

PERCOM

the
**ELECTRIC
WINDOW™**
USERS MANUAL

©1979

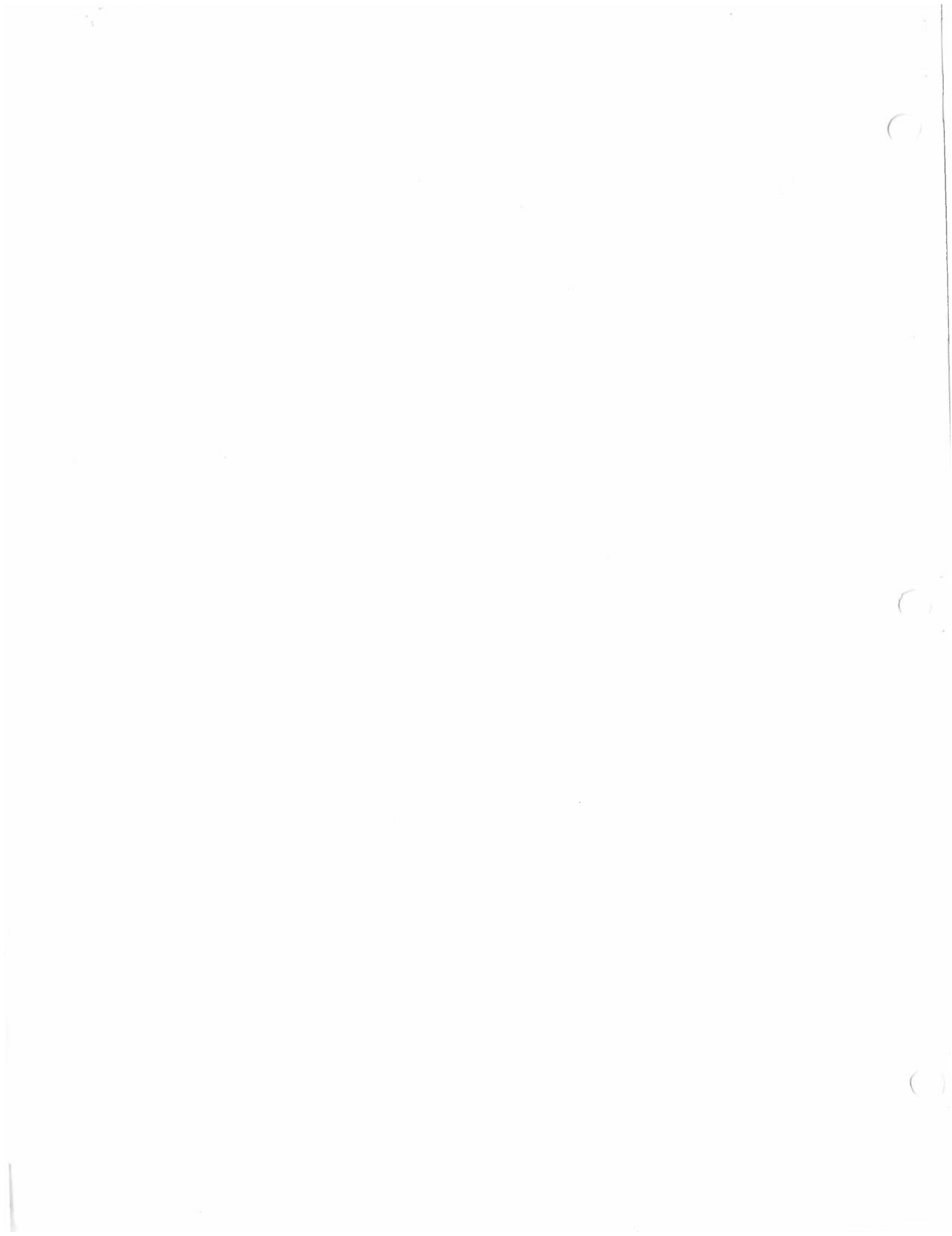
**PERCOM DATA COMPANY
211 N. KIRBY
GARLAND, TEXAS 75042**

Percom Data Co. Inc.
211 N. Kirby
Garland, Tx 75042
(214) 272-3421

The ELECTRIC WINDOW
A Memory Resident Video Controller
for SS-50 Bus Computers

Systems Information
and Programming Guide

Copyright (C) 1979
Percom Data Co.
All Rights Reserved

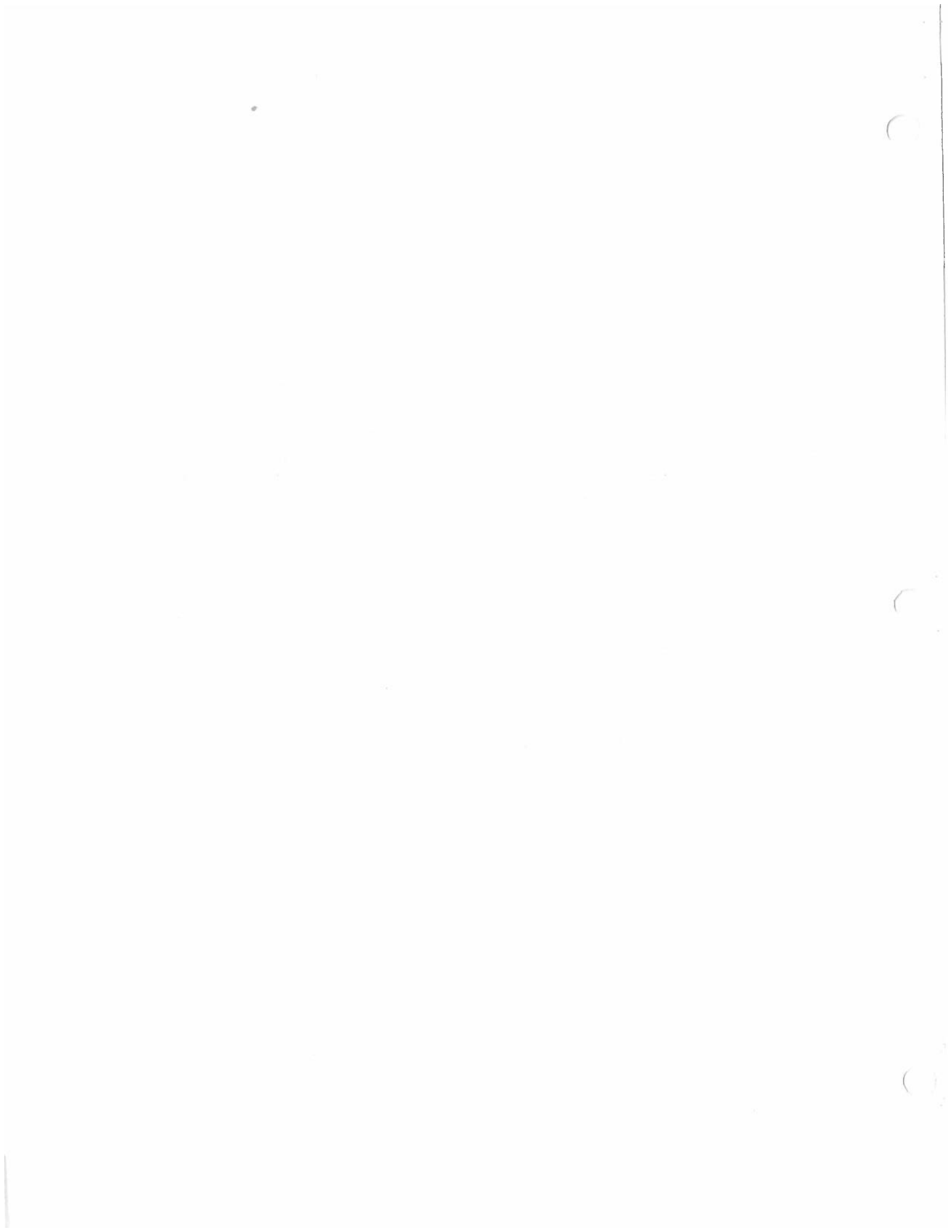


IMPORTANT NOTICE

All material in is manual is copyrighted by PERCOM DATA CO. INC. No portion of it may be copied or reproduced in any manner without the written permission of PERCOM DATA CO. INC.

Although the information contained in this publication has been thoroughly checked for accuracy and reliability, PERCOM DATA CO. INC. shall have no liability or responsibility to customer or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by products or programs sold by PERCOM DATA*CO. INC., including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such products or programs. Furthermore, PERCOM DATA CO. INC. does not represent the described equipment or programs as suitable for any purpose and does not assume any liability arising out of the application or use of any product, circuit or program described herein.

PERCOM DATA CO. INC. reserves the right to make changes to any products or specifications described herein without notice.



PERCOM DATA CO. INC.
211 N. Kirby
Garland, Tx 75042
(214) 272-3421

STATEMENT OF LIMITED WARRANTY

For a period of 90 days from the date of delivery, Percom Data Co. Inc. warrants to the original purchaser that the computing equipment described herein shall be free from defects in material and workmanship under normal use and service. During this period, if a defect should occur, the equipment must be returned to the Percom Data Co. Service Facility at the above address for repair. The purchaser must prepay all shipping and insurance charges and must supply proof of purchase from Percom Data Co. or an authorized Percom dealer or distributor. Purchaser's sole and exclusive remedy in the event of defect is expressly limited to the correction of the defect by adjustment, repair or replacement at Percom's election and sole expense, except there shall be no obligation to replace or repair items which by their nature are expendable. No representation or other affirmation of fact, including, but not limited to, statements regarding capacity, suitability for use, or performance of the equipment, shall be or be deemed to be a warranty or representation by Percom Data Co. Inc., for any purpose, nor give rise to any liability or obligation of Percom Data Co. Inc. whatsoever.

EXCEPT AS SPECIFICALLY PROVIDED IN THIS AGREEMENT, THERE ARE NO OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND IN NO EVENT SHALL PERCOM DATA CO. INC. BE LIABLE FOR LOSS OF PROFITS OR BENEFITS, INDIRECT, SPECIAL, CONSEQUENTIAL OR OTHER SIMILAR DAMAGES ARISING OUT OF ANY BREACH OF THIS WARRANTY OR OTHERWISE.

PERCOM DATA CO. INC.
211 N. Kirby
Garland, TX 75042
(214) 272-3421

NOTICE

All COMPUTER PROGRAMS sold or distributed by PERCOM DATA CO. INC. are sold or distributed on an AS-IS basis WITHOUT WARRANTY.

PERCOM DATA CO. INC. shall have no LIABILITY or responsibility to customers, or any other person or entity with respect to any LIABILITY, LOSS, OR DAMAGE caused or alleged to be caused directly or indirectly by equipment or computer programs sold by PERCOM DATA CO. INC. including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such equipment or computer programs.

Good data processing procedure dictates that the user test the program, run and test sample sets of data, and run the system in parallel with the system previously in use for a period of time adequate to insure that results of operation of the computer or program are satisfactory.

This program is the sole property of the author or PERCOM DATA CO. INC. and has been registered with the United States Copyright Office. Lawful users of this program may use the program themselves, but may not make copies or translations of the program in any form other than as necessary to use the program. It is a violation of the Federal Copyright Laws, punishable by fines and/or imprisonment, for anyone to Copy or Translate this program for any other purpose, including for purposes of resale, license or lease to others.



1.0 Introduction:

The ELECTRIC WINDOW is a memory resident video display controller. It is designed for SS-50 bus computers and standard video (CRT) Monitors or appropriately modified TV sets. It is NOT designed for RF coupling to standard television receivers.

The ELECTRIC WINDOW can replace or supplement standard Data Terminals in most applications. In applications such as WORD PROCESSING, the instantaneous REAL TIME display updating capability of the ELECTRIC WINDOW permits a tremendous speed and efficiency improvement over corresponding systems using a standard data terminal.

Since the ELECTRIC WINDOW utilizes a programmable CRT controller (Standard Micro Systems 5027) it can be programmed for different display configurations.

1.1 Character Generators:

Although the ELECTRIC WINDOW has provision for two character generator ROMs, only the normal character set ROM is supplied. Refer to Appendix D for the standard character set. The second ROM is optional and may be provided by the user. The socket will accept a 2708 EPROM or equivalent, however the circuit card is easily modified to accept a 2716. Refer to Appendix F. The optional ROM may be programmed for special symbols or characters as required. Refer to Appendix E for more specific optional ROM programming information. Character generator selection is software controlled.

1.2 Display Highlighting:

A programmable dual intensity display feature is an alternative to the optional character generator. Under program control individual characters may be either full or half intensity.

1.3 Memory Map:

The ELECTRIC WINDOW contains its own video refresh memory which is also accessible by the host processor. The ELECTRIC WINDOW memory map is as follows:

Address ----- (Hexadecimal)	Function -----
D800-DFBF	Display Refresh Memory
DFC0	Horizontal Line Count
DFC1	Horizontal Sync Width/Delay
DFC2	Scans-Characters/Data Row
DFC3	Skew-Data Rows/Frame
DFC4	Scans/Frame
DFC5	Vertical Data Start
DFC6	Last Displayed Row
DFC7	--
DFC8	--
DFC9	--
DFCA	Reset controller
DFCB	Up scroll
DFCC	Cursor Character Address
DFCD	Cursor Line Address
DFCE	Start controller

7BF BYTES

X
Y



The contents of the Display Refresh Memory determines what will be displayed on the CRT. For example, if the processor stores a \$41 in Location \$D800, an upper case 'A' will appear on the CRT in the position corresponding to Location \$D800. The exact physical position of the character on the CRT is also controlled by the LAST DISPLAYED ROW (\$DFC6) and UP SCROLL (\$DFCB) registers.

