 has a negative pulse that coincides with the horizontal synchronization pulse furnished to the television receiver by the VDG. Z19 pin 38 connects to Z5 pin 12. Z5 pin 11 is enabled whenever Z5 pin 15 goes low. This occurs during the time the AIM is addressing the video EPROM or the video RIOT. Z5 pin 11 pin 11 connects to the AIM's SO input which will set the CPU's overflow flag. This is used by the video program so that it knows when to write to the video RAM.

The remaining eight pins on the VDG are used as mode control input lines. Pin 34 switches between alphanumerics and semigraphics modes. Pin 31 determines if an internal or external character generator is used. Pin 35 switches between alphanumeric/semigraphic and full graphic modes. Pin 39 selects the color of the character or dot. Pin 32 selects the color of the character and background. Pins 27,29,and 30 select one of the eight full graphic modes.

A general description of the different VDG modes follows:

1. Alphanumeric internal mode: uses an internal character generator which contains 64 five dot by seven dot characters. Requires six bit ASCII code leaving D6 and D7 free. Displays 32 characters across and 16 characters down in one of two colors. Requires 512 bytes of video RAM
2. Alphanumeric external mode: uses an external character and row counter. Up to 256 different eight dot by twelve dot custom characters may be displayed in one of two different colors. Requires 512 bytes of video RAM.
3. Semigraphics four mode: Translates data bits D0 thru D3 into a rectangle (8 X 12 dots) divided into four equal parts. Data bits D4 thru D6 determine one of eight colors for the rectangle. The display consists of 64 elements across and 32 elements down. Requires 512 bytes of video RAM.
4. Semigraphics six mode: Translates data bits D0 thru D5 into a recatangle (8 X 12 dots) divided into six equal parts. Data bits D6 and D7 determine one of four colors for the rectangle. The display consists of 64 elements across and 48 elements down. Requires 512 bytes of video RAM. (D7 must remain high in this mode so only two colors are selected by the data bus)
5. Graphics ONE C mode: Generates a display matrix of 64 elements by 64 elements. Each element may be one of four colors. Requires 1K bytes of video RAM.
6. Graphics ONE R mode: Generates a display matrix of 128 elements by 64 elements down. Each element may be one of two colors. Requires 1K bytes of video RAM.
7. Graphics TWO C mode: Same as graphics ONE R except each element may be one of four colors requiring 2K bytes of video RAM.
8. Graphics TWO R mode: Generates a display matrix of 128 elements across by 96 down. Each element may be one of two colors. Requires 1.5K bytes of video RAM.
9. Graphics THREE C mode: Same as graphics TWO R except each element may be one of four colors requiring 3K bytes of RAM
10. Graphics THREE R mode: Generates a display matrix of 128 elements across by 192 elements down. Each element may be one of two colors. Requires 3K bytes of video RAM.
11. Graphics SIX C mode: Same as graphics THREE R except each element may be one of four colors requiring 6K bytes of video RAM.
12. Graphics SIX R mode: Generates a display matrix of 256 elements across by 192 elements down. Each element may be one of two colors. Requires 6K bytes.

RF MODULATOR

Z20 is a color video modulator. It takes the luminance and color information from Z19 pins 9,10,11, and 28 and generates the RF signal for the television receiver. Z20 pin 12 is the RF output that connects to a receiver VHF input. A 3.58 MHz crystal connects to an oscillator inside Z20 on pins 1 and 2. The exact frequency of 3579545 is adjusted using the 9-35 pF trimmer capacitor at Z20 pin 2. The frequency of the RF signal is determined by the components connected to Z20 pins 13 and 14. The 0.1 uH coil can be expanded or compressed to adjust the RF output for VHF channels 2,3,or 4. Z20 supplies the clock (pin 1) used by the VDG (pin 33) to generate the luminance and color information.

The video board may be connected to the video portion of a television receiver thereby bypassing the VHF tuner and IF circuits. The video signal from the VDG can be connected to the television video amplifier. The video signal is obtained from Z19 pin 28 through a 0.01 uF capacitor. Be sure the television receiver does not have a "hot" chassis, in other words, one side of the AC line input must not connect to the chassis. Make sure the receiver uses a power transformer or is a battery operated portable.

POWER SUPPLY CONNECTIONS

All of the ICs are connected from +5V to GND. The video board uses approximately 500 mA with 1K of video RAM installed.

The -12V is regulated down to -5V by three terminal regulator Z18. The -5V is connected to Z16 pin 19.

MISCELLANEOUS CIRCUITS

The ENB signal generated on the video board may be used to enable bi-directional data bus buffers. This pin can be ignored when no additional data bus buffering is required. Z7A pin 12 goes high whenever the video RAM, video EPROM, or video RIOT is addressed. Z8A pin 2 goes low when SYSTEM Ø2 and Z7A pin 12 are both high.

AIM ADDRESS MODIFICATION

In order to use the software and hardware described in this article as written requires an AIM address change. The directions for implementing this change is included elsewhere is this article.

Software Description

PRELIMINARY COMMENTS

The MC6847 Video Display Generator is primarily a video graphics IC with minimal character mode capability. With adequate programming, however, it provides an excellent alternative to the AIM's LED display while you are developing your graphic programs.

This program is designed specifically to interface with the AIM-65 monitor program. Much effort was given to overcome short-comings of the AIM monitor itself, while still providing a fast, convenient alternative to the LED display.

FEATURES

1. Display 32 X 16 characters
2. Flicker-free operation
3. Fast software scrolling
4. Horizontal tabs (programmable)
5. CRT type deletes (even works on tabs)
6. Stand-alone subroutines for user programs
7. Completely self contained (uses on-board RAM)
8. Compact (less than 1K of code)
9. Direct keyboard control (CNTLR,C,V)

All "writes" to the video RAM are synchronized with the horizontal sync pulse. This allows an entire screen worth of characters to be written during one frame. Since the horizontal sync pulse is only about 8 usec (including the back porch), it is necessary to respond almost instantaneously with the write instruction. This is accomplished by wiring the sync signal to the 6502's overflow pin and testing for overflow, followed immediately by the WRITE instruction. The result is an apparent high BAUD rate and rapid, flicker-free scrolling.

During the AIM tape dumping and loading, a block count is displayed by the AIM monitor. Although the AIM manages to keep this count in the same position on its own display, the video program sees a sequence of ASCII characters, like any other, and so would display them across the line and on to the next line, and so on. The problem here is that the time required for scrolling can interfere with the timing of the tape load routine, causing errors. Similiar display trickery is performed elsewhere in the AIM monitor, and needed information is not passed on to the video routine. The video program has been written to take care of these problems.

SOURCE LISTING

The source listing is included here and contains many comments so a detailed description will not be given on the software.

AIM ADDRESS MODIFICATION

Introduction

The AIM allocates address space $A000 thru $AFFF for I/O devices. This address space is decoded into four 1K blocks as shown below:

ALLOCATED ADDRESS	DEVICE	BYTES REQ.
$A000 to $A3FF	User Via(Z1)	16
$A400 to $A7FF	AIM RIOT(Z33)	152
$A800 to $ABFF	AIM VIA (Z32)	16
$AC00 to $AFFF	AIM PIA (U1)	67

As shown, the devices do not require all of the address space allocated. When the address modification is complete, the address space is decoded into four 512 byte blocks as shown below:

ALLOCATED ADDRESS	DEVICE
$A000 to $A1FF	USER VIA (Z1)
$A400 to $A5FF	AIM RIOT (Z33)
$A800 to $A9FF	AIM VIA (Z32)
$AC00 to $ADFF	AIM PIA (U1)

This modification frees up the following 512 byte blocks of address space:

$A200 to $A3FF
$A600 to $A7FF
$AA00 to $ABFF
$AE00 to $AFFF

Description

Refer to the AIM schematic for the following description. The modification replaces the crystal oscillator Z14 with a low power schottky device. This prevents address bus loading when an unused gate in Z14 is connected to address bus signal A9 (Z14 pin 9).

A 4.7K ohm pull-up resistor connected to Z19 pin 1 is removed and pin 1 is connected to inverted bus signal A9 (Z14 pin 8).

Implementation

The modification is implemented by following the steps shown below:

1. Un-solder and remove Z14 from the P.C. board.
2. Install a 14 pin dip socket in place of Z14.
3. Plug a SN74LS04N IC into Z14's socket.
4. Cut P.C. trace going to Z19 pin 1.
5. Connect 30 AWG wire from Z14 pin 8 to Z19 pin 1.
6. Connect 30 AWG wire from Z14 pin 9 to Z9 pin 18.

This completes the modification.

BOOKS

6502 Assembly Language Programming by Lance Leventhal from Osborne/McGraw-Hill, $12.50 plus $1.00 handling (non-US or CAN $4.00 handling).

Take Aim-Volume I by James H. Clark from Dilithium Press $14.00 plus $1.00 handling (non-US or CAN $4.00 handling).

This lab and learning manual for the AIM-65 and other 6502 microcomputers is a valuable tool for students and practitioners. Manual includes computer precations, programming basics, a glossary, a cross-index of AIM-65 documentations. Also included are more than 30 utility and game programs.

Back Issues- A consolidated 1979 issue is available for $6.00 ($12). In addition 1980 issues are available beginning with the January/ February and at subsequent two month intervals. Individual 1980 issues are $1.00 (US and CAN, $2.00 elsewhere).

Time to Renew- The mailing label contains the last issue that you will receive. If no date appears you have at least two issues left on you subscription.

Target- an AIM 65 newsletter is published bimonthly with an annual subscription rate of $6.00 in the US and Canada, $12.00 elsewhere. Contact Donald Clem, RR# 2, Spencerville,OH 45887.

CONSTRUCTION TECHNIQUES

The video board should be built on a prototype P.C. board. The board can be wired using wire-wrap or solder techniques. Number 30 AWG wire can be used to wire the ICs but the +5V and GND wires should be 24 AWG or larger. The P.C. board will have to be about 10 inches by 6 inches to accomodate all the parts.

Sockets should be used for all of the ICs. This makes it easy to check out the wiring before inserting the ICs. Check out the board a section at a time, and replace any defective parts.

To help filter noise out of the +5V supply, 0.1uF capacitors should be located on about every third IC. Connect the capacitor from +5V and GND. These capacitors are not shown on the schematic.

The RF connection from modulator Z20 to the television receiver should be 75 ohm coax cable. Be sure to use a 75 ohm to 300 ohm converter at the VHF input to the receiver.

The video board can connect to the AIM expansion connector J3 by using a cable between the video board and the AIM or by attaching a card edge connector to the video board and plugging the video board directly onto J3.

When additional boards are connected to the AIM bus signals, buffering will be required. The buffering would take place between J3 and the additional boards. The address bus can be buffered using SN74LS SN74LS367N ICs. The data bus can be buffered using one SN74LS245N which is enabled by connecting its pin 19 to the ENB signal generated on the additional boards. Also, SYSTEM R/W and SYSTEM Ø2 should be buffered.

Parts List

INTEGRATED CIRCUITS

Z1,Z3	SN74LS245N
Z2,Z12,Z13	SN7430 or SN74LS30N
Z4	SN74LS08N
Z5,Z6	SN74LS367N
Z7,Z11	SN74LS10N
Z8	SN7416N or SN7406N
Z9	SN74LS04N
Z10	SN74LS20N
Z14	N8233N (Signetics)
Z15	SN74LS138N
Z16	2708 (Intel)
Z17	SYP6532 (Synertek)
Z18	uA79M05AUC (Fairchild)
Z19	MC6847P (Motorola)
Z20	MC1372P (Motorola)
Z21-Z32	P2114L (Intel)

SPECIAL PARTS

1. (1 ea) 3.58 MHz crystal (used with MC1372P)
2. (1 ea) Trimmer capacitor 9-35 pF
3. (1 ea) 0.1 uhenry coil, 5 and 1/2 turns of 20 AWG enameled wire, 1/4 inch in diameter, 3/8 inch long.
4. (1 ea) 5 position dip switch
5. (1 ea) 10K ohm P.C. mount variable
6. (2 ea) 40 pin dip socket
7. (1 ea) 24 pin dip socket
8. (2 ea) 20 pin dip socket
9. (12 ea) 18 pin dip socket
10. (4 ea) 16 pin dip socket
11. (10 ea) 14 pin dip socket

GENERAL

1. All resistors are 1/4 watt 5% composition
2. All polarized capacitors are 20Vvtantalum
3. All non-polarized capacitors are mica or ceramic disc

RIOT PORT PROGRAMMING

VDG MODE	PA0	PA1	PA2	PA6	D7
ALPHA INT	1/0	1/0	0	X	0
ALPHA EXT	1/0	1/0	0	*	0
SEMI 4	X	X	0	0	1
SEMI 6	X	1/0	0	1	1

X= don't care
PA3=PA4=PA5= don't care

* The alphanumeric external mode can't be selected because Z19 pin 31 must be high when D7 is low, and Z11A prevents this.