The address locations from \$DFC0 thru \$DFCE program the CRT controller. The CRT controller must be initialized to determine the proper display format before the ELECTRIC WINDOW can be used. For detailed controller Initialization information, refer to Appendix A. The paragraph on controller Initialization contains the Initialization parameters for 80x24 Interlace scan or 80x16 Non-Interlace scan.

1.4 Choosing a CRT Monitor:

The ELECTRIC WINDOW is configured to drive a 12 mHz bandwidth or better CRT Display with a standard composite Video (Sync combined with Video) input. The ELECTRIC WINDOW may also be configured for CRT monitors with separate Sync and video inputs.

The ELECTRIC WINDOW is capable of a 24 line, 80 character/line display in the interlace scan mode. However, to avoid the annoying flicker component characteristic of interlace scanning, it is necessary to select a CRT monitor with a long persistence phosphor (P31 or equivalent).

If you are using a CRT monitor with the more common short persistence P4 phosphor you should use a non-interlaced scan mode such as 80 characters by 16 lines. Initialization procedures for both interlace and non-interlace scan mode are described later.

You may convert or use a converted TV set. Refer to the TV TYPWRITER COOKBOOK by Don Lancaster for additional information. 80 character/line display requires a wider video bandwidth than is normally available in the standard TV set, consequently, the display will not be as crisp as on a CRT monitor designed for the purpose.

We have used the low cost Leedex Video 100 monitor with excellent results. It is about the same price as a small black and white TV but provides considerably superior result. We found it necessary to adjust a trim pot inside the Leedex cabinet to reduce the size of the display so that all 80 characters/line were visible. The only criticism we have of the Leedex monitor is the lack of sharp focus in the corners of the display.

2.0 Installation:

2.1 Configuring the ELECTRIC WINDOW:

The ELECTRIC WINDOW circuit card has several strapping options.

2.1.1 Optional Symbols ROM

To enable the optional symbols ROM:

strap point U to point T (default)

strap point P to point R (default)

To disable the optional symbols ROM:

remove strap from U to T

strap point U to point S

break circuit trace from P to R

strap point P to point Q

2.1.2 Dual Intensity Display

To enable the Dual Intensity feature:

Install a IN914 diode in CR3 (default)

To disable the Dual Intensity feature:

Remove diode CR3

The Intensity differential level is controlled by resistors R5 and R6. Change the value of R6 to suit your taste.

2.1.3 Separate Sync:

The Electric Window is configured for composite Sync and Video. To separate the Sync pulses from the video:

Break the circuit trace between points X and Z

Jumper point X to Y

2.1.4 Cable Termination:

The ELECTRIC WINDOW is designed to drive a 75 ohm terminated video cable. If your CRT monitor does not have 75 ohm termination (it should for best performance) you must install a 75 ohm resistor in R1. Refer to FIGURE 1.

2.1.5 Address decoding:

All of the driver software and initialization routines described in this manual assumes the ELECTRIC WINDOW is addressed at \$D800. The address decoding circuit may be modified to permit the ELECTRIC WINDOW to be located at another address. However, we suggest you study the schematic if you wish to make such changes. Points A, B, C, D are provided for address modification.

2.2 Video Connection:

A length of coaxial cable should be used to connect the ELECTRIC WINDOW to a CRT display. Refer to FIGURE 1. Connect the center conductor of the coax cable to the VIDEO OUTPUT connection, connect the coax shield to one of the GROUND connections. The SYNC OUTPUT is not used unless your CRT monitor requires a SYNC signal separate from the VIDEO. Coaxial cable and various coaxial cable fittings are available from most RADIO SHACK or electronic supply stores.

To relieve the stress on the connections, secure the cable to the PCB with a CABLE TIE at the point indicated.

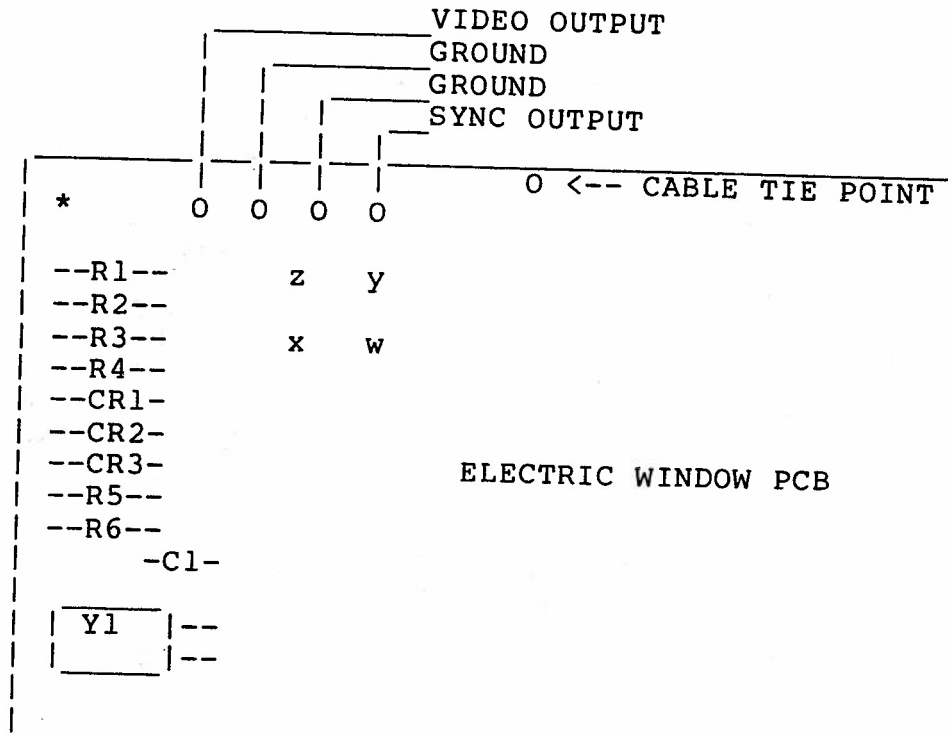


FIGURE 1.

3.0 Software Drivers:

Appendix G contains software drivers for the 6800 MPU. These listings may be used as models for your own software development.

3.1 Initialization:

Since the ELECTRIC WINDOW uses a programmable CRT controller it must be initialized before use. The ELECTRIC WINDOW may be initialized to many different formats; however, the following "Down-Load" sequences will permit you to get started with either a 16 line or 24 line format. Simply store the indicated values in the appropriate addresses in the order shown.

Address	24-line Interlaced	16-line Non-Interlaced	Remarks
DFCA	00	00	Reset
DFC0	65	65	
DFC1	E7	64	Horizontal centering
DFC2	7D	6D	
DFC3	97	8F	
DFC4	06	03	
DFC5	20	20	Vertical centering
DFC6	17	00	Bottom line
DFCA	00	00	Reset
DFCE	00	00	Start

APPENDIX A
 CRT CONTROL REGISTER
 PROGRAMMING CHART

Nine 8-bit registers fully program the CRT Controller. Bit assignments for these registers are shown in Appendix B.

HORIZONTAL FORMATTING:

Characters/Data Row
 (\$DFC2, Bits 0-2)

A 3-bit code providing 8 line lengths of 20, 32, 40, 64, 72, 80, 96, or 132 characters.

DB2	DB1	DB0	CHAR/DATA ROW
0	0	0	20
0	0	1	32
0	1	0	40
0	1	1	64
1	0	0	72
1	0	1	80
1	1	0	96
1	1	1	132

Horizontal Sync Delay
 (\$DFC1, Bits 0-2)

A 3-bit code providing up to 8 character times for "front-porch" generation.

N = 1 to 7 character times
 N <> 0, DB0 = LSB

Horizontal Sync Width
 (\$DFC1, Bits 3-6)

A 4-bit code providing up to 16 character times for Sync pulse generation.

N = 1 to 15 character times
 N <> 0, DB3 = LSB

Horizontal Line Count
 (\$DFC0, Bits 0-7)

An 8-bit code providing up to 256 character times for TOTAL Horizontal formatting (including Sync time).

Total character times/line = N+1
 N = 0 to 255, DB0 = LSB

Skew Bits
 (\$DFC3, Bits 6-7)

A 2-bit code providing a 0 to 2 character skew between the horizontal address counter and the Horizontal Blanking and Sync signals to permit retiming of the video data prior to generation of the composite video signal. The cursor is also skewed as a function of this code.

DB7	DB6	Sync/Blank (Skew {char times})	Cursor
0	0	0	0
1	0	1	0
0	1	2	1
1	1	2	2

VERTICAL FORMATTING:

Interlaced/Non-Interlaced
(\$DFC1, Bit 7)

This bit provides odd/even field formatting for interlaced display systems.

NOTICE: Interlaced display may cause annoying flicker on short persistence CRT monitors.

Bit 7 = 0 Non-Interlaced

Bit 7 = 1 Interlaced

Scans/Frame
(\$DFC4, Bits 0-7)

An 8-bit code defined as follows:

Let X = the value of the 8-bit code
1) Interlace mode: Scans/frame = $2X + 513$. i.e. for 525 scans, $X = 6$ (0000 0110). Vertical Sync will occur every 262.5 scans producing two interlaced fields.

Range = 513 to 1023 Scans/Frame
(Odd counts only)

2) Non-Interlaced mode: Scans/Frame = $2X + 256$. i.e. for 262 scans, $X = 3$ (0000 0011).

Range = 256 to 766 Scans/Frame
(Even counts only)

In either mode Vertical Sync Width is THREE Horizontal Scans (3H).

Vertical Data Start
(\$DFC5, Bits 0-7)

An 8-bit code providing Vertical positioning with respect to Vertical Sync. The Data Row Counter is reset at Vertical Sync time and does not begin counting until the Scan Line Number selected by this code.

$N = 0$ to 255, DB0 = LSB

Data Rows/Frame
(\$DFC3, Bits 0-5)

A 6-bit code providing up to 64 data rows per frame.

DB0 = LSB

Last Data Row
(\$DFC6, Bits 0-5)

A 6-bit code which allows scrolling by defining the count of the last displayed data row on the screen.

Scans/Data Row
(\$DFC2, Bits 3-6)

A 4-bit code which provides up to 16 scan lines per data row.

20-24 LPSE (X=5)

03

F = 16 Lines

20 = 513 24 = 517

CP = 0114

CURSOR POSITIONING.

Cursor Character Address (\$DFCC, Bits 0-7)	An 8-bit code which defines the cursor position in a Data Row.
Cursor Row Address (\$DFCD, Bits 0-5)	A 6-bit code which defines the Data Row under which the cursor will be placed.
Up Scroll (\$DFCB)	Writing any code to this address increments the address of the first displayed data row. The LAST DATA ROW REGISTER (\$DFC6) IS NOT UPDATED and does not indicate the actual Last Data Row following an UP SCROLL command. To avoid confusion, we recommend the LAST DATA ROW REGISTER (\$DFC6) be used for scrolling instead of UP SCROLL.

CONTROL:

Reset (\$DFCA)	Writing any code to this address resets the display timing chain to the top left of the display. The Reset is held by the CRT Controller until released by the START command.
Start (\$DFCE)	Writing any code to this address after a RESET will release the CRT controller timing chain approximately one scan line later.

APPENDIX B
BIT ASSIGNMENT TABLE

	INTENSITY/GRAPHICS		DISPLAY CODE					
D800-DFBF	7	6						0
	HORIZONTAL LINE COUNT							
DFC0	7							0
	INTERLACE		H SYNC WIDTH			H SYNC DELAY		
DFC1	7	6			3	2		0
	SCANS/DATA ROW				CHARS/DATA ROW			
DFC2		6		3	2			0
	SKEW BITS		DATA ROWS/FRAME					
DFC3	7	6	5					0
	SCANS/FRAME							
DFC4	7							0
	VERTICAL DATA START							
DFC5	7							0
	LAST DISPLAYED DATA ROW							
DFC6			5					0
	CURSOR CHARACTER ADDRESS							
DFCC	7							0
	CURSOR ROW ADDRESS							
DFCD			5					0

APPENDIX C

RESTRICTIONS:

1. In Interlace Mode, the Horizontal Line Count (\$DFC0) must be even to insure that vertical Sync will occur precisely between Horizontal Sync Pulses.
2. In Interlace Mode, the number of Scan Lines/Data Row (\$DFC2) must be even.

APPENDIX D STANDARD CHARACTER SET

A3..A0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
A5..A4		D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0	D6..D0
000	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
001	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
010	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
011	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
100	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
101	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
110	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
111	R0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	R8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

= Shifted character. The character is shifted three rows to R3 at the top of the font and R11 at the bottom.

APPENDIX E
PROGRAMMING INFORMATION FOR
OPTIONAL SYMBOL GENERATOR ROM

The ELECTRIC WINDOW is designed to accommodate a 2708 or equivalent EPROM as an alternate Symbol Generator. Since 16 consecutive address location in the ROM define each symbol, a 2708 ROM permits up to 64 symbols. If more symbols are required the ELECTRIC WINDOW may be modified to accommodate a 2716 or equivalent EPROM which will permit up to 128 symbols. Refer to APPENDIX F for information.

Program the EPROM as illustrated in the following example:

ROM ADDRESS	D	D	D	D	D	D	D	D
	0	1	2	3	4	5	6	7
YY0	*	*	*	*	*	X	X	X
YY1	*	*	*	*	.	X	X	X
YY2	*	*	*	.	.	X	X	X
YY3	*	*	.	.	.	X	X	X
YY4	*	X	X	X
YY5	*	*	.	.	.	X	X	X
YY6	*	*	*	.	.	X	X	X
YY7	*	*	*	*	.	X	X	X
YY8	*	*	*	*	*	X	X	X
YY9	*	*	*	*	.	X	X	X
YYA	*	*	*	.	.	X	X	X
YYB	*	*	.	.	.	X	X	X
YYC	*	X	X	X
YYD	X	X	X
YYE	-	-	-	-	-	X	X	X
YYF	-	-	-	-	-	X	X	X

- YY = ROM SYMBOL ADDRESS
- * = SYMBOL BIT (LOGIC 1 - WHITE)
- .
- X = UNUSED BIT (DOES NOT AFFECT DISPLAY)
- = THESE BITS ARE AVAILABLE FOR SYMBOL GENERATION. REFER TO THE FOLLOWING DISCUSSION.

The number of consecutive ROM locations used during character display is controlled by the SCANS/DATA ROW register (\$DFC2). The procedures and programs described in this manual initialize the CRT Controller for 14 SCANS/DATA ROW. Consequently rows YYE and YYF in the above example will not be displayed unless the CRT Controller initialization is altered.

A character or symbol occupies 10 dot positions along the horizontal scan line. In the normal character mode 7 dots define the character image, the remaining 3 dots separate one character from the next.

In the Symbol (Graphics) mode the 10 horizontal dot positions are defined by the 5 data outputs from the symbol generator. Each data output defines 2 dots. There is no separation between adjacent symbols unless provided within the symbol itself.

APPENDIX F

Modifying the ELECTRIC WINDOW to accept a 2716 EPROM as the special Symbols Generator.

The ELECTRIC WINDOW is designed to accept a 2708 EPROM for special Symbols Generation. A 2708 permits up to 64 symbols (16 bytes/symbol). If more symbols are required, the ELECTRIC WINDOW circuit card may be modified to accept a 2716 EPROM which will permit up to 128 symbols.

NOTICE: There are TWO types of 2716 EPROMS! The two are NOT compatible.

- 1) the Single Voltage (+5 volts) 2716
- 2) the Triple Voltage (+12,+5,-5) TMS-2716

The following instructions provide modification information for both types.

Modifications for the single voltage 2716:

- 1) Cut the -5 volt circuit trace connecting to U10-21 on the solder side of the card. Make the cut near the 'feedthru' adjacent to U10-21. U10 is the 24-pin socket which holds the Special Symbols ROM.
- 2) Cut the 12 volt circuit trace to U10-19 on the solder side of the card. Make the cut near U10-19.
- 3) Connect U10-21 to U10-24 (+5v)
- 4) Connect U10-19 to U11-4

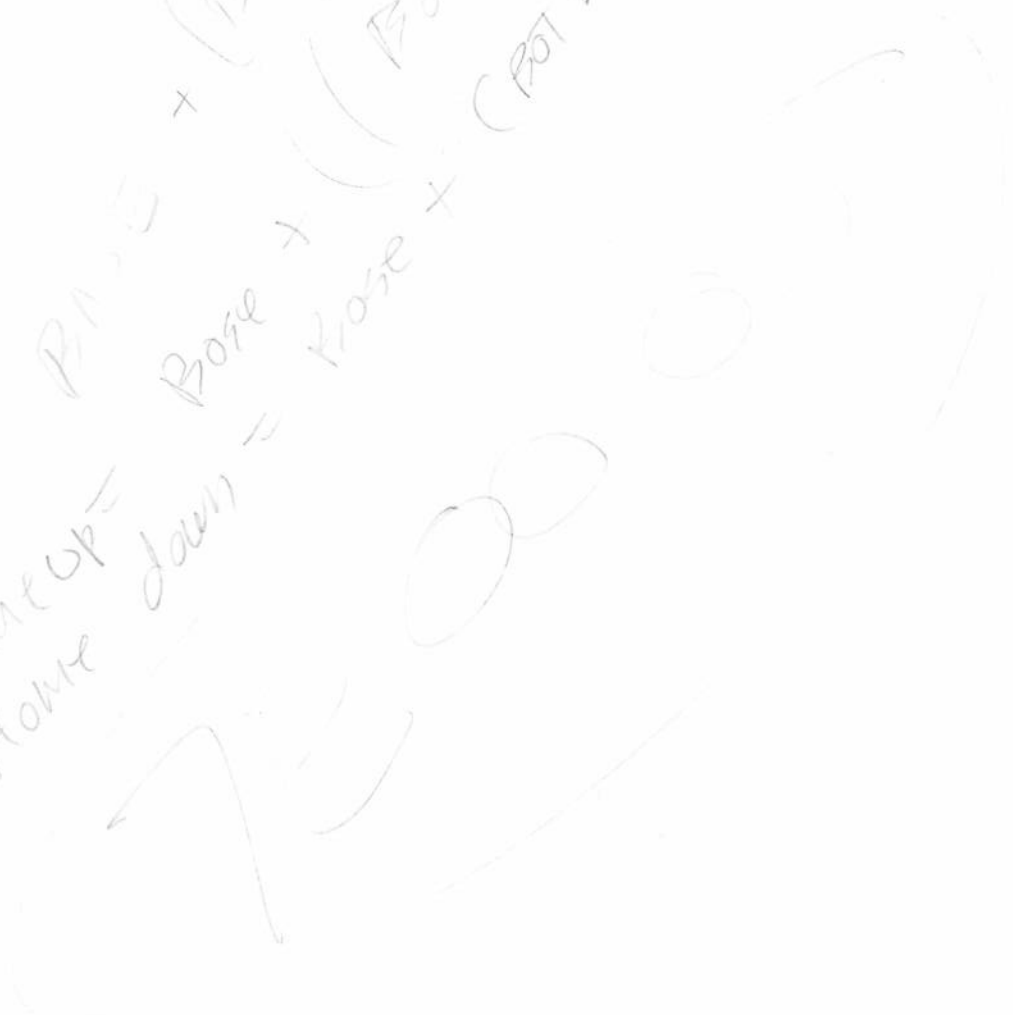
Modifications for triple voltage TMS-2716:

- 1) Cut the circuit trace connecting to U10-20 on the solder side of the card. Make the cut close to U10-20.
- 2) Cut the large circuit trace connecting to U10-18 on the solder side of the card. Make the cut close to U10-18.
- 3) Connect U10-20 to U11-4.
- 4) Connect U10-18 to the circuit TRACE formerly connected to U10-20 (U25-4)



00055	F080				ORG	\$F080	REVISION 1.12	
00056	F080	0002			NEXT	RMB	2	*NEXT LINE TO BE ERASED
00057	F082	0001			SCRL	RMB	1	*SCROLL FLAG
00058	F083	0001			MASK	RMB	1	*MASK FOR GRAPHICS
00059	F084	0001			CAPLOK	RMB	1	CAPS LOCK FLAG
00060	F085	000A			WINDCB	RMB	10	ELECTRIC WINDOW DCB
00061					*			
00062					*			
00063					*			
00064	F800				ORG	\$F800		
00065	F800	7E	F913	4	JMP	INIT		*INITIALIZE VIDEO
00066					*			
00067	F803	81	08	2	VIDOUT	CMPA	##8	*BACKSPACE
00068	F805	1027	00AA	6	LBEQ	BS		
00069	F809	81	0D	2	CMFA	##D		*CARRIAGE RETURN
00070	F80B	1027	00CE	6	LBEQ	CR		
00071	F80F	81	0A	2	CMFA	##A		*LINE FEED
00072	F811	1027	00AB	6	LBEQ	LF		
00073	F815	81	1A	2	CMFA	##1A		*CLEAR SCREEN?
00074	F817	1027	00C8	6	LBEQ	HUEF		
00075	F81B	81	0B	2	CMFA	##B		*VERTICAL TAB?
00076	F81D	1027	00AB	6	LBEQ	VT		
00077	F821	81	0F	2	CMFA	##F		*TURN ON GRAPHICS
00078	F823	1027	0082	6	LBEQ	GON		
00079	F827	81	0E	2	CMFA	##E		*TURN OFF GRAPHICS
00080	F829	1027	0082	6	LBEQ	GOFF		
00081	F82D	81	0C	2	CMFA	##C		*NON DESTRUCTIVE SPACE
00082	F82F	27	27	3	BEQ	FF		
00083	F831	81	19	2	CMFA	##19		ERASE TO EOF
00084	F833	1027	00AE	6	LBEQ	EEOF		
00085					*			
00086	F837	85	E0	2	BITA	##E0		*CONTROL CHARACTERS MAS
00087	F839	1027	0071	6	LBEQ	OUT		
00088	F83D	BA	F083	5	ORA	MASK		*GRAPHICS MASK
00089					*			
00090	F840	34	02	5	PSHS	A		*SAVE CHARACTER
00091	F842	B6	DFCD	5	LDA	CURV		*GET VERTICAL POSITION
00092	F845	8E	D800	3	LDX	##BASE		*GET START OF RAM
00093	F848	B6	DFCD	5	LDA	CURV		GET CURSOR VERTICAL
00094	F84B	C6	50	2	LDB	##80		LINE LENGTH
00095	F84D	3D		11	MUL			
00096	F84E	30	8B	8	LEAX	D, X		CHARACTER VERTICAL ADDR
00097	F850	F6	DFCC	5	LDB	CURH		
00098	F853	3A		3	ABX			CHARACTER HORIZONTAL AD
00099	F854	35	02	5	PULS	A		
00100	F856	A7	1E	5	STA	-2, X		
00101	F858	7C	DFCC	7	FF	INC	CURH	*MOVE CURSOR RIGHT ONE
00102	F85B	B6	DFCC	5	LDA	CURH		
00103	F85E	81	51	2	CMFA	##81		*END OF LINE?
00104	F860	2F	4C	3	BLE	OUT		
00105	F862	86	02	2	LDA	##2		*SCROLL ROUTINE
00106	F864	B7	DFCC	5	STA	CURH		*MOVE CURSOR
00107	F867	B6	DFCD	5	LDA	CURV		*GET VERTICAL POSITION
00108	F86A	81	0F	2	CMFA	##F		*BOTTOM OF SCREEN

$$\begin{aligned}
 \text{Home} &= \text{BASE} + (\text{BOTLINE} - 116) + 80 \\
 \text{Home up} &= \text{Base} + (\text{BOTLINE} + 1) \text{ (F)} * 2.5 \\
 \text{Home down} &= \text{Base} + (\text{BOTLINE} - 1) \text{ (F)} * 50
 \end{aligned}$$



00109	F86C	26	14	3		BNE	SCROLL	*NO
00110	F86E	8E	D800	3	FF1	LDX	#BASE	*SET UP NEXT LINE ERASE
00111	F871	BF	F080	6		STX	NEXT	
00112	F874	7F	F082	7		CLR	SCRL	*SET SCROLL FLAG
00113	F877	7C	F082	7		INC	SCRL	
00114	F87A	86	00	2		LDA	#0	
00115	F87C	B7	DFCD	5		STA	CURV	*MOVE CURSOR BOTTOM
00116	F87F	16	0008	5		LBRA	EOL	
00117	F882	7C	DFCD	7	SCROLL	INC	CURV	*MOVE CURSOR DOWN 1
00118	F885	B6	F082	5		LDA	SCRL	*SCROLL?
00119	F888	27	24	3		BEQ	OUT	*NO
00120	F88A	BE	F080	6	EOL	LDX	NEXT	*ERASE TO END OF LINE
00121	F88D	C6	50	2		LDB	#80	*1 LINE
00122	F88F	86	20	2		LDA	#'	*CLEAR
00123	F891	A7	80	6	EOL1	STA	0,X+	*CLEAR NEXT
00124	F893	5A		2		DECB		
00125	F894	26	FB	3		BNE	EOL1	*END?
00126	F896	BF	F080	6		STX	NEXT	*YES
00127	F899	7C	DFC6	7		INC	BOTLNE	*SCROLL
00128	F89C	B6	DFC6	5		LDA	BOTLNE	
00129	F89F	81	10	2		CMPA	##10	*OUT OF RAM?
00130	F8A1	26	05	3		BNE	SCRO1	*NO
00131	F8A3	86	00	2		LDA	#0	*YES *SCROLL FROM TOP M
00132	F8A5	B7	DFC6	5		STA	BOTLNE	
00133	F8A8	39		5	SCRO1	RTS		
00134				*				
00135				*				
00136	F8A9	86	80	2	GON	LDA	##80	*TURN ON GRAPHICS
00137	F8AB	B7	F083	5		STA	MASK	
00138	F8AE	39		5	OUT	RTS		
00139				*				
00140	F8AF	7F	F083	7	GOFF	CLR	MASK	*TURN OFF GRAPHICS
00141	F8B2	39		5		RTS		
00142				*				
00143	F8B3	B6	DFCC	5	BS	LDA	CURH	*GET POSITION
00144	F8B6	81	02	2		CMPA	#2	*AT LEFT MARGIN?
00145	F8B8	1027	FFF2	6		LBEQ	OUT	*YES IGNORE
00146	F8BC	7A	DFCC	7		DEC	CURH	*MOVE CURSOR LEFT 1
00147	F8BF	39		5		RTS		
00148				*				
00149	F8C0	B6	DFCD	5	LF	LDA	CURV	*GET VERTICAL POSITION
00150	F8C3	81	0F	2		CMPA	##F	*OUT OF RAM?
00151	F8C5	1026	FFB9	6		LBNE	SCROLL	*NO
00152	F8C9	16	FFA2	5		LBRA	FF1	*YES *SCROLL FROM TOP 0
00153				*				
00154	F8CC	B6	DFCD	5	VT	LDA	CURV	*GET POSITION
00155	F8CF	81	00	2		CMPA	#0	*TOP OF RAM?
00156	F8D1	27	04	3		BEQ	VT1	*YES *MOVE TO BOTTOM OF
00157	F8D3	7A	DFCD	7		DEC	CURV	*MOVE UP 1 LINE
00158	F8D6	39		5		RTS		
00159	F8D7	86	0F	2	VT1	LDA	##F	*SCROLL FROM TOP OF RAM
00160	F8D9	B7	DFCD	5		STA	CURV	
00161	F8DC	39		5		RTS		
00162				*				