VDG MODE	PA1	PA2	PA3	PA4	PA5
GRAPH 1C	1/0	1	0	0	0
GRAPH 1R	1/0	1	1	0	0
GRAPH 2C	1/0	1	0	1	0
GRAPH 2R	1/0	1	1	1	0
GRAPH 3C	1/0	1	0	0	1
GRAPH 3R	1/0	1	1	0	1
GRAPH 6C	1/0	1	0	1	1
GRAPH 6R	1/0	1	1	1	1

PA0=PA6=BIT D7= don't care

— AIM 65 —

P/N		QTY 1-9
A65-1	AIM-65 w/1K RAM	$399
A65-4	AIM w/4K RAM	$439
A65-A	Assembler ROM	$ 85
A65-B	BASIC ROMS	$100
A65-PL	PL/65 ROMS	$125
A65-F	FORTH ROMS	$150

SPECIALS

A65-4AB	**AIM-65 w/4K RAM, Assembler & BASIC**	**$599**
A65-4B	**AIM-65 w/4K RAM, BASIC**	**$529**

Power Supplies (AIM-65 Compatible, Industrial Quality Open Frame)

PRS3	+5V at 3A, +24V at 1A w/mtg hardware, cord, etc	$ 75
PRS4	+5V at 2A, +24V at .5A w/mtg hardware, cord, etc	$ 60
PRS5	+5V at 2A, +24V at .5A, ± 12V to ± 15V at .4A	$ 75
PRS6	+5V at 3A, +24V at .5A, + 12V at 2A	$ 85

From The Enclosure's Group

ENC1	AIM-65 case	$ 45
ENC1A	AIM-65 case w/space for one expansion bd	$ 50

Cases With Power Supplies (for ENC1A Add $5)

ENC3	ENC1 w/PRS3 mounted inside	$125
ENC4	ENC1 w/PRS4 mounted inside	$110
ENC5	ENC1 w/PRS5 mounted inside	$125
ENC6	ENC1 w/PRS6 mounted inside	$135

From Optimal Technology

ADC1	A/D eight channels, D/A 2 channels, requires ± 12V to ± 15V at 100 MA & 2-I/O Ports from AIM-6522	$126
	w/Cable for AIM-65	$150

From The Computerist

MCP1	Mother Plus™ Dual 44 pin mother card & card cage, fully buffered, 5 expansion slots underneath the AIM	$150
MEB1-2	DRAM Plus™ 32K RAM, 16K PROM sockets, 2-6522 I/O chip and programmer for 5V EPROMS	**16K RAM $325** **32K RAM $395**
PTC1	Proto Plus II™ Prototype card same size as KIM-1, MEB1-2, VIB1 (Bare Bd $60) assembled	$ 75
VIB1	Video Plus™ bd with 128 char, 128 user char, up to 4K display RAM, light pen and ASCII keybd interface	$325
CBL1	CABLE for MEB1-2, VIB1, PTC1	$ 25

NEW PRODUCTS! **CUBIT** featuring a 4½x6½ AIM-65* for $195
*less monitor, printer, display & keybd
APPLIED BUSINESS COMPUTER featuring
64K RAM Board - $495
80 column Video Controller - $325
5¼" Disk Controller w/ADOS™ - $485

From Seawell MicroSystems

MCP2	Little Buffered Mother™ Single 44 pin (KIM-4 style) mother card. Has on bd 5V regulator for AIM-65, 4 expansion slots. Routes A&E signals to duplicates on sides with 4K RAM	$199
MEB2	SEA 16™ 16K static RAM bd takes 2114L with regulators and address switches	$250
MEB2-3	CMOS RAM, realtime clock, EPROM bd, up to 8K RAM, 16K EPROM. (w/1k CMOS, 7K NMOS)	$395
PGR2	Programmer for 5V EPROMS with ROM firmware, regulators, low force sockets, up to 8 EPROMS simultaneously, can execute after programming	$299
PIO2	Parallel I/O bd with 4-6522's	$260
PTC2	Proto/Blank™ Prototype card that fits MCP2	$ 49
PTC2A	Proto/Pop™ with regulator, decoders, switches	$ 99
FDC2	Floppy disk controller bd & DOS, up to four 8" drives, double sided, double density (DD/DS)	$425
SBC2	SBC/CPU card, 9K RAM, 18K EPROM, 3 serial ports, 1 parallel port, audio tape interface	$495

From Micro Technology Unlimited (MTU)

DAC3	8 bit DAC bd	$ 49
FDC3	Floppy disk controller bd & DOS, up to four 8" drives, double sided, double density, 16K DRAM, Boot PROM	$595
MCP3	Card file w/4 slot expansion mother bd w/keybd brackets	$ 85
MEB3-2	Banker Board™, low power, 32K DRAM	$450
PIO3	24K PROM, 4-8 bit I/O ports w/RS-232 port to 4800 bps, PROM Programmer	$295
VIB3	8K DRAM bd, low power, w/composite video out in 200 lines 320 dot/line format	$240
PTC3	Prototype Bd w/regulators	$ 42
MPS3	AIM-65 Power Supply w/12V for MTU Bds, can drive one 8" disk drive	$ 65
CBL3	Cable for MEB3-2, VIB3, FDC3	$ 25

All MTU Software Available For These Products

Miscellaneous

TPT3	Approved Thermal Paper Tape, 3/165' rolls	$7.50
MEM6	6/2114 RAM Chips	$ 45
CAS1	Audio Cassette Recorder	$ 40
CAS1-1	CAS1 w/cable	$ 65
2716	16K 5V EPROM	$ 10
2532	32K 5V EPROM	$ 35
A65-P	Printer	$ 70
A65-DM	Display Module-DL1416	$ 30
FDD8	QUME Data Trak 8, DD/DS Disk Drive/up to 1 Megabyte	$650
FD8C-1	Cable set for 1 drive w/AC cord	$ 50
FD8C-2	Cable set for 2 drives w/AC cords	$ 65
MON1	9" composite video monitor w/80 char line resolution. Requires 12V DC only at .8A	$125

All AIM-65 Spare Parts Are Available
All "Microflex" Products Available
COMING SOON!
AIM-65/40 w/32K RAM- $1395

ASSEMBLED & TESTED SYSTEMS

We have been specializing in assembled and tested systems made from the above items for over 2 years. Normally, the price will be the total of the items, plus $5 for handling; shipping is extra on all COD's or invoiced orders. Please call or write for exact prices or if questions arise. **Six month warranty on all systems.**

Higher quantities quoted upon request, COD's accepted, shipping will be added. Add $5 for shipping, insurance, and handling on prepaid orders. Minnesota residents add 5% sales tax. Prices subject to change without notice.

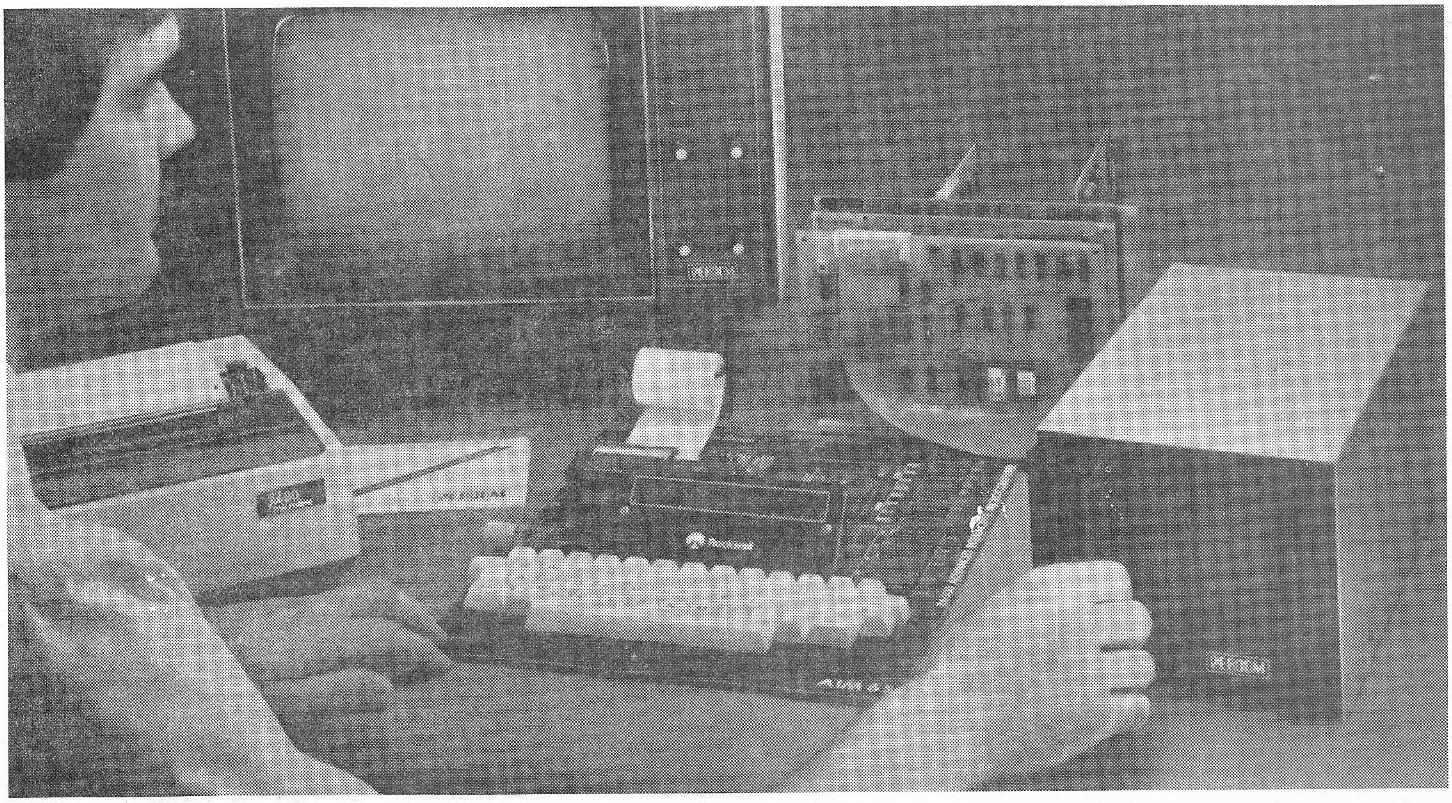
July 1, 1981

Mail Check or Money Order to:

Educational Computer Division
EXCERT INCORPORATED

- SALES
- SERVICE
- INSTALLATION
- CONSULTING

P.O Box 8600
White Bear Lake, MN 55110
(612) 426-4114



Introducing the M line . . .

Now! Drive Systems for AIM, KIM and SYM Computers – from PERCOM.

Assembled and tested systems start at only $599.95, including the drive controller circuit card, disk-operating system, interconnecting cable, drive and comprehensive users manual.

- **The right storage capacity** – Available in 1-, 2- and 3-drive systems, with either 40- or 80-track drives.
- **Flippy storage** – Flippy drives (optional) let you flip a diskette and store data and programs on the second recording surface.
- **High Storage Capacity** – Formatted, one-side storage capacity is 102 Kbytes (40-track drive), 205 Kbytes (80-track drive).
- **Proven Controller** – The drive controller design is the same as the design used in the Percom 680X LFD mini-disk system. This system – introduced in 1977 – has given reliable service in thousands of applications. Two versions are available: the MFD-C65 for the AIM-65 expansion bus, and the MFD-C50 for the System-50 (SS-50) bus.
  - Includes an explicit data separator circuit that's reliable even at the highest bit densities.
  - Provides for on-card firmware.
  - Includes a motor inactivity time-out circuit.
  - Capable of handling up to four drives.
  - Capable of reading both hard- and soft-sectored diskettes.
- **DOS included** – The MFD disk-operating system works with the AIM monitor, editor, assembler, Basic and PL/65 programs, interface is direct, through user I/O and F1, F2 keys. Diskette includes DOS source code and library of 20 utility commands.
- **Reliability assurance** – Drives are burned-in 48 hours, under operating conditions, to flag and remove any units with latent defects.
- **Full documentation** – Comprehensive hardware and software manuals are included with each system.

Now! Expand your AIM-65 with Low-cost System-50 Modules.

The Percom M65/50 Interface Adapter connects your M-65 bus to Percom's System-50 (SS-50) motherboard, allowing you to expand your AIM, KIM or SYM with proven System-50 modules. You can add disk storage, memory modules,even a video display system. The M65/50 provides buffer-amplification of address, data and control lines. On-card decode circuitry lets you allocate address space either to the computer or to the expansion motherboard. Price: only $89.95, *including* System-50 motherboard.

System Requirements: AIM-65, KIM or SYM computer with expansion bus and four Kbytes RAM (min).

Yes . . . I'd like to know more about Percom MFD drive systems. Rush me free literature.

Send to
PERCOM DATA COMPANY, Inc., Dept. 65-T
11220 Pagemill Rd. Dallas, TX 75243

name

address

city state

zip phone number

MAIL TODAY!

PERCOM

PERCOM DATA COMPANY, INC.
11220 PAGEMILL RD. DALLAS, TX 75243
(214) 340-7081

Toll-Free Order Number: **1-800-527-1222**

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

```
;
;                         VECHO
;
;
;             VIDEO ECHO PROGRAM
;
;
;        **** HARVEY CRISLER ***
;
;
;                 23 OCT. 1980
;
;
;     1. 19 JUL 80     ORIGINAL
;     2.  8 OCT 80     MINOR CORR.
;     3. 23 OCT 80     ADDED MORE COMMENTS
;
;
;     USE OF DIP SWITCH ON VIDEO BOARD:
;
;     S1    S2      BKGND    CHAR
;
;     OFF   OFF     GNG      BLK
;     ON    OFF     BLK      GNG
;     OFF   ON      GRN      BLK
;     ON    ON      BLK      GRN
;
;
;     VIDEO CONTROL CHARACTERS:
;
;     ^C = CLEAR SCREEN
;     ^R = REVERSE COLORS
;     ^V = ENABLE/DISABLE VIDEO
```

```
;
;
UIN    =     $0108   ; USER INPUT VECTOR
BLK    =     $0115   ; TAPE BLOCK COUNT
;
;
;     AIM KEYBOARD PORTS (6532):
;
DRA2   =     $A480   ; DATA REGISTER A
DRB2   =     $A482   ; DATA REGISTER B
DLINK  =     $A406   ; AIM DISPLAY VECTOR
DIBUF  =     $A438   ; AIM DISPLAY BUFFER
;
;
;     AIM MONITOR SUBROUTINES
;
PHXY   =     $EB9E   ; PUSH X & Y ONTO STACK
PLXY   =     $EBAC   ; PULL X & Y FROM STACK
CREAD  =     $FE83   ; READ CURSOR
RED2   =     $E976   ; ECHO CHAR FROM KB
PSLS   =     $E7DC   ; DELETE ROUTINE
KEPR   =     $E970   ; OUTPUT MESSAGE, INPUT CHAR
MON    =     $E182   ; MONITOR
OUTDS  =     $EF05   ; DISPLAY ROUTINE
;
;
;     AIM MONITOR VARIABLES
;
INFLG  =     $A412   ; INPUT DEVICE
OUTFG  =     $A413   ; OUTPUT DEVICE
CRPO2  =     $A415   ; DISPLAY POINTER
MNJMP  =     $A47D
```

```
;
;
;     6532 RAM (VIDEO BOARD) AF00 - AF7F
;
;                .OR  $AF00
;
;
;
;     THE FOLLOWING ROUTINE IS COPIED INTO RAM BY THE
;     "VCOPY" ROUTINE. THE OPERAND FIELD OF THE STA
;     INSTRUCTION SERVES AS A POINTER INTO THE
;     VIDEO RAM, AND IS MODIFIED BY THE PROGRAM AS
;     THE CURSOR POSITION CHANGES.
;
VPUT   =     $AF00   ; CLV
HWAIT  =     $AF01   ; BVC HWAIT
VPOS   =     $AF03   ; STA $6000
;                    ; RTS
;
CRPOS  =     VPOS+01      ;CURSOR POSITION
;
;
;     THE ROUTINE BELOW IS COPIED INTO RAM BY THE "SCROL"
;     ROUTINE. THE OPERAND FIELDS OF THE LDA AND STA
;     INSTRUCTIONS SERVE AS POINTERS FOR THE SCROLLING
;     FUNCTION AND ARE MODIFIED BY THE PROGRAM AS SCROLLING
;     PROGRESSES, LINE BY LINE. NOTE THAT THE LDA AND STA
;     INSTRUCTIONS FIT WITHIN THE 8USEC WINDOW.
;
UPLIN  =     $AF07   ; LDX #31
UPLN1  =     $AF09   ; CLV
UPLN2  =     $AF0A   ; BVC UPLN2
UPLN3  =     $AF0C   ; LDA $6020
UPLN4  =     $AF0F   ; STA $6000
;                    ; DEX
;                    ; BPL UPLN1
;                    ; RTS
;
LINEL  =     UPLN3+01     ;LOWER LINE POINTER
LINEU  =     UPLN4+01     ;UPPER LINE POINTER
;
;
CURLN  =     $AF16
XPOS   =     $AF17
YPOS   =     $AF18
VMODE  =     $AF1B
SAVCR  =     $AF1C
SAVEY  =     $AF1E
LENTH  =     $AF1F
```

```
;
;
;     THE FOLLOWING 10 BYTES ARE THE HORIZONTAL TAB LOCATIONS,
;     STARTING WITH THE RIGHT-MOST TAB. THE VALUES SHOWN
;     ARE THE DEFAULT VALUES WICH ARE LOADED IN DURING
;     INITIALIZATION (SEE "DFTAB"). THESE VALUES WERE
;     CHOSEN TO MATCH OUR ASSEMBLER FORMAT, AND MAY NOT
;     SUIT EVERYONE.
;
;                  .OR  $AF21
;
AF21  322D28231E
                  TABS  .BYT 50,45,40,35,30
AF26  1914110B07
                        .BYT 25,20,17,11,7
;
;
;
;
;     RIOT REGISTERS
;
  DRA    =     $AF80   ; DATA REGISTER A
  DDRA   =     $AF81   ; DATA DIRECTION REG A
  DRB    =     $AF82   ; DATA REGISTER B
  DDRB   =     $AF83   ; DATA DIRECTON REG B
;
;
;
;                  .OR  $7800
;                  .MV  $1000
```

```
;
;         VIDEO  INITIALIZATION  ROUTINE
;
7800  78       VINIT SEI
7801  A97F           LDA  #$7F       ; INIT THE VIDEO 6532
7803  8D81AF         STA  DDRA
7806  A900           LDA  #$00
7808  8D83AF         STA  DDRB
780B  AD82AF         LDA  DRB        ; READ VIDEO DIP SWITCH
780E  2918           AND  #$18
7810  4A             LSR  A
7811  4A             LSR  A
7812  4A             LSR  A
7813  8D80AF         STA  DRA        ; THESE ARE THE COLORS
7816  A9FF           LDA  #$FF       ; INIT VIDEO TO "ON"
7818  8D1BAF         STA  VMODE
781B  A950           LDA  #<VTEST    ; COPY VIDEO LINK ADDRESS
781D  8D06A4         STA  DLINK      ; TO AIM DISPLAY VECTOR
7820  A978           LDA  #>VTEST
7822  8D07A4         STA  DLINK+01
7825  A99F           LDA  #<RDRUB     ; SET UP USER INPUT VECTOR
7827  8D0801         STA  UIN
782A  A97B           LDA  #>RDRUB
782C  8D0901         STA  UIN+1
782F  20DC78         JSR  VCOPY      ; COPY "VPUT" ROUTINE INTO 6532 RAM
7832  20C57A   VINT1 JSR  VCLR       ; CLEAR SCREEN
7835  A9E0           LDA  #$E0       ; INIT CURSOR POSITION
7837  8D04AF         STA  CRPOS
783A  A95F           LDA  #$5F
783C  8D05AF         STA  CRPOS+01
783F  A209           LDX  #9         ; COPY DEFAULT TABS INTO RAM
7841  BDFF7A   TLOOP LDA  DFTAB,X
7844  9D21AF         STA  TABS,X
7847  CA             DEX
7848  10F7           BPL  TLOOP
784A  A2FF           LDX  #$FF       ; RESET STACK POINTER
784C  9A             TXS
784D  4C82E1         JMP  MON        ; GO TO AIM MONITOR
```

```
;
;
;     THE FOLLOWING CODE IS FOR SETTING UP TO GET AROUND
;     THE AIM MONITOR PECULIARITIES. FOR FURTHER EXPLANATION,
;     SEE THE MAIN VIDEO ECHO ROUTINE, "VECHO". PAY PARTICULAR
;     ATTENTION TO THE COMMENTS CONCERNING CURSOR POSITIONING
;     MODES, AND "DEL" KEY PROBLEMS.
;
7850  48       VTEST PHA
;
7851  AD7EA4         LDA  MNJMP+1    ; ARE WE IN BASIC?
7854  C9B0           CMP  #$B0
7856  D00E           BNE  VTSTO      ; IF NOT
7858  AD12A4         LDA  INFLG      ; IF SO, IS THE INPUT FLAG "U"?
785B  C955           CMP  #'U
785D  D048           BNE  VT03       ; IF NOT
785F  A94D           LDA  #'M        ; IF SO, IT PROBABLY SHOULD BE "M"
7861  8D12A4         STA  INFLG
7864  D041           BNE  VT03       ; SKIP THE REST OF THIS NONSENSE
;
7866  AD13A4   VTSTO LDA  OUTFG      ; ARE WE DOING TAPE OUTPUT?
7869  C954           CMP  #'T
786B  D008           BNE  VT000      ; IF NOT
786D  A900           LDA  #0         ; IF SO, SET UP TO GET...
786F  8D7DA4         STA  MNJMP      ; CURSOR POSITION FROM...
7872  8D7EA4         STA  MNJMP+1     ; THE DISPLAY
7875  AD12A4   VT000 LDA  INFLG      ; IS INPUT FROM THE KEYBOARD?
7878  C90D           CMP  #$0D
787A  D007           BNE  VT00       ; IF NOT
787C  A955           LDA  #'U        ; IF SO, FORCE THE USER INPUT HANDLER
787E  8D12A4         STA  INFLG      ; SO DELETES WILL WORK PROPERLY
7881  D019           BNE  VT01
7883  C954     VT00  CMP  #'T        ; ARE WE DOING TAPE INPUT?
7885  D01A           BNE  VT02       ; IF NOT
7887  A900           LDA  #0         ; IF SO, SET UP TO GET CURSOR POSITION
7889  8D7DA4         STA  MNJMP      ; FROM THE DISPLAY
788C  8D7EA4         STA  MNJMP+1
788F  AD1501         LDA  BLK        ; HAVE WE STARTED LOADING TAPE?
7892  F00D           BEQ  VT02       ; IF NOT
7894  68             PLA
7895  48             PHA
7896  C98D           CMP  #$8D       ; IS THIS A DELETE W/ SIGN BIT?
7898  F012           BEQ  VTST1      ; IF SO, IGNORE IT
789A  D005           BNE  VT02
```

```
;
;
789C  A900      VT01  LDA  #0        ; IF NOT DOING TAPE,
789E  8D1501          STA  BLK       ; CLEAR THE BLOCK COUNT
78A1  68        VT02  PLA
78A2  48              PHA
78A3  C916            CMP  #$16      ; CTRL-V? (VIDEO ON/OFF)
78A5  F009            BEQ  VTST2     ; YES
78A7  68        VT03  PLA
78A8  20C678          JSR  VLINK
78AB  60              RTS
78AC  68        VTST1 PLA
78AD  4C05EF          JMP  OUTDS     ; NO VIDEO
;
;
78B0  AD1BAF    VTST2 LDA  VMODE     ; TOGGLE THE VIDEO FLAG
78B3  49FF            EOR  #$FF
78B5  8D1BAF          STA  VMODE
78B8  D006            BNE  VTST3     ; IF NOW ON
78BA  20BB79          JSR  CSOFF     ; IF NOW OFF, TURN OFF CURSOR
78BD  4C4579          JMP  CTRL3
78C0  20C479    VTST3 JSR  CSON      ; TURN ON CURSOR
78C3  4C4579          JMP  CTRL3
;
78C6  48        VLINK PHA
78C7  2C1BAF          BIT  VMODE     ; IS VIDEO ENABLED?
78CA  1003            BPL  VLNK2     ; IF NOT
78CC  20F878          JSR  VECHO     ; IF SO, ECHO TO VIDEO
78CF  2005EF    VLNK2 JSR  OUTDS
78D2  AD04AF          LDA  CRPOS     ; COMPUTE START OF LINE ADDRESS
78D5  29E0            AND  #$E0
78D7  8D16AF          STA  CURLN     ; SAVE IT
78DA  68              PLA
78DB  60              RTS
;
;       COPY MACHINE INSTRUCTIONS FOR
;       THE SYNCHRONIZED VIDEO WRITE
;       INTO THE VIDEO 6532 RAM
;
78DC  48        VCOPY PHA
78DD  A9B8            LDA  #$B8
78DF  8D00AF          STA  VPUT
78E2  A950            LDA  #$50
78E4  8D01AF          STA  VPUT+01
78E7  A9FE            LDA  #$FE
78E9  8D02AF          STA  VPUT+02
78EC  A98D            LDA  #$8D
78EE  8D03AF          STA  VPUT+03
78F1  A960            LDA  #$60
78F3  8D06AF          STA  VPUT+06
78F6  68              PLA
78F7  60              RTS
;
;       VIDEO ECHO ROUTINE
;     (THIS IS THE ENTRY POINT WHERE
;     THE AIM MONITOR PASSES CHARACTERS
;     TO THE VIDEO HANDLER)
;
78F8  48        VECHO PHA
78F9  20DC78          JSR  VCOPY     ; COPY "VPUT" TO RAM
78FC  20BB79          JSR  CSOFF     ; ERASE CURSOR
;
;     SINCE THE AIM AT CERTAIN TIMES ECHOS "SPACE"
;     WHEN "DEL" IS HIT, WE MUST WATCH THE KEYBOARD
;     TO KNOW WHICH IT IS
;
78FF  AD80A4          LDA  DRA2      ; SEE IF "DEL" KEY WAS JUST HIT
7902  C9BF            CMP  #$BF
7904  D00B            BNE  VECHO     ; IF NOT
7906  AD82A4          LDA  DRB2
7909  C9FD            CMP  #$FD
790B  D004            BNE  VECHO     ; IF NOT
790D  68              PLA
790E  A97F            LDA  #$7F      ; IF SO, FORCE DELETE
7910  48              PHA
7911  68        VECHO PLA
7912  297F            AND  #$7F
7914  48              PHA
7915  C90D            CMP  #$0D      ; CARRIAGE RETURN?
7917  D003            BNE  DLTST     ; IF NOT
7919  4C9479          JMP  VCLF1     ; IF SO
791C  C97F      DLTST CMP  #$7F      ; DELETE CHARACTER?
791E  D005            BNE  VCHO1     ; NO
7920  201D7A          JSR  DELET
7923  68              PLA
7924  60              RTS
;
7925  C909      VCHO1 CMP  #$09      ; HORIZONTAL TAB?
7927  D006            BNE  CTRL1     ; IF NOT
7929  20747B          JSR  HTAB      ; IF SO
792C  4C8C79          JMP  VECH2
;
;
792F  C903      CTRL1 CMP  #$03      ; CTRL-C?
7931  D003            BNE  CTRL2     ; NO
7933  4C3278          JMP  VINT1     ; YES - CLEAR SCREEN
7936  C912      CTRL2 CMP  #$12      ; CTRL-R?
7938  D011            BNE  VECO      ; NO
793A  AD80AF          LDA  DRA       ; TOGGLE CONTROL BIT TO REVERSE COLORS
793D  4901            EOR  #$01
793F  8D80AF          STA  DRA
7942  20C479          JSR  CSON      ; TURN CURSOR ON
7945  A2FF      CTRL3 LDX  #$FF      ; RESET STACK POINTER
7947  9A              TXS
7948  4C8AE1          JMP  MON+08    ; BACK TO MONITOR (AFTER PROMPT)
;
;
;     THE FOLLOWING CODE IS TO DETERMINE WHEN WE ARE IN THE
;     AIM MONITOR. WHEN IN THE MONITOR, THE CURSOR POSITION
;     WILL BE COMPUTED FROM THE AIM DISPLAY VECTOR, OTHERWISE
;     ALL OUTPUT IS SEQUENTIAL.
;
794B  AD7DA4    VECO  LDA  MNJMP     ; ARE WE IN THE EDITOR?
794E  C939            CMP  #$39
7950  F032            BEQ  VECH      ; YES
7952  C9CF            CMP  #$CF
7954  F02E            BEQ  VECH      ; YES
7956  AD7EA4          LDA  MNJMP+01  ; ARE WE IN THE AIM ASSEMBLER?
7959  C9D0            CMP  #$D0
795B  F027            BEQ  VECH      ; YES
795D  C9B0            CMP  #$B0      ; ARE WE IN BASIC?
795F  F023            BEQ  VECH
7961  AD80A4          LDA  DRA2      ; WAS LAST KEY HIT AN ESCAPE?
7964  C9FB            CMP  #$FB
7966  D007            BNE  VEC       ; NO
7968  AD82A4          LDA  DRB2
796B  C97F            CMP  #$7F
796D  F015            BEQ  VECH      ; YES
796F  18        VEC   CLC
7970  AD16AF          LDA  CURLN     ; COMPUTE CURSOR POSITION
7973  6D15A4          ADC  CRPO2
7976  8D04AF          STA  CRPOS
7979  AD05AF          LDA  CRPOS+01
797C  6900            ADC  #00
797E  8D05AF          STA  CRPOS+01
7981  20617A          JSR  ENCHK     ; CHECK FOR END OF SCREEN
7984  68        VECH  PLA
7985  48              PHA
7986  2000AF    VECH1 JSR  VPUT
7989  20E579          JSR  NXTPO
798C  68        VECH2 PLA
798D  60              RTS
;
;
;     FROM HERE ON, ALL SUBROUTINES ARE WRITTEN IN STAND-ALONE
;     FASHION TO ALLOW CALLING FROM A USER PROGRAM. ALL
;     REGISTERS ARE ALWAYS PRESERVED. THE VIDEO PROGRAM
;     ITSELF USES MANY OF THESE ROUTINES, WHILE SOME ARE
;     STRICTLY FOR USER CONVENIENCE.
;
;       OUTPUT A SPACE
;
798E  48        VSPAC PHA
798F  A920            LDA  #$20
7991  D0F3            BNE  VECH1
;
;       DO A CARRIAGE-RETURN / LINE-FEED
;
7993  48        VCRLF PHA
7994  20A779    VCLF1 JSR  VCR
7997  4C9B79          JMP  VLF1
;
;       DO A LINE-FEED
;
799A  48        VLF   PHA
799B  20BB79    VLF1  JSR  CSOFF     ; TURN OFF CURSOR
799E  18              CLC
799F  AD04AF          LDA  CRPOS     ; MOVE 32 POS. AHEAD
79A2  6920            ADC  #32
79A4  4CEC79          JMP  NXPO1
;
;       DO A CARRIAGE-RETURN
;
79A7  48        VCR   PHA
79A8  20BB79    VCR1  JSR  CSOFF     ; TURN OFF CURSOR
79AB  AD04AF          LDA  CRPOS     ; MOVE PNTR TO BEG. OF LINE
79AE  29E0            AND  #$E0
79B0  8D04AF          STA  CRPOS
79B3  A900            LDA  #0        ; ZERO THE LINE LENGTH
79B5  8D1FAF          STA  LENTH
79B8  4CFD79          JMP  NXPO2
;
;       TURN OFF CURSOR
;
79BB  48        CSOFF PHA
79BC  AD1CAF          LDA  SAVCR     ;RESTORE SAVED CHARACTER
79BF  2000AF          JSR  VPUT
79C2  68              PLA
79C3  60              RTS
;
;       TURN ON CURSOR
;
79C4  48        CSON  PHA
79C5  20D279          JSR  VFECH     ; SAVE THE CHARACTER
79C8  8D1CAF          STA  SAVCR
79CB  A9AF            LDA  #$AF      ; DISPLAY CURSOR CHARACTER
79CD  2000AF          JSR  VPUT
79D0  68              PLA
79D1  60              RTS
;
;
;       READ CHAR. FROM CURRENT SCREEN POS.
;
79D2  A9AD      VFECH LDA  #$AD      ; CHANGE "VPUT" ROUTINE TO READ
79D4  8D03AF          STA  VPOS
79D7  2000AF          JSR  VPUT      ; READ FROM VIDEO RAM
79DA  48              PHA
79DB  A98D            LDA  #$8D      ; RESTORE "VPUT" ROUTINE
79DD  8D03AF          STA  VPOS
79E0  68              PLA
79E1  60              RTS
;
```

```
;
;        ADVANCE CURSOR TO NEXT POSITION
;
79E2  20BB79   NXPOS  JSR  CSOFF       ; TURN OFF CURSOR AND FALL THRU
;
79E5  48       NXTPO  PHA
79E6  18              CLC              ; INCREMENT 16-BIT POINTER
79E7  AD04AF          LDA  CRPOS
79EA  6901            ADC  #01
79EC  8D04AF   NXPO1  STA  CRPOS
79EF  AD05AF          LDA  CRPOS+01
79F2  6900            ADC  #0
79F4  8D05AF          STA  CRPOS+01
79F7  20617A          JSR  ENCHK       ; TEST FOR END OF SCREEN
79FA  EE1FAF          INC  LENTH
79FD  20C479   NXPO2  JSR  CSON        ; TURN ON CURSOR
7A00  68              PLA
7A01  60              RTS
;
;
;       MOVE CURSOR BACK ONE POSITION
;
7A02  48       LSPOS  PHA
7A03  20BB79   LSPO1  JSR  CSOFF       ; TURN OFF CURSOR
7A06  AD04AF          LDA  CRPOS       ; DECREMENT 16-BIT POINTER
7A09  D003            BNE  LSPO3
7A0B  CE05AF          DEC  CRPOS+1
7A0E  CE04AF   LSPO3  DEC  CRPOS
7A11  AD05AF          LDA  CRPOS+1
7A14  C960            CMP  #$60        ; TOP OF SCREEN?
7A16  903D            BCC  VTOP1       ; YES
7A18  CE1FAF          DEC  LENTH
7A1B  B0E0            BCS  NXPO2


;
;
;      DELETE ROUTINE
;
;      CAN HANDLE TABS IN AIM DISPLAY BUFFER
;
7A1D  48       DELET  PHA
7A1E  B938A4          LDA  DIBUF,Y     ; LOOK AT CHARACTER TO BE DELETED
7A21  C909            CMP  #$09        ; IS IT A TAB?
7A23  D022            BNE  VDEL1       ; IF NOT, DO NORMAL DELETE
7A25  8C1EAF          STY  SAVEY       ; IF SO, SAVE BUFFER POINTER
7A28  AC1FAF          LDY  LENTH       ; GET VIDEO LINE LENGTH
7A2B  88              DEY
7A2C  20027A   DELT1  JSR  LSPOS       ; BACK UP TO START OF TEXT LINE
7A2F  88              DEY
7A30  D0FA            BNE  DELT1
7A32  CC1EAF   DELT2  CPY  SAVEY       ; NOW RE-DISPLAY THE LINE
7A35  F009            BEQ  DELT3       ; FROM THE BEGINNING
7A37  B938A4          LDA  DIBUF,Y
7A3A  20427A          JSR  DELT4
7A3D  C8              INY
7A3E  D0F2            BNE  DELT2
7A40  68       DELT3  PLA
7A41  60              RTS
;
7A42  48       DELT4  PHA              ; SNEAK INTO VECHO
7A43  4C1179          JMP  VECHO       ; WITHOUT LOOKING FOR THE DELETE KEY
;
;                DELETE
;
;      (BACK UP ONE AND ERASE)
;
7A46  48       VDEL   PHA
7A47  20027A   VDEL1  JSR  LSPOS       ; MOVE BACK ONE POSITION
7A4A  A920            LDA  #$20        ; BLANK OUT SAVED CHARACTER
7A4C  8D1CAF          STA  SAVCR
7A4F  68              PLA
7A50  60              RTS
;
;
;      MOVE CURSOR TO HOME
;
7A51  48       VTOP   PHA
7A52  20BB79          JSR  CSOFF       ; TURN OFF CURSOR
7A55  A960     VTOP1  LDA  #$60        ; SET POINTER TO FIRST VIDEO LOCATION
7A57  8D05AF          STA  CRPOS+01
7A5A  A900            LDA  #$00
7A5C  8D04AF          STA  CRPOS
7A5F  F09C            BEQ  NXPO2
;
;      TEST FOR END-OF-SCREEN
;      SCROLL IF NECESSARY
;
7A61  48       ENCHK  PHA
7A62  AD05AF          LDA  CRPOS+01
7A65  C962            CMP  #$62        ; HAVE WE PASSED END OF SCREEN?
7A67  B012            BCS  SCRL1
7A69  68              PLA
7A6A  60              RTS


;
;      COPY A VIDEO LINE TO THE
;      LINE ABOVE
;
;      THIS CODE IS COPIED TO RAM FOR EXECUTION
;
7A6B  A21F     UPL    LDX  #31
7A6D  B8       UPL1   CLV
7A6E  50FE     UPL2   BVC  UPL2
7A70  BD2060          LDA  $6020,X
7A73  9D0060          STA  $6000,X
7A76  CA              DEX
7A77  10F4            BPL  UPL1
7A79  60              RTS
```