18

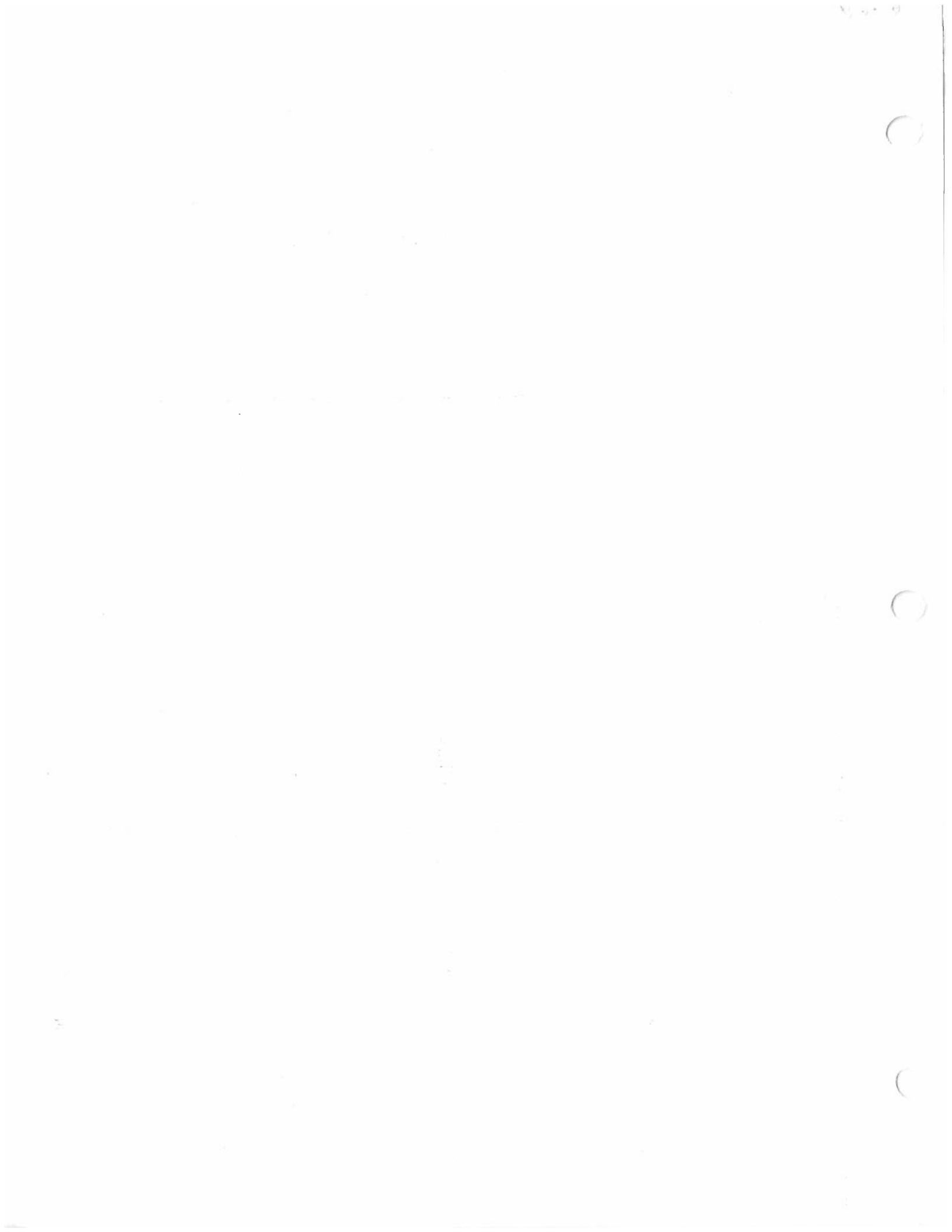
To
down

00163	F8DD	86	02	2	CR	LDA	#2	*CARRIAGE RETURN ROUTIN
00164	F8DF	B7	DFCC	5		STA	CURH	
00165	F8E2	39		5		RTS		
00166					*			
00167					*			
00168	F8E3	8D	0F	7	HUEF	BSR	HOUF	*CLEAR SCREEN
00169					*			
00170	F8E5	86	20	2	EEOF	LDA	#'	*CLEAR
00171	F8E7	8E	D800	3		LDX	#BASE	*START
00172	F8EA	A7	84	4	EEOF2	STA	O,X	*CLEAR NEXT
00173	F8EC	30	01	5		INX		
00174	F8EE	8C	DCFF	4		CPX	#BOT	*BOTTOM?
00175	F8F1	23	F7	3		BLS	EEOF2	*** 2.11 VERSION MODIFI
00176	F8F3	39		5		RTS		
00177					*			
00178					*			
00179					*			
00180	F8F4	86	02	2	HOUF	LDA	#2	*LEFT MARGIN
00181	F8F6	B7	DFCC	5		STA	CURH	*SET UP CURSOR
00182	F8F9	86	00	2		LDA	#0	*TOP OF MEMORY
00183	F8FB	B7	DFCD	5		STA	CURV	*CURSOR VERTICAL
00184	F8FE	86	0F	2		LDA	##F 24	*BOTTOM OF MEMORY
00185	F900	B7	DFC6	5		STA	BOTLINE	*SCROLL START
00186	F903	7F	F082	7		CLR	SCRL	*CLEAR SCROLL FLAG
00187	F906	39		5		RTS		
00188					*			
00189					*			
00190	F907	B6	F7FC	5	KEYIN	LDA	KEYPRT	SCAN KEYBOARD
00191	F90A	2A	FB	3		BPL	KEYIN	LOOP IF NO STROBE *** 2
00192	F90C	7D	F7FC	7	KEYIN1	TST	KEYPRT	WAIT FOR END OF STROBE
00193	F90F	2B	FB	3		BMI	KEYIN1	*** 2.11 VERSION MODIFI
00194	F911	43		2		COMA		*** 2.11 VERSION MOD: I
00195	F912	39		5		RTS		
00196					*			
00197					*			
00198	F913	7F	DFCA	7	INIT	CLR	BASE+\$7CA	*RESET VIDEO CONTROLL
00199	F916	7F	DFCE	7		CLR	BASE+\$7CE	*SET UP TIMING CHAIN
00200	F919	86	65	2		LDA	##65	
00201	F91B	B7	DFC0	5		STA	BASE+\$7C0	*HORIZONTAL LINE COUN
00202	F91E	86	64	2		LDA	##64	
00203	F920	B7	DFC1	5		STA	BASE+\$7C1	*INTERLACE H SYNC
00204	F923	86	6D	2		LDA	##6D	
00205	F925	B7	DFC2	5		STA	BASE+\$7C2	*ROW SCAN CHAR
00206	F928	86	8F	2		LDA	##8F	
00207	F92A	B7	DFC3	5		STA	BASE+\$7C3	*ROW FRAME
00208	F92D	86	03	2		LDA	##03	
00209	F92F	B7	DFC4	5		STA	BASE+\$7C4	*LINE FRAME
00210	F932	86	20	2		LDA	##20	
00211	F934	B7	DFC5	5		STA	BASE+\$7C5	*VERT START
00212	F937	86	0F	2		LDA	##0F	
00213	F939	B7	DFC6	5		STA	BASE+\$7C6	*LAST ROW
00214	F93C	7F	DFCA	7		CLR	BASE+\$7CA	*RESET VIDEO CONTROLL
00215	F93F	7F	DFCE	7		CLR	BASE+\$7CE	*START TIMING CHAIN
00216	F942	86	1A	2		LDA	##1A	*CLEAR SCREEN

00217	F944	17	FEBC	9	LBSR	VIDOUT	
00218	F947	7F	F083	7	CLR	MASK	*TURN OFF GRAPHICS
00219	F94A	7F	F084	7	CLR	CAPLOK	*UPPER/LOWER KEYBOARD
00220	F94D	73	F084	7	COM	CAPLOK	
00221	F950	7F	F082	7	CLR	SCRL	
00222	F953	BE	F3E8	6	LDX	DCBCHN	GET FIRST DCB
00223	F956	BF	F085	6	STX	WINDCB	SAVE LINK
00224	F959	8E	4557	3	LDX	##4557	SET ID TO 'EW'
00225	F95C	BF	F087	6	STX	WINDCB+2	
00226	F95F	8E	F97D	3	LDX	#WINDEX	SET DRIVER
00227	F962	BF	F089	6	STX	WINDCB+4	
00228	F965	4F		2	CLRA		RESET REST OF DCB
00229	F966	5F		2	CLRB		
00230	F967	FD	F08B	6	STD	WINDCB+6	
00231	F96A	FD	F08D	6	STD	WINDCB+8	
00232	F96D	8E	F085	3	LDX	#WINDCB	POINT TO DCB
00233	F970	BF	F3E8	6	STX	DCBCHN	ADD TO CHAIN
00234	F973	BF	F3EA	6	STX	CIDCB	SET UP FOR PSYMON
00235	F976	BF	F3EE	6	STX	CODCB	
00236	F979	BF	F3EC	6	STX	CEDCB	
00237	F97C	39		5	RTS		
00238				*			
00239				*			
00240	F97D	54		2	WINDEX	LSRB	CHECK FUNCTION
00241	F97E	1025	FF85	6	LBCS	KEYIN	GO IF INPUT
00242	F982	54		2	LSRB		
00243	F983	1025	FE7C	6	LBCS	VIDOUT	GO IF OUTPUT
00244	F987	54		2	LSRB		
00245	F988	25	01	3	BCS	VIDSTS	GO IF STATUS
00246	F98A	39		5	RTS		
00248	F98B	86	02	2	VIDSTS	LDA	#2
00249	F98D	7D	F7FC	7	TST	KEYPRT	ANY KEY PRESSED?
00250	F990	2B	01	3	BMI	VIDSTX	GO IF NOT
00251	F992	4C		2	INCA		SET INPUT BIT
00252	F993	39		5	VIDSTX	RTS	
00254				*			
00255		0000			END		

*Leax Windex
STX windcb#
RTS*

TOTAL ERRORS 00000
TOTAL WARNINGS 00000




```

*
* JULY 24 1979 4:00
*****
*
*
*
*          WINDEX V1.1
*        BY CLIFF RUSHING
*      A VIDEO DRIVER FOR THE PERCOM
*      ELECTRIC WINDOW
*
*
*****

```

```

          NAM      WINDEX
(D800)   BASE     EQU    $D800      *START OF VIDEO RAM
(DCFF)   BOT      EQU    BASE+$4FF  *END OF VIDEO RAM
(DFCD)   CURV     EQU    BASE+$7CD  *CURSER LINE REGISTER
(DFCC)   CURH     EQU    BASE+$7CC  *CURSOR HORIZONTAL REGISTER
(DFC6)   BOTLNE  EQU    BASE+$7C6  *BOTTOM LINE

```

```

          (A04A)
A04A    POINT    RMB    2           *POSITION TEMPORARY
A04C    NEXT     RMB    2           *NEXT LINE TO BE ERASED
A04E    SCRL     RMB    1           *SCROLL FLAG
A04F    FLAG     RMB    1           *LAST CHAR
A050    MASK     RMB    1           *GRAPHICS MASK
A051    CCPRT    RMB    1           *CONTROL CHARACTER PRINT
A052    TEMP     RMB    2           *TEMPORARY STORAGE

```

```

          (0100)
0100 7E 027B   JMP     TYPE
0103 7E 0238   JMP     INIT           *INITIALIZE VIDEO
0106 7E 0109   JMP     VIDOUT        *DISPLAY 1 CHAR ON SCREEN

```

```

0109 37        VIDOUT PSH B           *SAVE REGISTERS
010A FF A052   STX     TEMP

```

```

010D CE 021F   LDX     #TABLE        *POINT AT TABLE
0110 A1 00     TAB    CMP     A 0,X    *FOUND?
0112 26 07     BNE    TAB1         *NO
0114 EE 01     LDX     1,X
0116 6E 00     JMP     0,X           *YES DO IT
0118 7E 01CB   OUTC   JMP     OUT
011B 08        TAB1  INX
011C 08        INX
011D 08        INX
011E 6D 00     TST     0,X           *NOT FOUND?
0120 26 EE     BNE    TAB         *NO

```

```

0122 B5 A051   ON     BIT     A CCPRT    *MASK CONTROL CHARACTERS
0125 27 F1     BEQ     OUTC
0127 BA A050   ORA     A MASK         *GRAPHICS MASK

```

```

012A 36        PSH     A           *SAVE CHARACTER

```

012B	B6	DFCD		LDA A CURV	*GET VERT POSITION
012E	16			TAB	
012F	CE	D800		LDX #BASE	*GET START OF RAM
0132	FF	A04A		STX POINT	
0135	5A		S1	DEC B	*LAST VERTICAL ROW
0136	2B	0F		BMI S3	
0138	B6	A04B		LDA A POINT+1	
013B	8B	50		ADD A #80	
013D	24	03		BCC S2	
013F	7C	A04A		INC POINT	
0142	B7	A04B	S2	STA A POINT+1	
0145	20	EE		BRA S1	
0147	B6	DFCC	S3	LDA A CURH	*CALCULATE HORIZ POSITION
014A	BB	A04B		ADD A POINT+1	
014D	24	03		BCC S5	
014F	7C	A04A		INC POINT	
0152	80	02	S5	SUB A #2	*COMPENSATE FOR LEFT MARGIN
0154	24	03		BCC S6	
0156	7A	A04A		DEC POINT	
0159	B7	A04B	S6	STA A POINT+1	
015C	FE	A04A		LDX POINT	
015F	32			PUL A	*RESTORE CHARACTER
0160	A7	00		STA A 0,X	*DISPLAY CHAR ON SCREEN
0162	7C	DFCC	FF	INC CURH	*MOVE CURSOR LEFT
0165	B6	DFCC		LDA A CURH	
0168	81	51		CMP A #81	*END OF LINE?
016A	2F	47		BLE SCROL	
016C	86	02		LDA A #2	*SCROLL ROUTINE
016E	B7	DFCC		STA A CURH	*MOVE CURSOR
0171	B6	DFCD		LDA A CURV	*GET VERTICAL POSITION
0174	81	0F		CMP A #\$F	*BOTTOM OF SCREEN
0176	26	14		BNE SCROLL	*NO
0178	CE	D800	FF1	LDX #BASE	*SET UP NEXT LINE ERASE
017B	FF	A04C		STX NEXT	
017E	7F	A04E		CLR SCRL	*SET SCROLL FLAG
0181	7C	A04E		INC SCRL	
0184	86	00		LDA A #0	
0186	B7	DFCD		STA A CURV	*MOVE CURSOR BOTTOM
0189	7E	0194		JMP EOL	
018C	7C	DFCD	SCROLL	INC CURV	*MOVE CURSOR DOWN 1
018F	B6	A04E		LDA A SCRL	*SCROLL?
0192	27	1F		BEQ SCROL	
0194	FE	A04C	EOL	LDX NEXT	*ERASE TO END OF LINE
0197	C6	50		LDA B #80	*1 LINE
0199	86	20		LDA A #'	*CLEAR
019B	A7	00	EOL1	STA A 0,X	*CLEAR NEXT
019D	08			INX	
019E	5A			DEC B	
019F	26	FA		BNE EOL1	*END?
01A1	FF	A04C		STX NEXT	*YES
01A4	7C	DFC6		INC BOTLNE	*SCROLL
01A7	B6	DFC6		LDA A BOTLNE	
01AA	81	10		CMP A #\$10	*OUT OF RAM?
01AC	26	05		BNE SCROL	*NO
01AE	86	00		LDA A #0	*YES SCROLL
01B0	B7	DFC6		STA A BOTLNE	
01B3	20	16	SCROL	BRA OUT	

*

```

*
01B5 86 80      GON      LDA A #\$80      *TURN ON GRAPHICS
01B7 B7 A050    STA A MASK
01BA 20 0F      BRA      OUT
*
01BC 7F A050    GOFF     CLR      MASK      *TURN OFF GRAPHICS
01BF 20 0A      BRA      OUT
*
01C1 B6 DFCC    BS       LDA A CURH      *GET POSITION
01C4 81 02      CMP A #2      *AT LEFT MARGIN?
01C6 27 03      BEQ      OUT      *YES IGNORE
01C8 7A DFCC    DEC     CURH      *MOVE CURSOR LEFT 1
01CB 33          OUT     PUL B
01CC FE A052    LDX     TEMP
01CF 39          RTS
*
01D0 B6 DFCD    LF       LDA A CURV      *GET VERTICAL POSITION
01D3 81 0F      CMP A #\$F     *OUT OF RAM?
01D5 26 B5      BNE     SCROLL  *NO
01D7 7E 0178    JMP     FF1     *YES *SCROLL FROM TOP OF RAM
*
01DA B6 DFCD    VT       LDA A CURV      *GET POSITION
01DD 81 00      CMP A #0      *TOP OF RAM?
01DF 27 05      BEQ     VT1     *YES *MOVE TO BOTTOM OF RAM
01E1 7A DFCD    DEC     CURV      *MOVE UP 1 LINE
01E4 20 E5      BRA     OUT
01E6 86 0F      VT1     LDA A #\$F      *SCROLL FROM TOP OF RAM
01E8 B7 DFCD    STA A CURV
01EB 20 DE      BRA     OUT
*
01ED 86 02      CR       LDA A #2      *CARRIAGE RETURN ROUTINE
01EF B7 DFCC    STA A CURH
01F2 20 D7      BRA     OUT
*
*
01F4 8D 16      HUEF    BSR     HOUP1   *CLEAR SCREEN
*
01F6 8D 02      EEOF    BSR     EEOF1   *ERASE TO END OF FRAME
01F8 20 D1      BRA     OUT
*
01FA 86 20      EEOF1   LDA A #'      *CLEAR
01FC CE D800    LDX     #BASE   *START
01FF A7 00      EEOF2   STA A 0,X      *CLEAR NEXT
0201 08          INX
0202 8C DD00    CPX     #BOT+1  *BOTTOM?
0205 26 F8      BNE     EEOF2
0207 39          RTS
*
0208 8D 02      HOUP    BSR     HOUP1   *HOME UP
020A 20 BF      BRA     OUT
*
*
020C 86 02      HOUP1   LDA A #2      *LEFT MARGIN
020E B7 DFCC    STA A CURH      *SET UP CURSOR
0211 86 00      LDA A #0      *TOP OF MEMORY
0213 B7 DFCD    STA A CURV      *CURSOR VERTICAL
0216 86 0F      LDA A #\$F     *BOTTOM OF MEMORY
0218 B7 DFCD    STA A BOTLNE   *SCROLL START

```

```

021B 7F A04E      CLR      SCRL      *CLEAR SCROLL FLAG
021E 39           RTS

*
*
*
(021F)           TABLE EQU      *
021F 08           FCB      $8 *BACKSPACE
0220 01 C1       FDB      BS
0222 0D           FCB      $D *CARRIAGE RETURN
0223 01 ED       FDB      CR
0225 0A           FCB      $A *LINE FEED
0226 01 D0       FDB      LF
0228 1A           FCB      $1A *CLEAR SCREEN?
0229 01 F4       FDB      HUEF
022B 0B           FCB      $B *VERTICAL TAB?
022C 01 DA       FDB      VT
022E 0F           FCB      $F *TURN ON GRAPHICS
022F 01 B5       FDB      GON
0231 0E           FCB      $E *TURN OFF GRAPHICS
0232 01 BC       FDB      GOFF
0234 0C           FCB      $C *NON DESTRUCTIVE SPACE
0235 01 62       FDB      FF
0237 00           FCB      0

*
*
*
0238 4F           INIT     CLR      A
0239 B7 DFCA      STA      A BASE+$7CA *RESET CONTROLLER
023C B7 DFCE      STA      A BASE+$7CE *SET UP TIMING CHAIN
023F 86 65        LDA      A #$65
0241 B7 DF00      STA      A BASE+$7C0 *HORIZONTAL LINE COUNT
0244 86 64        LDA      A #$64
0246 B7 DF01      STA      A BASE+$7C1 *INTERLACE H SYNC
0249 86 6D        LDA      A #$6D
024B B7 DF02      STA      A BASE+$7C2 *ROW SCAN CHAR
024E 86 8F        LDA      A #$8F
0250 B7 DF03      STA      A BASE+$7C3 *ROW FRAME
0253 86 03        LDA      A #$03
0255 B7 DF04      STA      A BASE+$7C4 *LINE FRAME
0258 86 20        LDA      A #$20
025A B7 DF05      STA      A BASE+$7C5 *VERT START
025D 86 0F        LDA      A #$0F
025F B7 DF06      STA      A BASE+$7C6 *LAST ROW
0262 86 00        LDA      A #0
0264 B7 DF0A      STA      A BASE+$7CA *RESET CONTROLLER
0267 B7 DF0E      STA      A BASE+$7CE *START TIMING CHAIN
026A 86 1A        LDA      A #$1A *CLEAR SCREEN
026C BD 0109      JSR      VIDOUT
026F 7F A050      CLR      MASK      *TURN OFF GRAPHICS
0272 86 E0        LDA      A #$E0
0274 B7 A051      STA      A CCPRT    *DON'T PRINT CNTRL CODE
0277 7F A04E      CLR      SCRL
027A 39           RTS

*
027B BD 0238      TYPE   JSR      INIT
027E BD E1AC      TYPE1  JSR      $E1AC
0281 BD 0109      JSR      VIDOUT
0284 20 F8        BRA      TYPE1

```

```

*
* JULY 5 1979 10:00
*****
*
*
* LIFE
*
* BY CLIFF RUSHING
* 1820 EDNA ST
* ARLINGTON, TX
* 76010
*
*****

```

```

*
* NAM LIFE
*
* THIS IS A 6800 IMPLEMENTATION OF JOHN CONWAY'S GAME
* OF "LIFE". FOR MORE INFORMATION CONCERNING THE RULES
* OF THE GAME REFER TO:
* SCIENTIFIC AMERICAN OCTOBER 1970
* BYTE MAGAZINE SEPTEMBER 1975
* BYTE MAGAZINE DECEMBER 1978
* THIS PROGRAM IS WRITTEN FOR THE SWTP 6800 COMPUTER
* USING THE PERCOM "ELECTRIC WINDOW" VIDEO DISPLAY.
* THE PROGRAM PROVIDES ALL NECESSARY INSTRUCTIONS AND
* INCLUDES AN EASY TO USE UTILITY FOR "SEEDING" THE
* DISPLAY. THIS PROGRAM REQUIRES LESS THAN 3K AND
* TAKES ABOUT 6 SECONDS / GENERATION. THE PROGRAM IS
* AVAILIABLE THROUGH PERCOM USER'S GROUP (DISKETTE #3)
* COPYRIGHT (C) 1979
* PERCOM DATA CO. INC.
* ALL RIGHTS RESERVED
*

```

```

(0020)
0020 POINT RMB 2 *DISPLAY POINTER
0022 POINT2 RMB 2 *MATRIX POINTER
0024 XTEMP RMB 2 *X TEMPORARY STORAGE
0026 XTEMP1 RMB 2 *X TEMPORARY STORAGE
0028 NABOR RMB 1 *NUMBER OF NEIGHBOR'S
0029 CNT RMB 1 *NUMBER OF LIVE CELLS
(D800) BASE EQU $D800 *START OF VIDEO RAM
(E07E) PSTR EQU $E07E *PRINT STRING
(E1AC) IN EQU $E1AC *INPUT A CHARACTER
(8004) PORT EQU $8004 *CONTROL PORT
(0281) CKBRK EQU CKKBR *CHECK BREAK
(E0E3) MON EQU $E0E3
*
(004F) U EQU 'O *UP KEY
(004B) L EQU 'K *LEFT KEY
(003B) R EQU ';' *RIGHT KEY
(002C) D EQU ',' *DOWN KEY
*
(0100)
0100 8E A042 INIT LDS #$A042
0103 4F CLR A
0104 B7 DFCA STA A BASE+$7CA *RESET VIDEO CONTROLLER
0107 B7 DFCE STA A BASE+$7CE *SET UP TIMING CHAIN