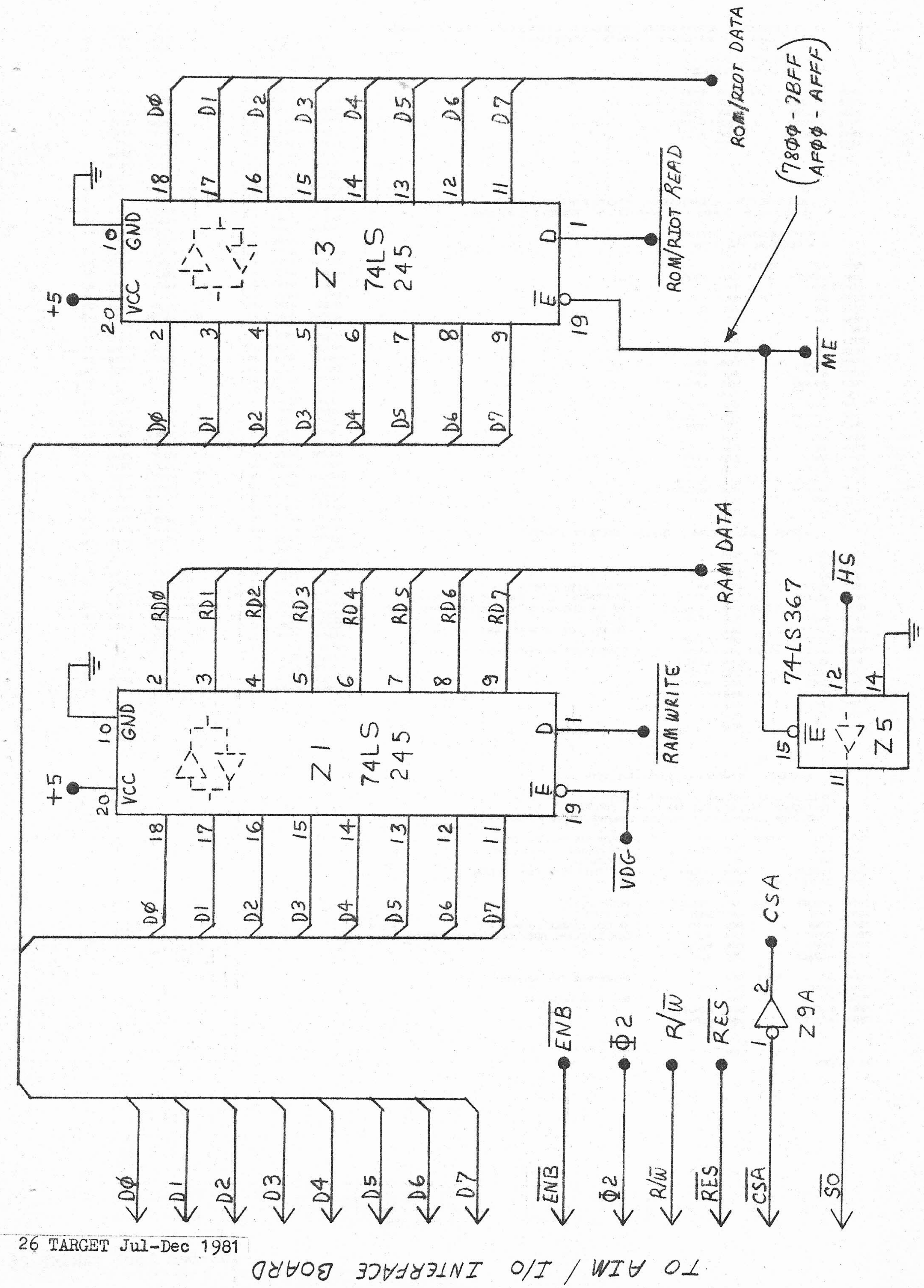
```
;
;
;         SCROLL THE SCREEN
;
7A7A  48       SCROL  PHA
7A7B  8A       SCRL1  TXA
7A7C  48              PHA
7A7D  A20E            LDX  #14         ; COPY "UPL" ROUTINE INTO RAM
7A7F  BD6B7A   SCRL2  LDA  UPL,X
7A82  9D07AF          STA  UPLIN,X
7A85  CA              DEX
7A86  10F7            BPL  SCRL2
7A88  2007AF   SCRL3  JSR  UPLIN       ; COPY A VIDEO LINE TO THE ONE ABOVE IT
7A8B  18              CLC
7A8C  AD10AF          LDA  LINEU
7A8F  6920            ADC  #32         ; SET POINTERS TO NEXT LINE DOWN
7A91  8D10AF          STA  LINEU
7A94  AD11AF          LDA  LINEU+01
7A97  6900            ADC  #00
7A99  8D11AF          STA  LINEU+01
7A9C  18              CLC
7A9D  ADODAF          LDA  LINEL
7AA0  6920            ADC  #32
7AA2  8D0DAF          STA  LINEL
7AA5  AD0EAF          LDA  LINEL+01
7AA8  6900            ADC  #00
7AAA  8D0EAF          STA  LINEL+01
7AAD  C962            CMP  #$62        ; LAST LINE?
7AAF  90D7            BCC  SCRL3       ; IF NOT, CONTINUE
7AB1  A961            LDA  #$61        ; SET POINTER TO BEG. OF LAST LINE
7AB3  8D05AF          STA  CRPOS+01
7AB6  AD04AF          LDA  CRPOS
7AB9  09E0            ORA  #$E0
7ABB  8D04AF          STA  CRPOS
7ABE  20DA7A          JSR  VCLRL       ; CLEAR THE LINE
7AC1  68              PLA
7AC2  AA              TAX
7AC3  68              PLA
7AC4  60              RTS


;
;
;          CLEAR THE SCREEN
;
;     THIS ROUTINE IS DESIGNED FOR SPEED. IT DOES NOT SYNC,
;     AND THEREFORE FLICKERS, BUT IS SCARCELY NOTICEABLE.
;
7AC5  48       VCLR   PHA
7AC6  8A              TXA
7AC7  48              PHA
7AC8  A200            LDX  #00
7ACA  A920            LDA  #$20
7ACC  9D0060   VCLR1  STA  $6000,X     ; WRITE SPACES TO ALL OF SCREEN RAM
7ACF  9D0061          STA  $6100,X
7AD2  E8              INX
7AD3  D0F7            BNE  VCLR1
7AD5  68              PLA
7AD6  AA              TAX
7AD7  4C557A          JMP  VTOP1
;
;       CLEAR THE CURRENT LINE
;
7ADA  48       VCLRL  PHA
7ADB  8A              TXA
7ADC  48              PHA
7ADD  AD04AF          LDA  CRPOS       ; SAVE CURSOR POSITION
7AE0  48              PHA
7AE1  091F            ORA  #$1F        ; START AT END OF LINE
7AE3  8D04AF          STA  CRPOS
7AE6  EE04AF          INC  CRPOS
7AE9  A920            LDA  #$20        ; WRITE SPACES
7AEB  A220            LDX  #32
7AED  CE04AF   VCL1   DEC  CRPOS
7AF0  2000AF          JSR  VPUT
7AF3  CA              DEX
7AF4  D0F7            BNE  VCL1        ; LOOP UNTIL BEGINING OF LINE
7AF6  68              PLA              ; RESTORE CURSOR POSITION
7AF7  8D04AF          STA  CRPOS
7AFA  68              PLA
7AFB  AA              TAX
7AFC  68              PLA
7AFD  60              RTS
;       DEFAULT TAB SETTINGS
;
7AFE  322D28231E
               DFTAB  .BYT 50,45,40,35,30
7B03  1914110B07
                      .BYT 25,20,17,11,7
                      .BT
;
;
;
;       GENERAL PURPOSE UTILITY ROUTINES
;
;
;       STANDARD CRT OUTPUT ROUTINE
;
;
;     (FOR USER CONVENIENCE ONLY)
;
7B08  48       VOUT   PHA
7B09  20DC78          JSR  VCOPY
7B0C  A920            LDA  #$20
7B0E  2000AF          JSR  VPUT
7B11  68              PLA
7B12  48              PHA
7B13  C90D            CMP  #$0D        ; CARRIAGE-RETURN?
7B15  D003            BNE  VOUT1       ; NO
7B17  4CA879          JMP  VCR1        ; YES
7B1A  C90A     VOUT1  CMP  #$0A        ; LINE-FEED?
7B1C  D003            BNE  VOUT2       ; NO
7B1E  4C9B79          JMP  VLF1        ; YES
7B21  C908     VOUT2  CMP  #$08        ; BACK-SPACE?
7B23  D003            BNE  VOUT3       ; NO
7B25  4C037A          JMP  LSPO1       ; YES
7B28  C97F     VOUT3  CMP  #$7F        ; DELETE?
7B2A  D003            BNE  VOUT4       ; NO
7B2C  4C477A          JMP  VDEL1       ; YES
7B2F  C900     VOUT4  CMP  #$00        ; NULL?
7B31  F003            BEQ  VOUT5       ; YES
7B33  4C8679          JMP  VECH1       ; NORMAL CHARACTER
7B36  68       VOUT5  PLA              ; EXIT
7B37  60              RTS
;
```

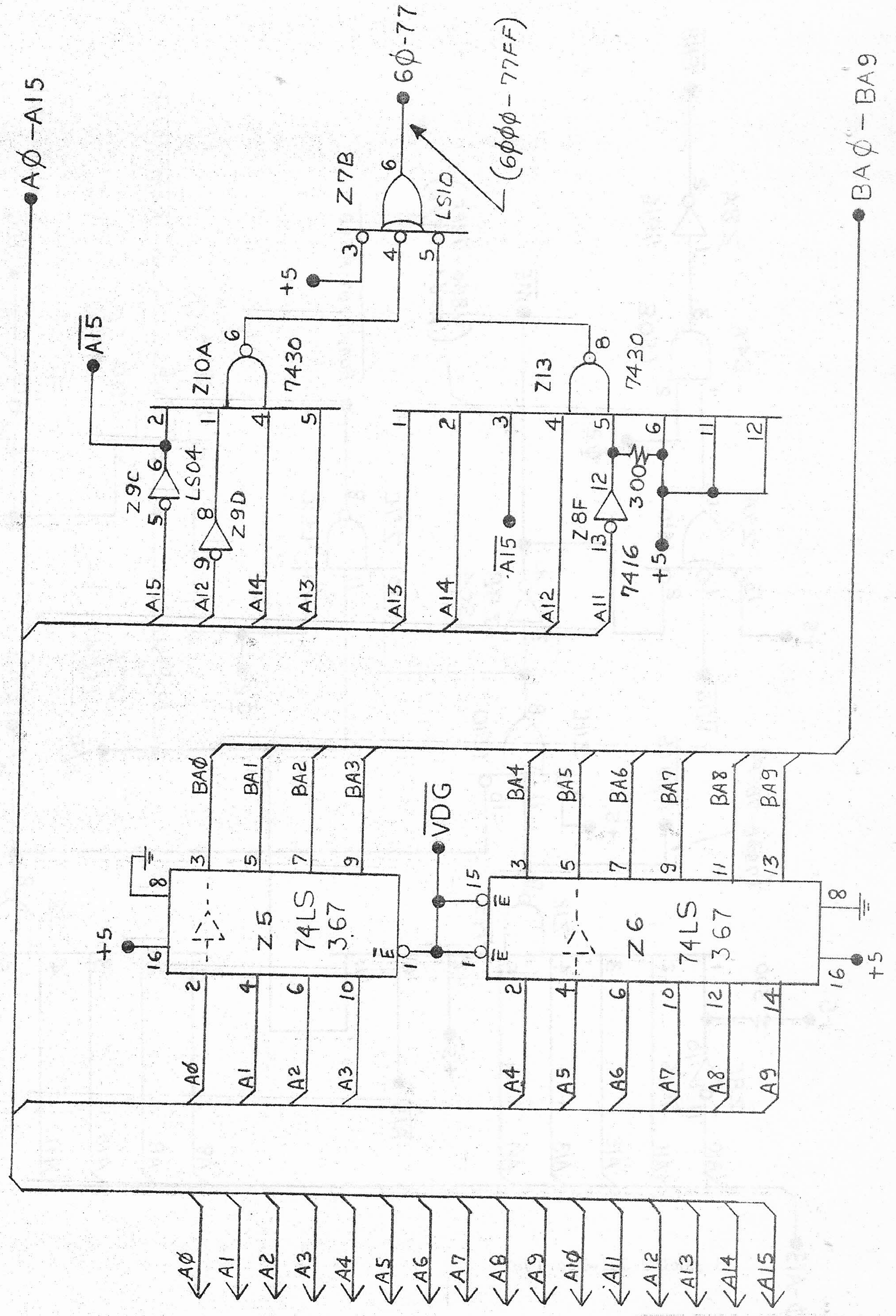
```
;
;          DIRECT CURSOR POSITIONING ROUTINE
;          USING "XPOS" AND "YPOS"
;
;       X = HORIZONTAL POSITION
;       Y = VERTICAL POSITION
;
7B38  209EEB  OUTXY JSR  PHXY
7B3B  AE17AF        LDX  XPOS
7B3E  AC18AF        LDY  YPOS
7B41  20487B        JSR  PUTXY
7B44  20ACEB        JSR  PLXY
7B47  60            RTS
```

```
;
;          DIRECT CURSOR POSITIONING ROUTINE
;          USING THE X AND Y REGISTERS
;
;       X = HORIZONTAL POSITION
;       Y = VERTICAL POSITION
;
7B48  48      PUTXY PHA
7B49  20BB79        JSR  CSOFF
7B4C  98            TYA
7B4D  0A            ASL  A
7B4E  0A            ASL  A
7B4F  0A            ASL  A
7B50  0A            ASL  A
7B51  0A            ASL  A
7B52  8D04AF        STA  VPOS+01
7B55  A960          LDA  #$60
7B57  6900          ADC  #00
7B59  8D05AF        STA  VPOS+02
7B5C  8A            TXA
7B5D  291F          AND  #$1F
7B5F  0D04AF        ORA  VPOS+01
7B62  8D04AF        STA  VPOS+1
7B65  68            PLA
7B66  48            PHA
7B67  F003          BEQ  PTXY1
7B69  2000AF        JSR  VPUT
7B6C  20D279  PTXY1 JSR  VFECH
7B6F  8D1CAF        STA  SAVCR
7B72  68            PLA
7B73  60            RTS
;
;         HORIZONTAL TAB ROUTINE
;
7B74  48      HTAB  PHA
7B75  8A            TXA
7B76  48            PHA
7B77  A209          LDX  #9          ; START WITH LOWEST TAB
7B79  AD1FAF        LDA  LENTH       ; COMPARE IT WITH CURRENT LINE POSITION
7B7C  DD21AF  HTAB1 CMP  TABS,X      ; IS THE CURSOR TO THE LEFT OF THIS TAB?
7B7F  9004          BCC  HTAB2       ; IF SO
7B81  CA            DEX              ; IF NOT, GO TO NEXT TAB
7B82  4C7C7B        JMP  HTAB1
7B85  20E279  HTAB2 JSR  NXPOS       ; SPACE OVER TO THIS TAB
7B88  AD1FAF        LDA  LENTH
7B8B  DD21AF        CMP  TABS,X
7B8E  90F5          BCC  HTAB2
7B90  68            PLA
7B91  AA            TAX
7B92  68            PLA
7B93  60            RTS
```