```

```

010A 86 65          LDA A #$65
010C B7 DFC0        STA A BASE+$7C0 *HORIZONTAL LINE COUNT
010F 86 64          LDA A #$64
0111 B7 DFC1        STA A BASE+$7C1 *INTERLACE H SYNC
0114 86 6D          LDA A #$6D
0116 B7 DFC2        STA A BASE+$7C2 *ROW SCAN CHAR
0119 86 8F          LDA A #$8F
011B B7 DFC3        STA A BASE+$7C3 *ROW FRAME
011E 86 03          LDA A #$03
0120 B7 DFC4        STA A BASE+$7C4 *LINE FRAME
0123 86 20          LDA A #$20
0125 B7 DFC5        STA A BASE+$7C5 *VERT START
0128 86 0F          LDA A #$0F
012A B7 DFC6        STA A BASE+$7C6 *LAST ROW
012D 86 00          LDA A #0
012F B7 DFCA        STA A BASE+$7CA *RESET VIDEO CONTROLLER
0132 B7 DFCE        STA A BASE+$7CE *START TIMING CHAIN
0135 86 FF          LDA A #$FF
0137 B7 DFCC        STA A BASE+$7CC
013A 8D 29          BSR CLEAR

*
013C DE 20          GETCMD LDX POINT *TURN ON CURSOR
013E E6 00          LDA B 0,X
0140 CA 80          ORA B #$80
0142 E7 00          STA B 0,X
0144 BD E1AC        JSR IN *GET COMMAND
0147 DE 20          LDX POINT *TURN OFF CURSOR
0149 E6 00          LDA B 0,X
014B C4 7F          AND B #$7F
014D E7 00          STA B 0,X
014F CE 029A        LDX #TABLE *POINT AT TABLE
0152 A1 00          G1 CMP A 0,X *MATCH?
0154 27 09          BEQ DO *YES
0156 08             INX *POINT AT NEXT
0157 08             INX
0158 08             INX
0159 6D 00          TST 0,X *END OF TABLE?
015B 26 F5          BNE G1 *NO
015D 20 DD          BRA GETCMD *NEXT COMMAND

*
015F EE 01          DO LDX 1,X *POINT AT ROUTINE
0161 AD 00          JSR 0,X *DO IT
0163 20 D7          BRA GETCMD *NEXT COMMAND

*
0165 CE D800        CLEAR LDX #BASE *POINT AT TOP LEFT
0168 86 20          LDA A #$20 *FILL WITH SPACES
016A A7 00          S1 STA A 0,X + *FILL
016C 08             INX *NEXT
016D 8C DCFE        CPX #BASE+$4FE *END?
0170 26 F8          BNE S1 *NO

*
0172 CE 02B6        LDX #PROMPT *PRINT PROMPT
0175 BD E07E        JSR PSTR
0178 BD E1AC        JSR IN *WAIT TO CONTINUE

*
017B CE DA56        LDX #BASE+$256 *POINT AT CENTER OF SCREEN
017E DF 20          STX POINT
0180 39             RTS

```

```

*
0181 DE 20 RIGHT LDX POINT *GET POINTER
0183 08 INX *MOVE RIGHT 1
0184 8C DCFE CPX #BASE+$4FF *END OF SCREEN?
0187 27 03 BEQ R1 *YES
0189 DF 20 STX POINT
018B 39 RTS

*
018C CE D800 R1 LDX #BASE *POINT AT TOP LEFT
018F DF 20 STX POINT
0191 39 RTS

*
0192 DE 20 LEFT LDX POINT *GET POINTER
0194 09 DEX *MOVE LEFT
0195 8C D7FE CPX #BASE-1 *END OF SCREEN?
0198 27 03 BEQ L1 *YES
019A DF 20 STX POINT
019C 39 RTS

*
019D CE DCFE L1 LDX #BASE+$4FF *POINT AT BOTTOM RIGHT
01A0 DF 20 STX POINT
01A2 39 RTS

*
01A3 C6 50 DOWN LDA B #80 *LINE COUNT
01A5 DE 20 LDX POINT *GET POINTER
01A7 08 D1 INX *MOVE RIGHT 1
01A8 8C DD00 CPX #BASE+$4FF+1 *END OF SCREEN?
01AB 27 06 BEQ D2 *YES
01AD 5A DEC B *FINISHED?
01AE 26 F7 BNE D1 *NO
01B0 DF 20 STX POINT
01B2 39 RTS

*
01B3 CE D800 D2 LDX #BASE *POINT AT TOP LEFT
01B6 20 EF BRA D1

*
01B8 C6 50 UP LDA B #80 *LINE COUNT
01BA DE 20 LDX POINT *GET POINTER
01BC 8C D800 U1 CPX #BASE *END OF SCREEN?
01BF 27 07 BEQ U2 *YES
01C1 09 DEX *MOVE LEFT 1
01C2 5A DEC B *FINISHED?
01C3 26 F7 BNE U1 *NO
01C5 DF 20 STX POINT
01C7 39 RTS

*
01C8 CE DD00 U2 LDX #BASE+$500 *POINT AT BOTTOM RIGHT
01CB 20 EF BRA U1

*
01CD DE 20 ENTER LDX POINT *GET POINTER
01CF 86 2A LDA A #' *CELL
01D1 A7 00 STA A 0,X *DISPLAY
01D3 39 RTS

*
01D4 DE 20 DELETE LDX POINT *GET POINTER
01D6 86 20 LDA A #$20 *DEAD
01D8 A7 00 STA A 0,X *DISPLAY
01DA 39 RTS

```

```

*
*****
*
* ON ENTRY X POINTS AT SCREEN
* B HAS COUNT TO MOVE FORWARD
* A HAS CHARACTER TO BE COMPARED
*
01DB 8C DCFE FWD CPX #BASE+$4FF *END OF SCREEN?
01DE 26 03 BNE FWD1 *NO
01E0 CE D7FF LDX #BASE-1 *POINT AT TOP RIGHT
01E3 08 FWD1 INX *MOVE RIGHT 1
01E4 5A DEC B *FINISHED?
01E5 26 F4 BNE FWD *NO
01E7 A1 00 FWD2 CMP A 0,X *OLD CELL LIVE?
01E9 26 03 BNE FWD3 *NO
01EB 7C 0028 INC NABOR *INCREMENT NEIGHBOR COUNT
01EE 39 FWD3 RTS
*
01EF CE 0425 FIN LDX #MAT *SET UP MATRIX POINTER
01F2 DF 22 STX POINT2
01F4 CE D800 LDX #BASE *SET UP DISPLAY POINTER
01F7 DF 20 STX POINT
01F9 7F 0029 CLR CNT *CLEAR LIVE COUNT
*
01FC 86 2A LIVE LDA A #' *CELL
01FE 7F 0028 CLR NABOR *CLEAR COUNT OF NEIGHBOR'S
*
*BACK UP TO TOP LEFT
0201 C6 51 LDA B #81 *LINE COUNT + 1
0203 8C D800 LI1 CPX #BASE *END OF SCREEN?
0206 26 03 BNE LI2 *NO
0208 CE DD00 LDX #BASE+$500 *POINT AT BOTTOM RIGHT
020B 09 LI2 DEX *MOVE RIGHT 1
020C 5A DEC B *FINISHED?
020D 26 F4 BNE LI1 *NO
*
020F 8D D6 BSR FWD2 *TOP ROW
0211 C6 01 LDA B #1
0213 8D C6 BSR FWD *MOVE TO NEXT AND TEST
0215 C6 01 LDA B #1
0217 8D C2 BSR FWD *MOVE TO NEXT AND TEST
*
0219 C6 4E LDA B #78 *MIDDLE ROW
021B 8D BE BSR FWD *MOVE TO NEXT AND TEST
021D C6 02 LDA B #2
021F 8D BA BSR FWD *MOVE TO NEXT AND TEST
*
0221 C6 4E LDA B #78 *BOTTOM ROW
0223 8D B6 BSR FWD *MOVE TO NEXT AND TEST
0225 C6 01 LDA B #1
0227 8D B2 BSR FWD *MOVE TO NEXT AND TEST
0229 C6 01 LDA B #1
022B 8D AE BSR FWD *MOVE TO NEXT AND TEST
*
022D DE 22 LDX POINT2 *BUILD 2ND MATRIX
022F 96 28 LDA A NABOR
0231 A7 00 STA A 0,X *SAVE NUMBER OF NEIGHBOR'S
0233 08 INX

```



```

0234 DF 22          STX  POINT2
0236 8D 49          BSR  CKBRK      *BREAK?
0238 DE 20          LDX  POINT      *INCREMENT DISPLAY POINTER
023A 08             INX
023B DF 20          STX  POINT
023D 8C DD00        CPX  #BASE+$500 *FINISHED?
0240 26 BA          BNE  LIVE      *NO
      *
0242 CE D800        NEWGEN LDX  #BASE      *SET UP POINTERS
0245 DF 26          STX  XTEMP1
0247 CE 0425        LDX  #MAT
      *
024A A6 00          NEW   LDA  A 0,X      *GET NEIGHBOR COUNT
024C 08             INX
024D DF 24          STX  XTEMP
      *
024F DE 26          LDX  XTEMP1
0251 E6 00          LDA  B 0,X
0253 C1 2A          CMP  B #'*      *LIVE?
0255 26 04          BNE  NEW1      *NO
0257 81 02          CMP  A #2
0259 27 04          BEQ  NEW2      *REMAINS LIVE
025B 81 03          NEW1  CMP  A #3
025D 26 09          BNE  NEW3      *DIES
025F 86 2A          NEW2  LDA  A #'*
0261 A7 00          STA  A 0,X
0263 7C 0029        INC  CNT      *INC NUMBER OF LIVE CELLS
0266 20 04          BRA  NEW4
0268 86 20          NEW3  LDA  A #$20      *GET RID OF OLD CELLS
026A A7 00          STA  A 0,X
      *
026C 08             NEW4  INX
026D DF 26          STX  XTEMP1
026F DE 24          LDX  XTEMP
0271 8C 0925        CPX  #MAT+$500 *END?
0274 26 D4          BNE  NEW      *NO
      *
0276 7D 0029        TST  CNT      *ANY LIVE CELLS?
0279 27 03          BEQ  YY3      *NO
027B 7E 01EF        JMP  FIN      *DO NEXT GENERATION
027E 7E 0100        YY3   JMP  INIT
      *
0281 B6 8004        CKKBR LDA  A PORT      *GET STATUS
0284 47             ASR  A      *DATA?
0285 25 01          BCS  CK1      *YES
0287 39             RTS
0288 B6 8005        CK1   LDA  A PORT+1      *GET DATA
028B 81 1B          CMP  A #$1B      *ABORT?
028D 26 03          BNE  CK2      *NO
028F 7E E0E3        JMP  $E0E3
0292 81 03          CK2   CMP  A #3      *RESTART?
0294 26 03          BNE  CK3      *NO
0296 7E 0100        JMP  INIT
0299 39             CK3   RTS
      *
      (029A)
029A 4F             TABLE EQU  *
029B 01 B8          FCB  U
          FDB  UP

```

```

029D 4B          FCB  L
029E 01 92      FDB  LEFT
02A0 2C          FCB  D
02A1 01 A3      FDB  DOWN
02A3 3B          FCB  R
02A4 01 81      FDB  RIGHT
02A6 0D          FCB  $D
02A7 01 EF      FDB  FIN
02A9 03          FCB  3
02AA 01 00      FDB  INIT *ENTER NEW FIELD
02AC 1B          FCB  $1B
02AD E0 E3      FDB  MON
02AF 20          FCB  $20
02B0 01 CD      FDB  ENTER
02B2 18          FCB  $18
02B3 01 D4      FDB  DELETE
02B5 00          FCB  0

*
02B6 0D          PROMPT FCB  $D,$A
02B7 0A
02B8 20          FCC  /                LIFE/
02B9 20 20
02BB 20 20
02BD 20 20
02BF 20 20
02C1 20 20
02C3 20 20
02C5 20 20
02C7 4C 49
02C9 46 45
02CB 0D          FCB  $D,$A,$A
02CC 0A 0A
02CE 45          FCC  /ENTER YOUR PATTERN USING/
02CF 4E 54
02D1 45 52
02D3 20 59
02D5 4F 55
02D7 52 20
02D9 50 41
02DB 54 54
02DD 45 52
02DF 4E 20
02E1 55 53
02E3 49 4E
02E5 47
02E6 20          FCC  / A (SPACE) FOR LIVE CELLS/
02E7 41 20
02E9 28 53
02EB 50 41
02ED 43 45
02EF 29 20
02F1 46 4F
02F3 52 20
02F5 4C 49
02F7 56 45
02F9 20 43
02FB 45 4C
02FD 4C 53

```

```
02FF 0D          FCB  $D,$A
0300 0A
0301 45          FCC  /ENTER A (RETURN) WHEN FINISHED/
0302 4E 54
0304 45 52
0306 20 41
0308 20 28
030A 52 45
030C 54 55
030E 52 4E
0310 29 20
0312 57 48
0314 45 4E
0316 20 46
0318 49 4E
031A 49 53
031C 48 45
031E 44
031F 0D          FCB  $D,$A
0320 0A
0321 45          FCC  /ENTER A (CONTROL C) TO START OVER/
0322 4E 54
0324 45 52
0326 20 41
0328 20 28
032A 43 4F
032C 4E 54
032E 52 4F
0330 4C 20
0332 43 29
0334 20 54
0336 4F 20
0338 53 54
033A 41 52
033C 54 20
033E 4F 56
0340 45 52
0342 0D          FCB  $D,$A
0343 0A
0344 45          FCC  /ENTER A (ESCAPE) TO ABORT/
0345 4E 54
0347 45 52
0349 20 41
034B 20 28
034D 45 53
034F 43 41
0351 50 45
0353 29 20
0355 54 4F
0357 20 41
0359 42 4F
035B 52 54
035D 0D          FCB  $D,$A
035E 0A
035F 45          FCC  /ENTER A (CONTROL X) TO DELETE/
0360 4E 54
0362 45 52
0364 20 41
```