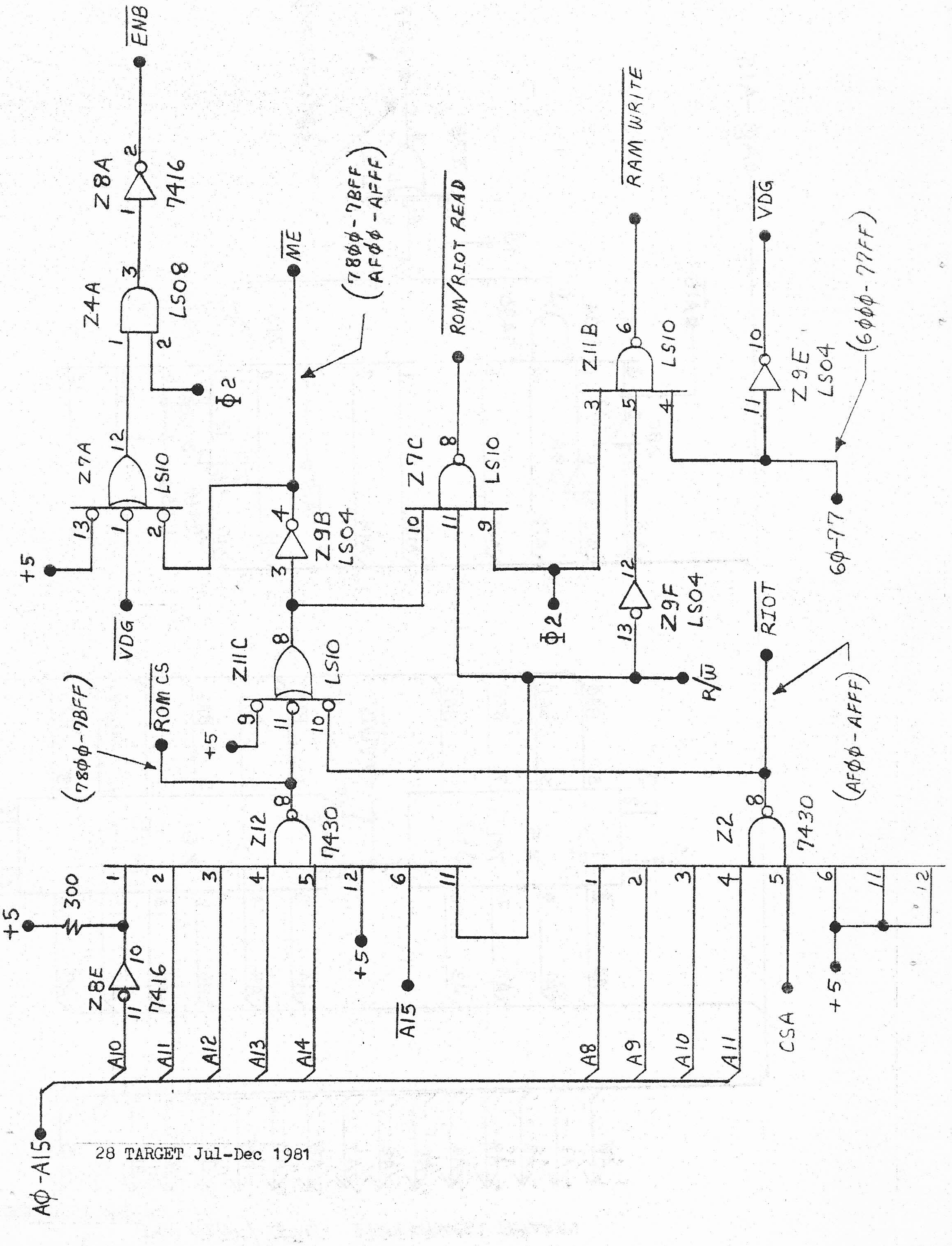
```
;
;         USER INPUT HANDLER
;
;      THIS IS USED FOR ALL KEYBOARD INPUTS
;      TO INSURE THAT DELETES ARE ECHOED PROPERLY
;      (THE USER INPUT FLAG IS FORCED EARLIER IN THE PROGRAM)
;
7B94  C015    RDR1  CPY  #21         ; IF 20 COLUMNS OR LESS
7B96  9015          BCC  RDR2        ; USE NORMAL PROCESSSING
7B98  88            DEY              ; OTHERWISE,
7B99  20F878        JSR  VECHO       ; ADD THIS ECHO FOR DELETES
7B9C  20DCE7  RB2   JSR  PSLS
;
7B9F  2083FE  RDRUB JSR  CREAD       ; READ A KEY
7BA2  C97F          CMP  #$7F        ; DELETE?
7BA4  F0EE          BEQ  RDR1        ; IF SO, HANDLE PROPERLY
7BA6  C03C          CPY  #60         ; DON'T ACCEPT MORE THAN 60 CHARACTERS
7BA8  B0F5          BCS  RDRUB
7BAA  4C76E9        JMP  RED2
;
7BAD  88      RDR2  DEY              ; DECREMENT THE POINTER
7BAE  10EC          BPL  RB2
7BB0  C8            INY              ; DON'T GO NEGATIVE
7BB1  F0EC          BEQ  RDRUB
7BB3  4C70E9        JMP  KEPR
;
;
                    .EN
00 ERRORS
```

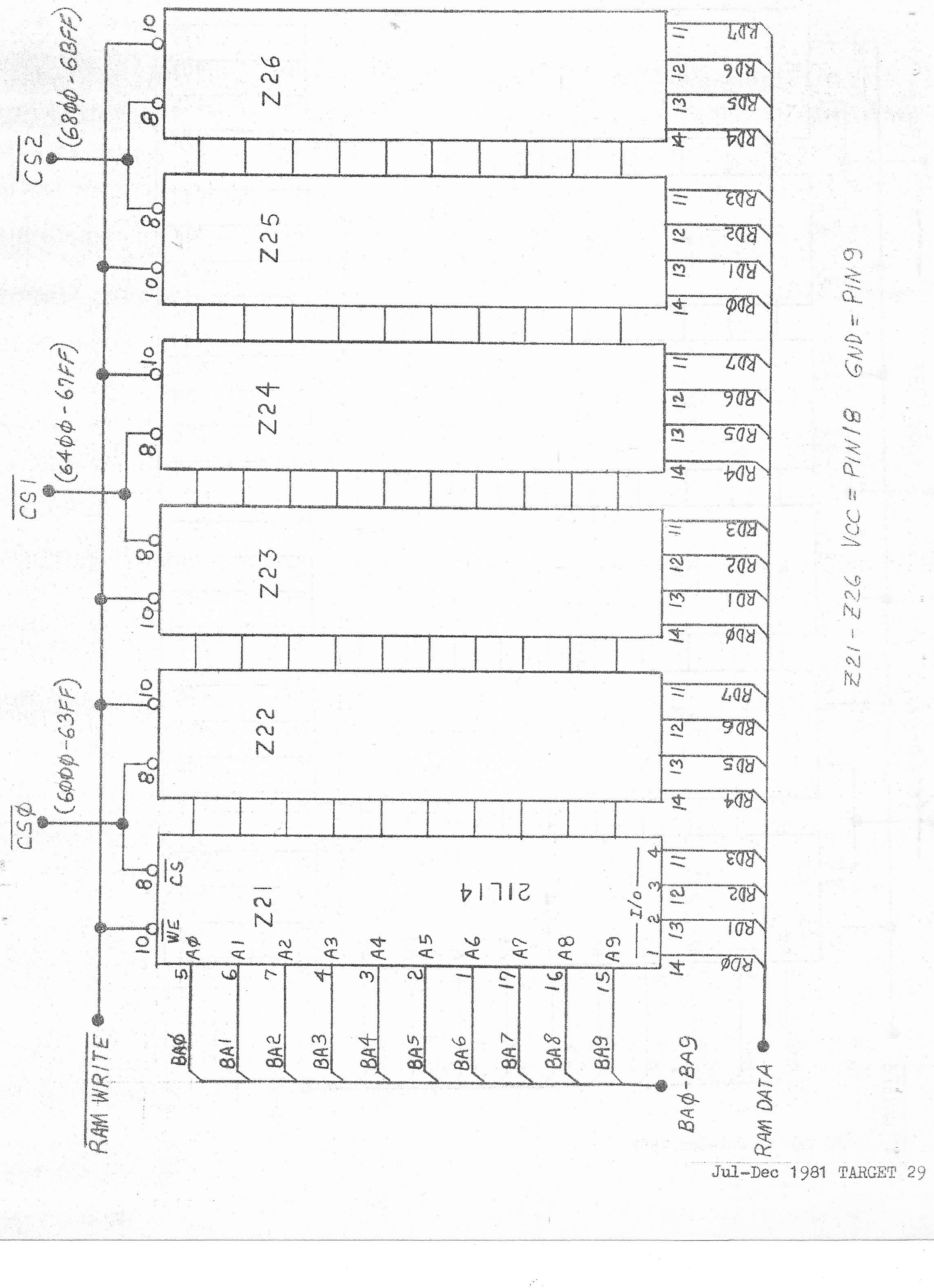
```
SYMBOL TABLE
BLK     0115  02        SCRL3   7A88  01
CREAD   FE83  01        SCROL   7A7A  00
CRPO2   A415  01        TABS    AF21  03
CRPOS   AF04  28        TLOOP   7841  01
CSOFF   79BB  08        UIN     0108  02
CSON    79C4  03        UPL     7A6B  01
CTRL1   792F  01        UPL1    7A6D  01
CTRL2   7936  01        UPL2    7A6E  01
CTRL3   7945  02        UPLIN   AF07  02
CURLN   AF16  02        UPLN1   AF09  00
DDRA    AF81  01        UPLN2   AF0A  00
DDRB    AF83  01        UPLN3   AF0C  00
DELET   7A1D  01        UPLN4   AF0F  00
DELT1   7A2C  01        VCH01   7925  01
DELT2   7A32  01        VCL1    7AED  01
DELT3   7A40  01        VCLF1   7994  01
DELT4   7A42  01        VCLR    7AC5  01
DFTAB   7AFE  01        VCLR1   7ACC  01
DIBUF   A438  02        VCLRL   7ADA  01
DLINK   A406  02        VCOPY   78DC  03
DLTST   791C  01        VCR     79A7  01
DRA     AF80  03        VCR1    79A8  01
DRA2    A480  02        VCRLF   7993  00
DRB     AF82  01        VDEL    7A46  00
DRB2    A482  02        VDEL1   7A47  02
ENCHK   7A61  02        VEC     796F  01
HTAB    7B74  01        VECH    7984  05
HTAB1   7B7C  01        VECHO   7911  03
HTAB2   7B85  02        VECH1   7986  02
HWAIT   AF01  00        VECH2   798C  01
INFLG   A412  04        VECHO   78F8  02
KEPR    E970  01        VECO    794B  01
LENTH   AF1F  06        VFECH   79D2  02
LINEL   AF0D  04        VINIT   7800  00
LINEU   AF10  04        VINT1   7832  01
LSPO1   7A03  01        VLF     799A  00
LSPO3   7A0E  01        VLF1    799B  02
LSPOS   7A02  02        VLINK   78C6  01
MNJMP   A47D  07        VLNK2   78CF  01
MON     E182  02        VMODE   AF1B  04
NXPO1   79EC  01        VOUT    7B08  00
NXPO2   79FD  03        VOUT1   7B1A  01
NXPOS   79E2  01        VOUT2   7B21  01
NXTPO   79E5  01        VOUT3   7B28  01
OUTDS   EF05  02        VOUT4   7B2F  01
OUTFG   A413  01        VOUT5   7B36  01
OUTXY   7B38  00        VPOS    AF03  06
PHXY    EB9E  01        VPUT    AF00  12
PLXY    EBAC  01        VSPAC   798E  00
PSLS    E7DC  01        VT00    7883  01
PTXY1   7B6C  01        VT000   7875  01
PUTXY   7B48  01        VT01    789C  01
RB2     7B9C  01        VT02    78A1  03
RDR1    7B94  01        VT03    78A7  02
RDR2    7BAD  01        VTEST   7850  02
RDRUB   7B9F  04        VTOP    7A51  00
RED2    E976  01        VTOP1   7A55  02
SAVCR   AF1C  04        VTST0   7866  01
SAVEY   AF1E  02        VTST1   78AC  01
SCRL1   7A7B  01        VTST2   78B0  01
SCRL2   7A7F  01        VTST3   78C0  01
                        XPOS    AF17  01
                        YPOS    AF18  01
```


ROM/RIOT DATA
(7800 - 7BFF
AF00 - AFFF)
ROM/RIOT READ
Z3
74LS
245
VCC
GND
ME
D0 D1 D2 D3 D4 D5 D6 D7
RAM DATA
RD0 RD1 RD2 RD3 RD4 RD5 RD6 RD7
RAM WRITE
Z1
74LS
245
VDG
74LS367
HS
Z5
+5
CSA
Z9A
ENB
Φ2
R/W
RES
CSA
SO
TO AIM / I/O INTERFACE BOARD

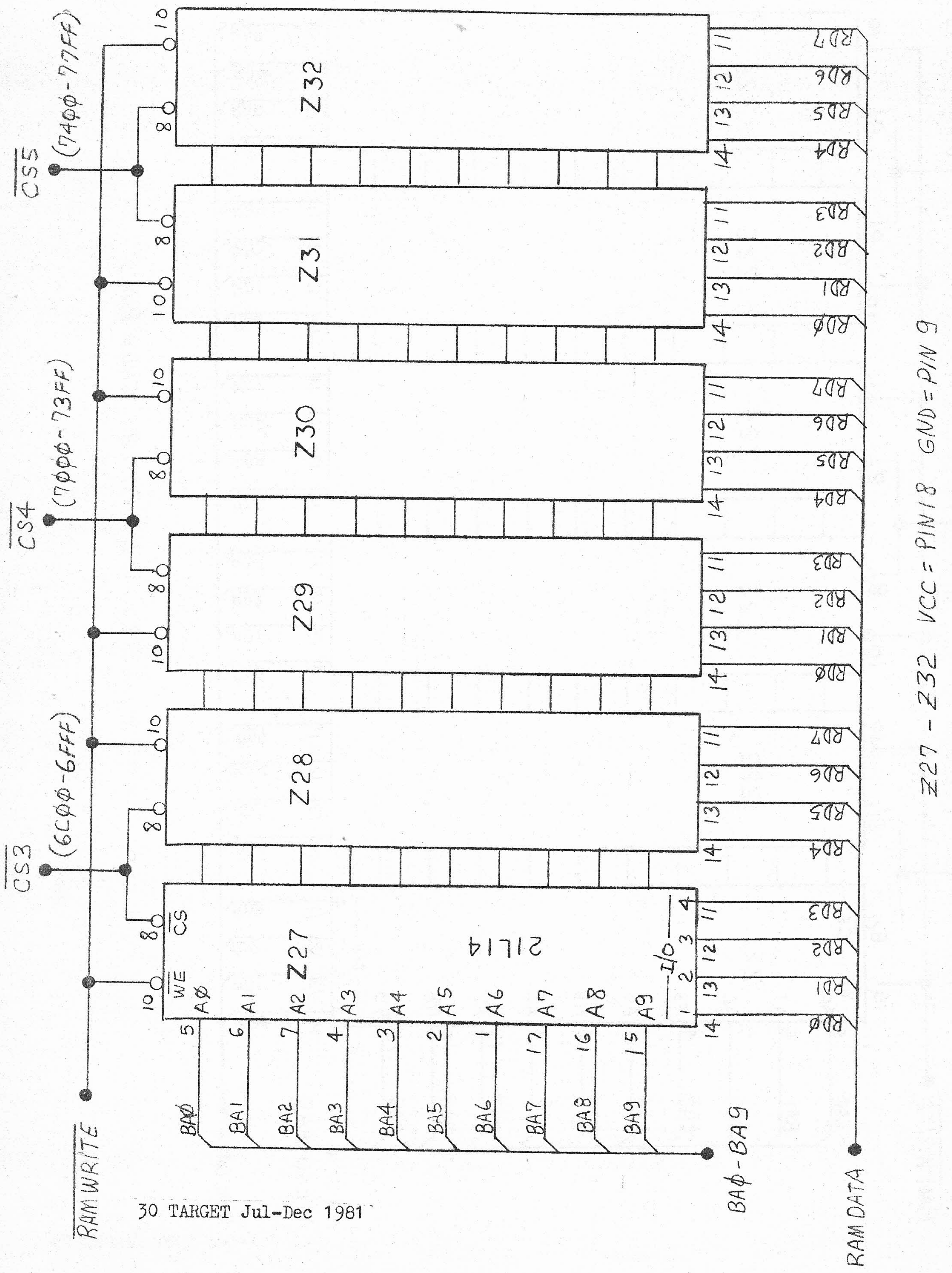


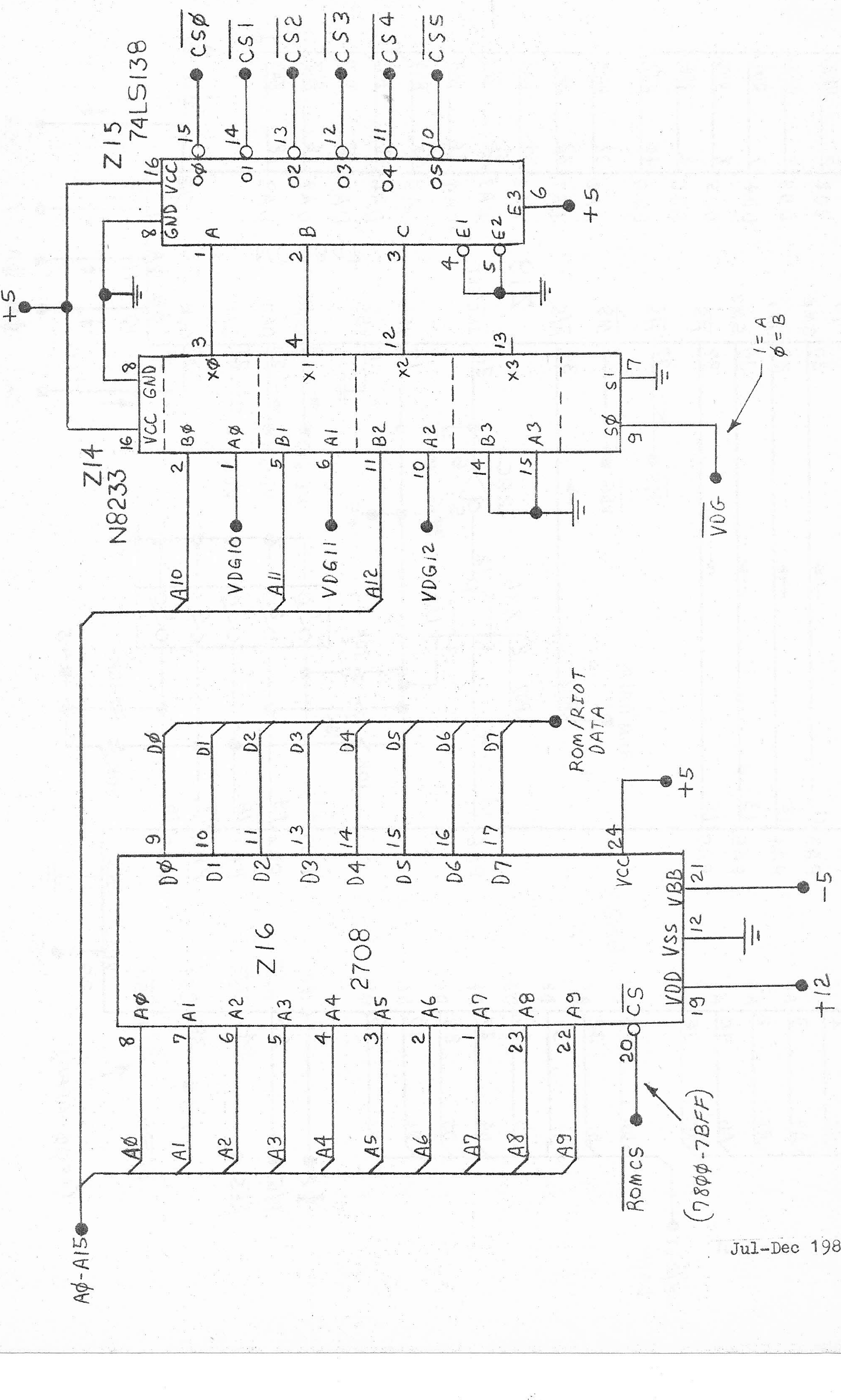
TO AIM / I/O INTERFACE BOARD

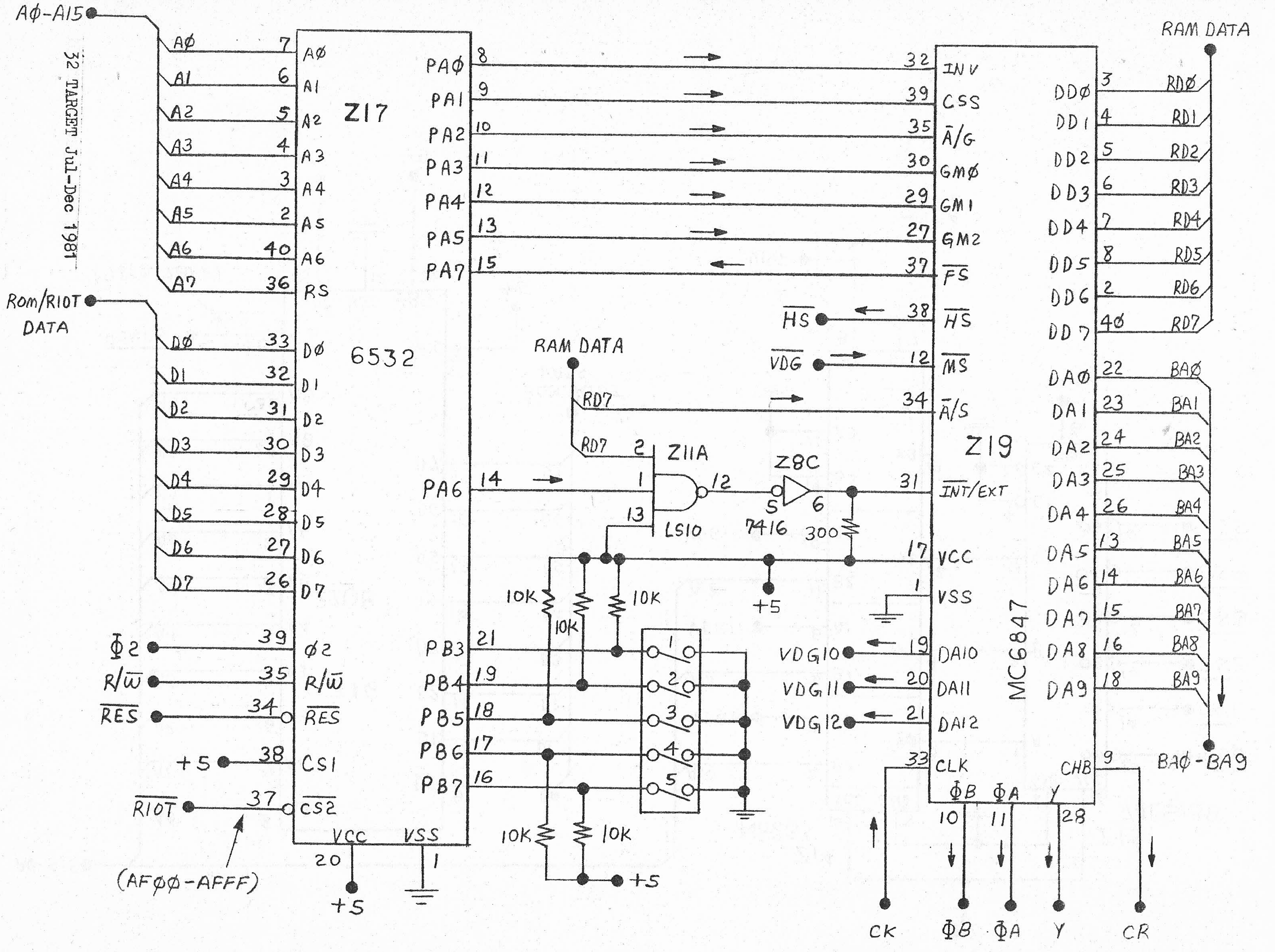

ENB
Z8A
7416
Z4A
LS08
Φ2
ME
(78ΦΦ-7BFF
AFΦΦ-AFFF)
ROM/RIOT READ
RAM WRITE
VDG
(6ΦΦΦ-77FF)
Z7A
LS10
+5
VDG
Z9B
LS04
Z7C
LS10
Z11B
LS10
Z9E
LS04
6Φ-77
Z9F
LS04
R/W
RIOT
(AFΦΦ-AFFF)
Z11C
LS10
ROM CS
(78ΦΦ-7BFF)
Z12
7430
Z2
7430
300
Z8E
7416
A15
CSA
AΦ-A15
A10
A11
A12
A13
A14
A8
A9
A10
A11