0366	20	28		
0368	43	4F		
036A	4E	54		
036C	52	4F		
036E	4C	20		
0370	58	29		
0372	20	54		
0374	4F	20		
0376	44	45		
0378	4C	45		
037A	54	45		
037C	0D		FCB	\$D,\$A,\$A
037D	0A	0A		
037F	20		FCC	/ /
0380	20	20		
0382	20	20		
0384	20	20		
0386	4F		FCB	U
0387	20		FCC	/ (/
0388	20	20		
038A	20	20		
038C	20	20		
038E	28			
038F	4F		FCB	U
0390	29		FCC	/) TO MOVE UP/
0391	20	54		
0393	4F	20		
0395	4D	4F		
0397	56	45		
0399	20	55		
039B	50			
039C	0D		FCB	\$D,\$A
039D	0A			
039E	20		FCC	/ /
039F	20	20		
03A1	20	20		
03A3	20			
03A4	4B		FCB	L,\$20,R
03A5	20	3B		
03A7	20		FCC	/ (/
03A8	20	20		
03AA	20	20		
03AC	20	28		
03AE	3B		FCB	R
03AF	29		FCC	/) TO MOVE RIGHT/
03B0	20	54		
03B2	4F	20		
03B4	4D	4F		
03B6	56	45		
03B8	20	52		
03BA	49	47		
03BC	48	54		
03BE	0D		FCB	\$D,\$A
03BF	0A			
03C0	20		FCC	/ /
03C1	20	20		
03C3	20	20		
03C5	20	20		

03C7	2C	FCB	D	
03C8	20	FCC	/	(/
03C9	20 20			
03CB	20 20			
03CD	20 20			
03CF	28			
03D0	2C	FCB	D	
03D1	29	FCC	/)	TO MOVE DOWN/
03D2	20 54			
03D4	4F 20			
03D6	4D 4F			
03D8	56 45			
03DA	20 44			
03DC	4F 57			
03DE	4E			
03DF	0D	FCB	\$D,\$A	
03E0	0A			
03E1	20	FCC	/	(/
03E2	20 20			
03E4	20 20			
03E6	20 20			
03E8	20 20			
03EA	20 20			
03EC	20 20			
03EE	20 20			
03F0	28			
03F1	4B	FCB	L	
03F2	29	FCC	/)	TO MOVE LEFT/
03F3	20 54			
03F5	4F 20			
03F7	4D 4F			
03F9	56 45			
03FB	20 4C			
03FD	45 46			
03FF	54			
0400	0D	FCB	\$D,\$A,\$A	
0401	0A 0A			
0403	45	FCC	/ENTER ANY CHARACTER TO CONTINUE/	
0404	4E 54			
0406	45 52			
0408	20 41			
040A	4E 59			
040C	20 43			
040E	48 41			
0410	52 41			
0412	43 54			
0414	45 52			
0416	20 54			
0418	4F 20			
041A	43 4F			
041C	4E 54			
041E	49 4E			
0420	55 45			
0422	0D	FCB	\$D,\$A,4	
0423	0A 04			
0425		MAT	RMB	2000

*

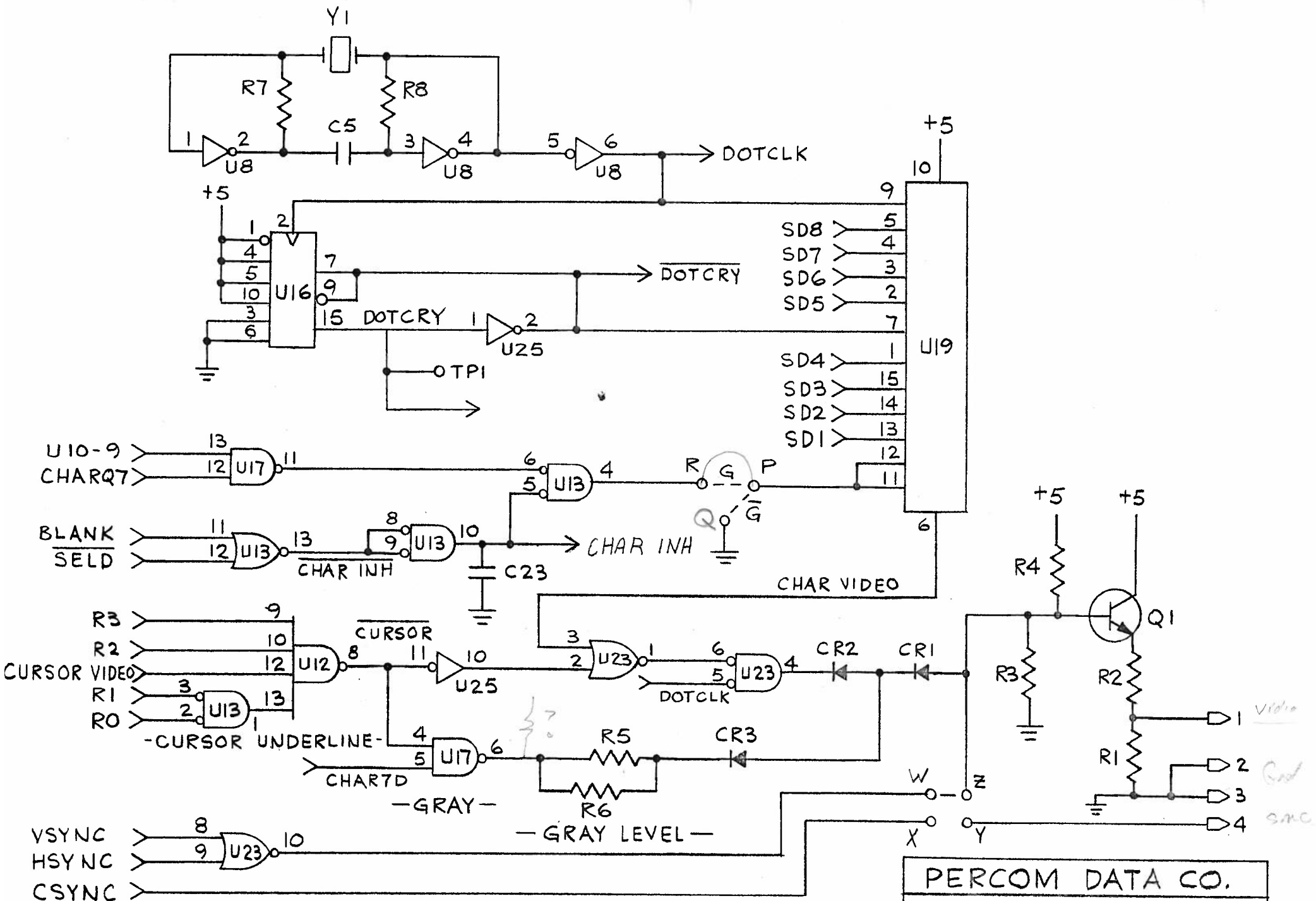
MAT

*

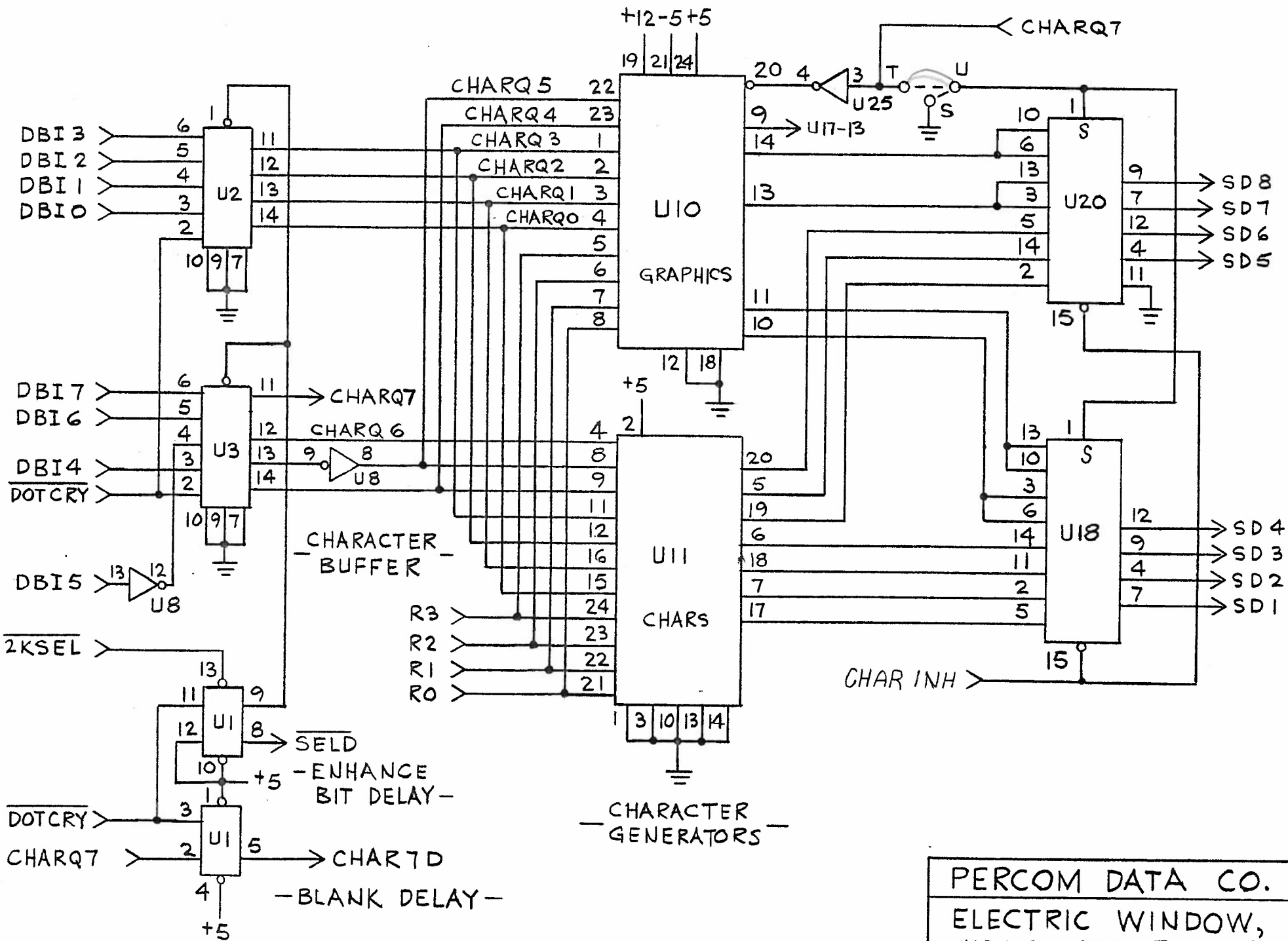
LIFE

PERCOM 6800 ASSEMBLER PAGE 10

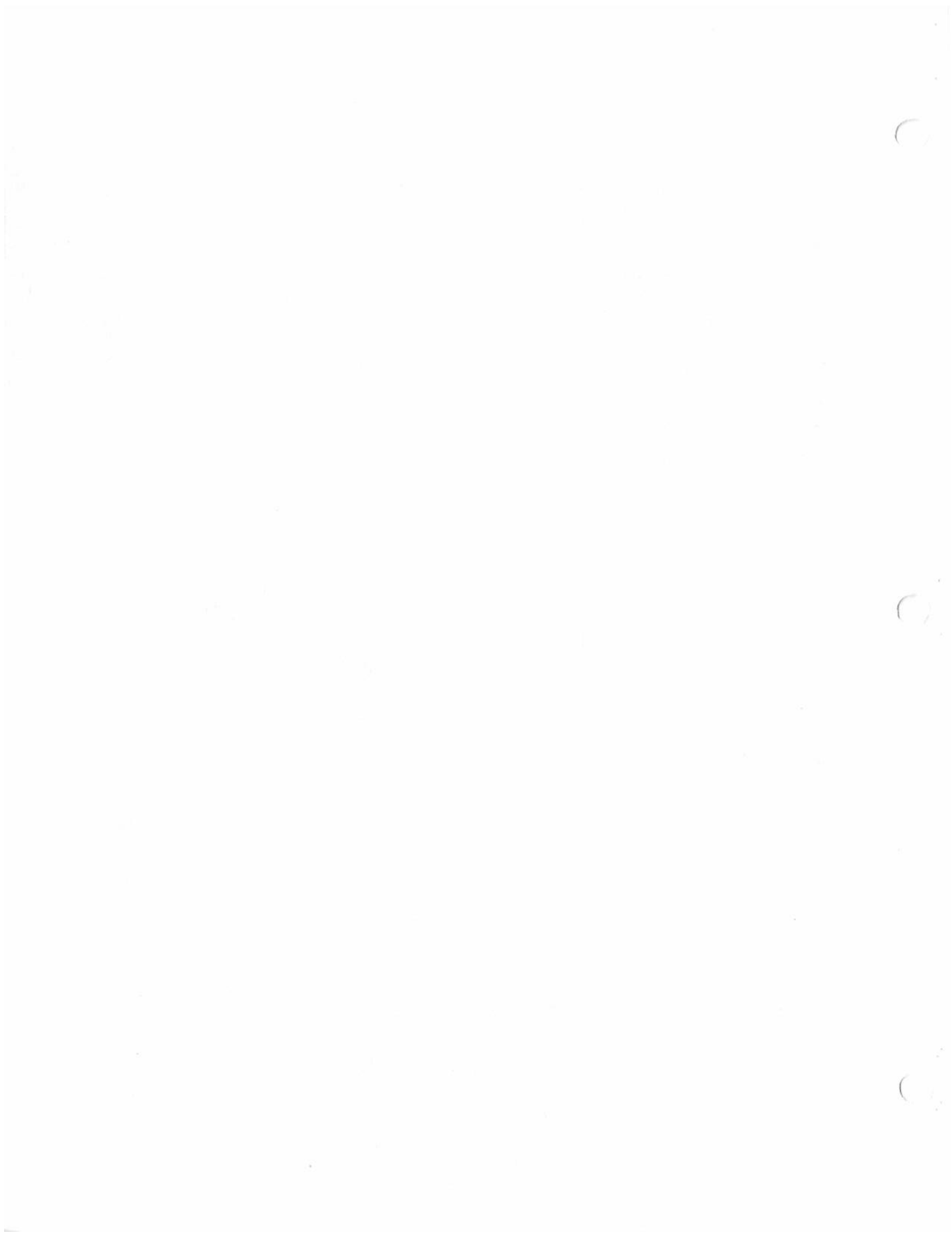
(A048)	ORG	\$A048
A048 01 00	FDB	INIT
	END	
00	ERROR(S) DETECTED	