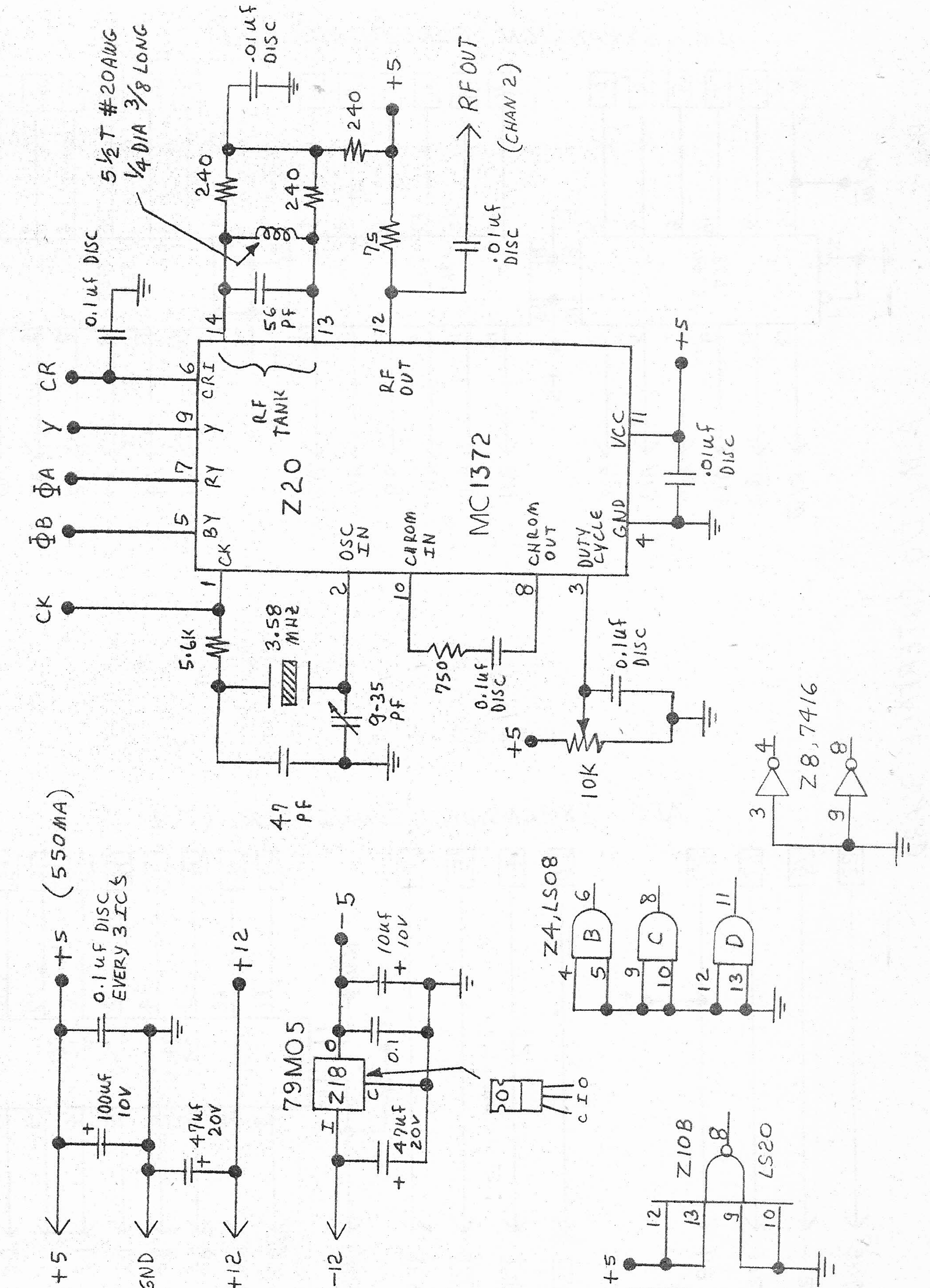
RAM WRITE
CSØ (6ØØØ-63FF)
CS1 (64ØØ-67FF)
CS2 (68ØØ-6BFF)
Z21
2114
WE
CS
I/O
Z22
Z23
Z24
Z25
Z26
BAØ
BA1
BA2
BA3
BA4
BA5
BA6
BA7
BA8
BA9
BAØ-BA9
RDØ
RD1
RD2
RD3
RD4
RD5
RD6
RD7
RAM DATA
Z21 - Z26 VCC = PIN 18 GND = PIN 9


RAM WRITE
CS3 (6C00-6FFF)
CS4 (7000-73FF)
CS5 (7400-77FF)
Z27
2114
Z28
Z29
Z30
Z31
Z32
WE
CS
I/O
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9
BA0 BA1 BA2 BA3 BA4 BA5 BA6 BA7 BA8 BA9
RD0 RD1 RD2 RD3 RD4 RD5 RD6 RD7
BA0-BA9
RAM DATA
Z27 - Z32 VCC = PIN 18 GND = PIN 9


Z15
74LS138
Z14
N8233
Z16
2708
CS0 CS1 CS2 CS3 CS4 CS5
VDG10 VDG11 VDG12
VDG
1 = A
0 = B
ROM/RIOT DATA
ROMCS
(7800-7BFF)
A0-A15
+5
-5
+12

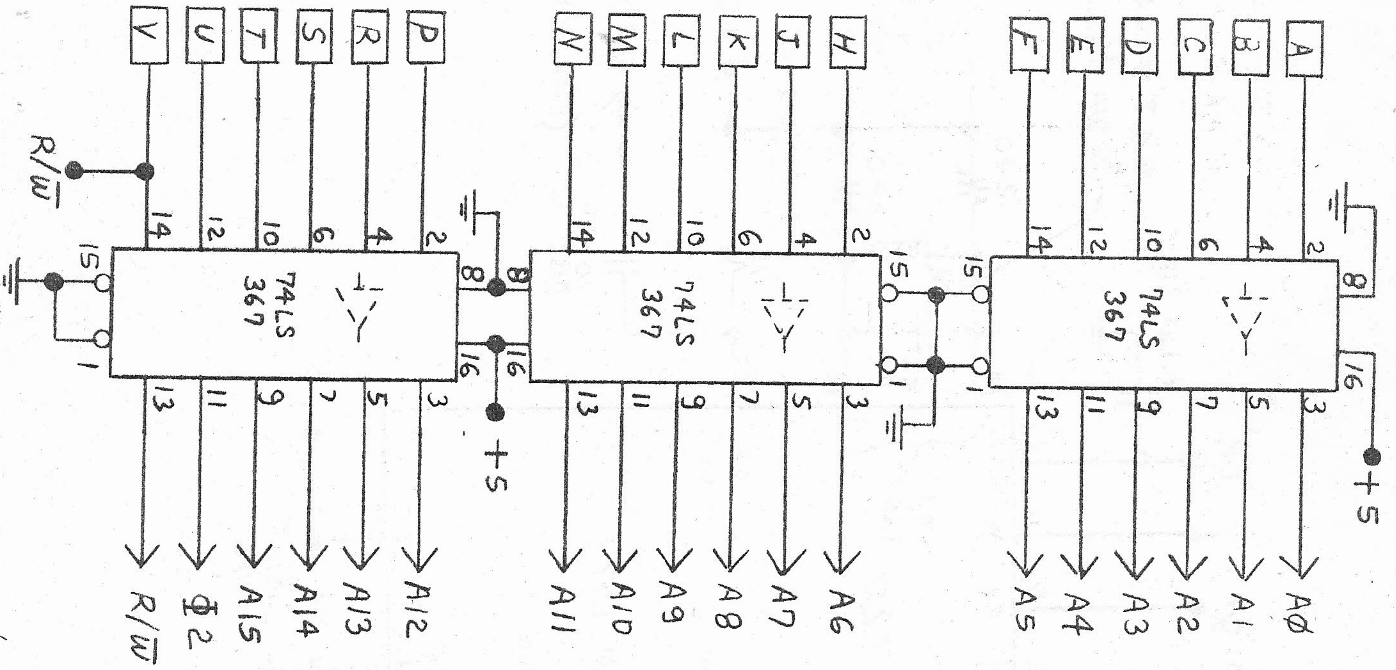


Z17
6532
Z19
MC6847
Z11A
LS10
Z8C
7416
RAM DATA
ROM/RIOT DATA
AØ-A15
BAØ-BA9
INV
CSS
FS
HS
MS
INT/EXT
VCC
VSS
CLK
CHB
ΦB
ΦA
Y
CK
CR
VDG
VDG10
VDG11
VDG12
Φ2
R/W
RES
RIOT
CS1
CS2
(AFØØ-AFFF)
+5
10K
300

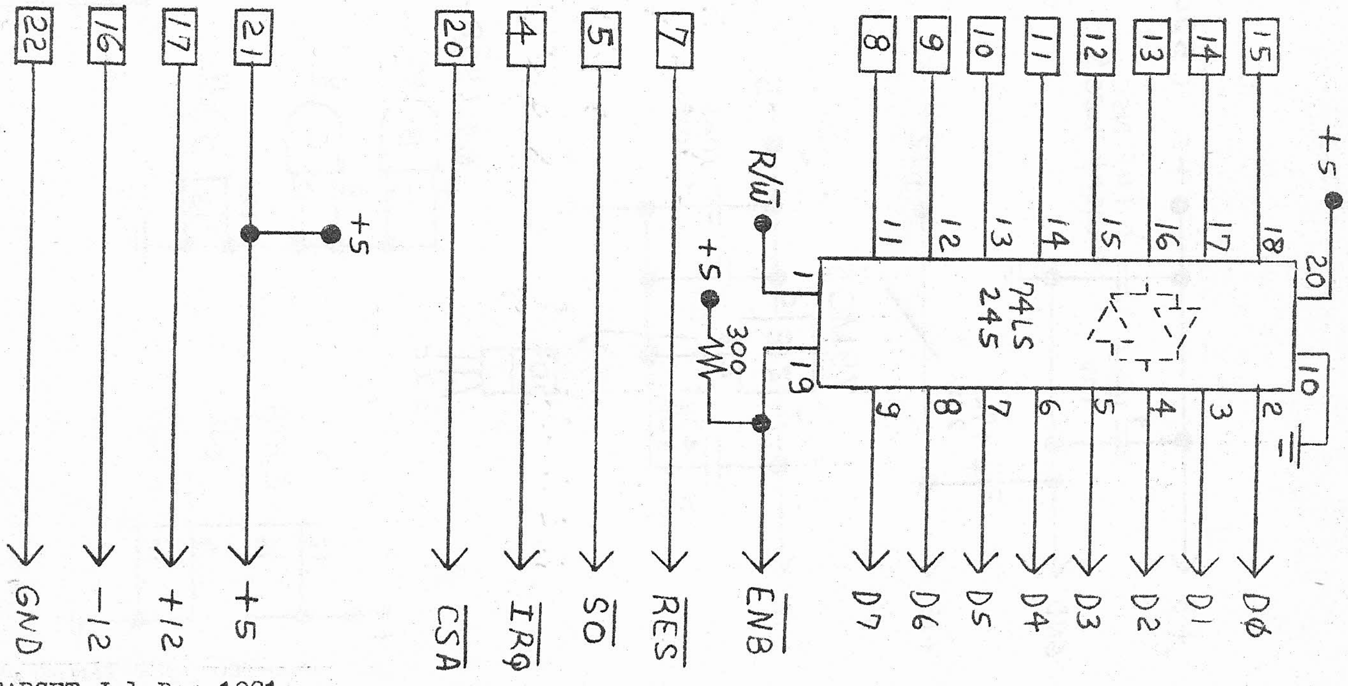

MC1372
Z20
RF TANK
RF OUT
(CHAN 2)
5½ T #20AWG
¼ DIA 3/8 LONG
3.58 MHZ
9-35 PF
47 PF
56 PF
10K
750
5.6K
240
75
.01uf DISC
0.1uf DISC
CR
Y
ΦA
ΦB
CK
CRI
RY
BY
OSC IN
CHROM IN
CHROM OUT
DUTY CYCLE
GND
VCC
Z8, 7416
Z4, LS08
Z10B
LS20
79MO5
Z18
(550MA)
0.1uf DISC EVERY 3 ICS
100uf 10V
47uf 20V
10uf 10V
+5
GND
+12
-12
-5
TO AIM/I/O INTERFACE BOARD

AIM/I/O INTERFACE BOARD

AIM EXPANSION CONNECTOR (J3)



AIM EXPANSION CONNECTOR (J3)



If you desire to use products from different suppliers you must implement your own bus. I am partial to the idea of using 22/44 connectors with wire wrap pins. You then bus the processor signals to the appropriate edge connector pins to meet the requirements of the boards to be added. Also I would suggest adding a buffer board between the AIM 65 and this expansion bus. Something like Dick Buchen's I/O Interface board in this issue will do the trick. This approach will cover most situations. If an odd connector is needed then just wire it in. What this arrangement lacks is neatness or compactness. Also not very easy to put together. To avoid this hassle choose a particular product line and stick with it.

Which product line should be used? Most offer basic products such as RAM. So if this is your basic need then choose any product line. Instrumental in choosing a product line is in the products offered and how they meet your present as well as future needs. Plan ahead. One special consideration is the addition of a high-speed storage device. Even if you have no plans for this device give it special thought as you may change your mind in the future.

What should I add to my AIM 65? Add memory! For assembly language have 12K on hand and for BASIC 20K. The more the better.

This is all for this installment. If necessary we will continue at a later time.

THE TARGET
c/o DONALD CLEM
R.R. #2, CONANT RD.
SPENCERVILLE, OHIO 45887

EXCERT, INC
P O BOX 8600
WHITE BEAR LAKE, MN 55110