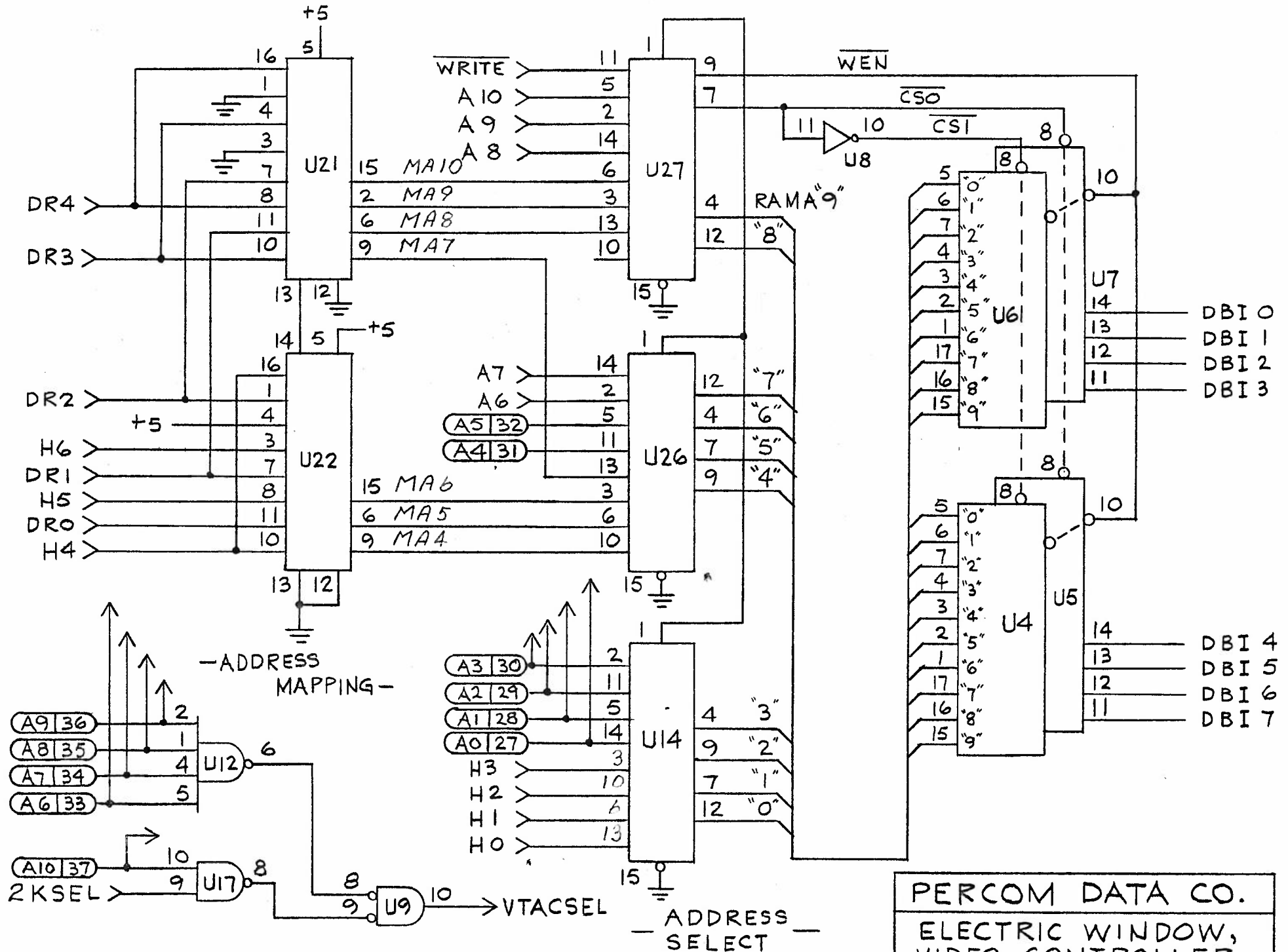


PERCOM DATA CO.
 ELECTRIC WINDOW,
 VIDEO CONTROLLER
 COPYRIGHT 1979 SHEET 1 OF 4



PERCOM DATA CO.
 ELECTRIC WINDOW,
 VIDEO CONTROLLER
 COPYRIGHT 1979 SHEET 2 OF 4

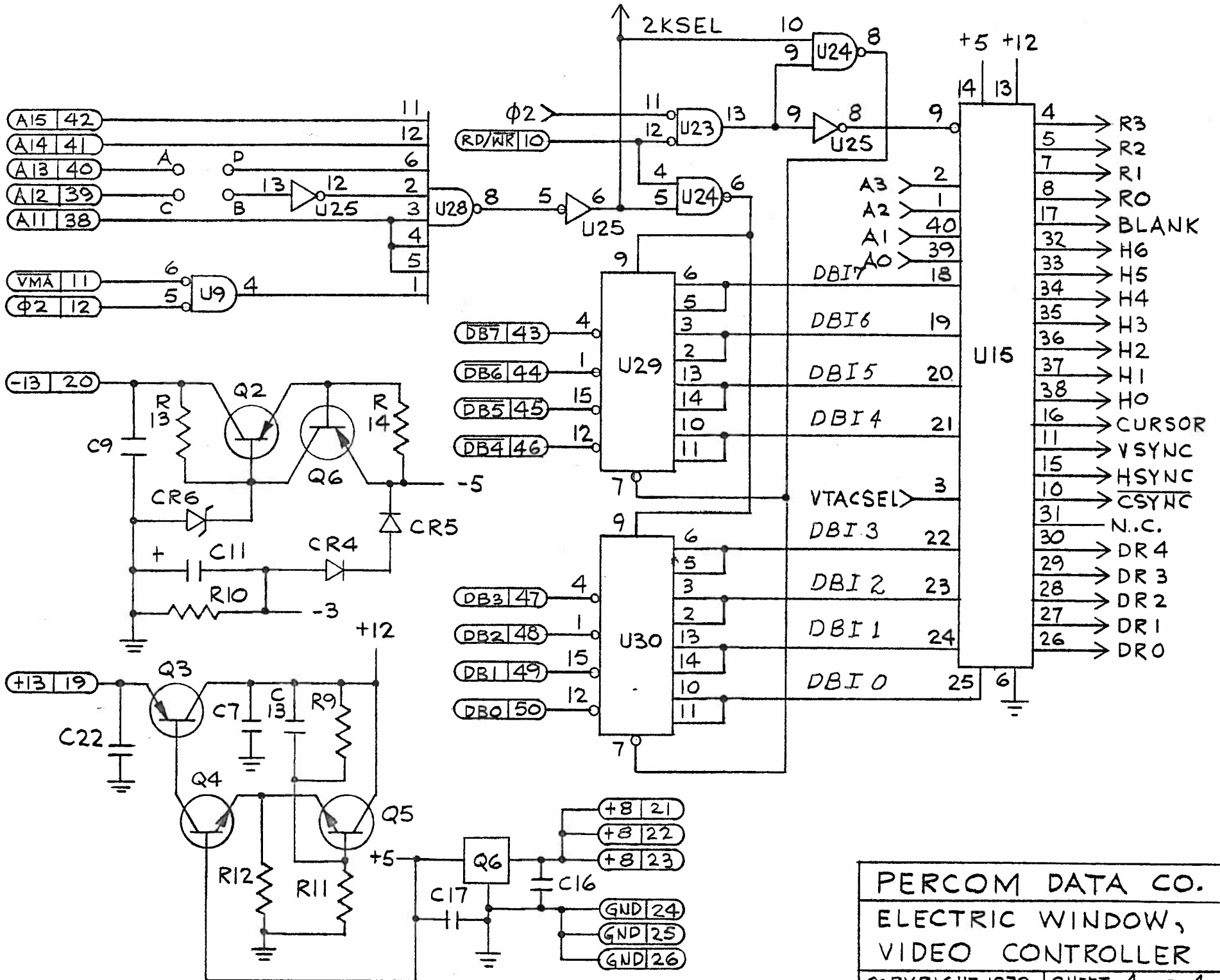




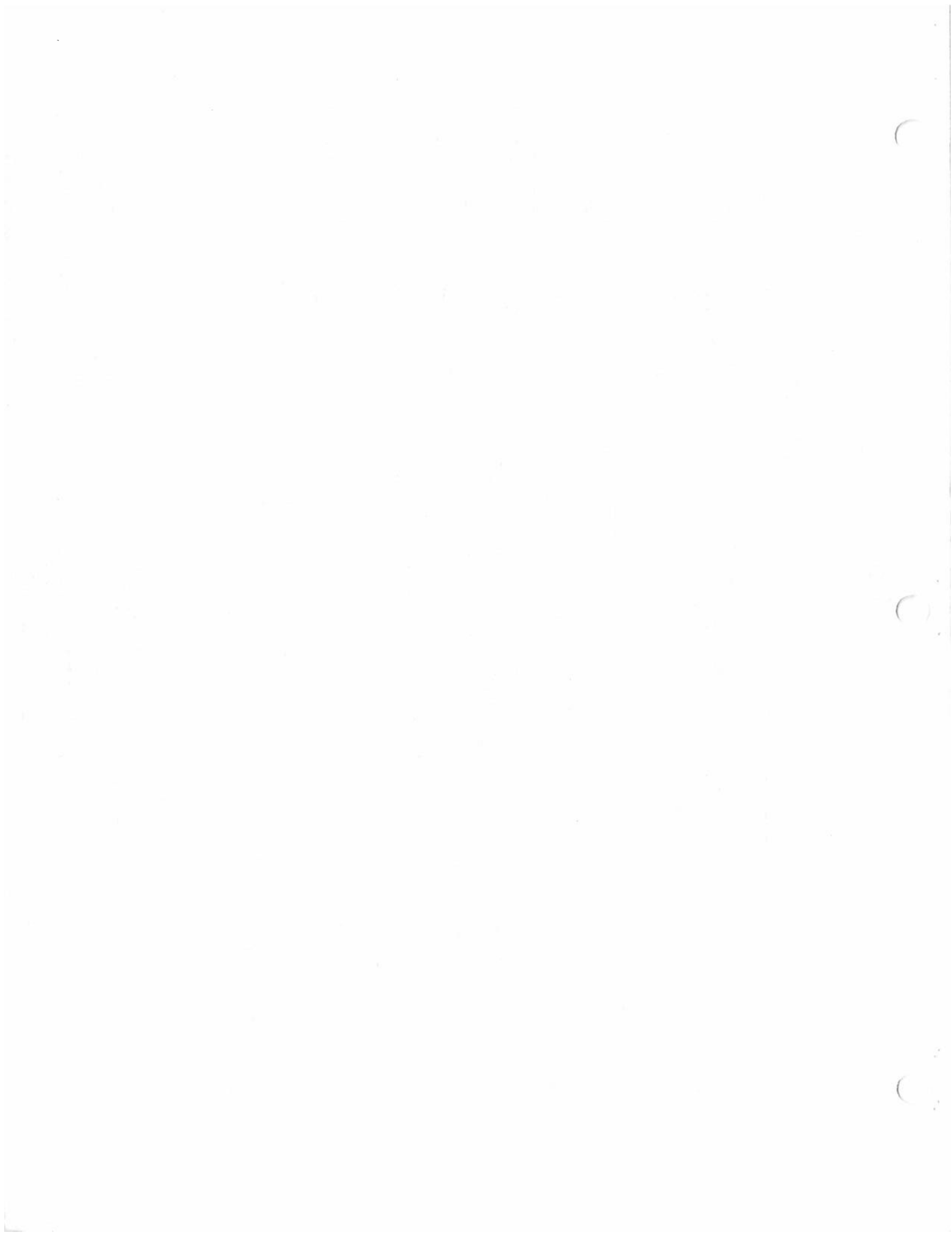
— ADDRESS DECODERS —

— ADDRESS SELECT —

PERCOM DATA CO.
 ELECTRIC WINDOW,
 VIDEO CONTROLLER
 COPYRIGHT 1979 SHEET 3 OF 4



PERCOM DATA CO.
 ELECTRIC WINDOW,
 VIDEO CONTROLLER
 COPYRIGHT 1979 | SHEET 4 OF 4



PERCOM