

OBJECTIVE DESIGN, INC.

PROGRAMMABLE CHARACTER GENERATOR

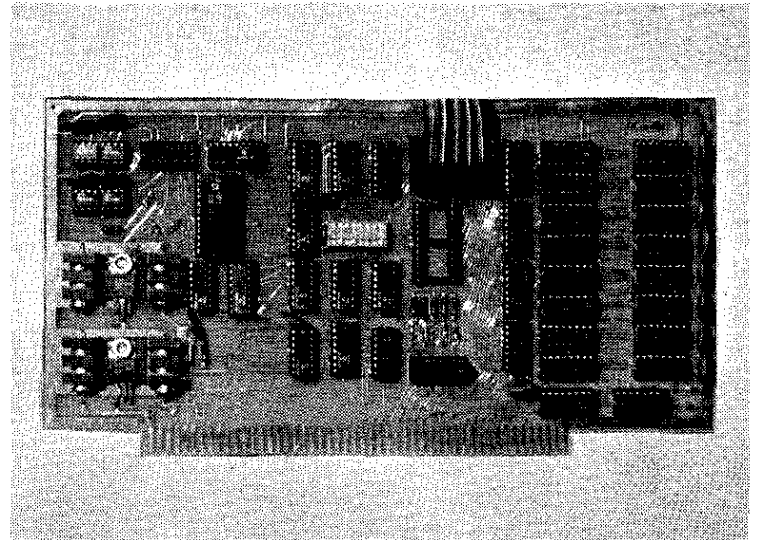
- Adds software created characters to video display
- S-100 format card
- Works with existing video devices using the Motorola family of 9x7 matrix character generator roms
- 2K of character memory is on the board. Memory is accessible via the buss, has software controlled protect and provision for battery backup.
- Full parallel keyboard interface is provided, with ready status input and -12, +5, GND available to the keyboard
- Two dimensional joystick interfaces are implemented. Requires only 100 K resistive inputs

The PCG is a unique S-100 card designed to increase the capabilities of existing video generating devices. When combined with the Processor Technology VDM-1 and SOL, Polymorphic systems VTI, Solid State Music Video Board or other devices which use the Motorola family of character generators, it adds the ability to dynamically create new character sets.

To join the PCG with a video device, one transfers the character generator ROM from the video board to a socket on the PCG. A cable from the PCG, which terminates in a 24 pin IC type connector, is used to join the circuits. This connector is placed in the socket formerly occupied by the character generator. Use of the PCG in a SOL terminal requires a longer connector. Please specify video unit in order.

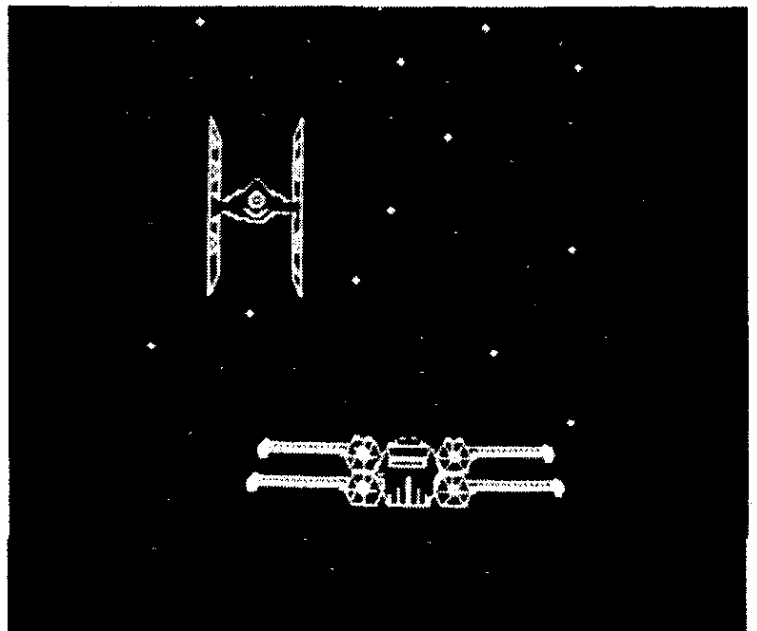
Under software control, or via an on-board switch setting, the data in the PCG RAM can be made to replace some or all of the character patterns normally supplied by ROM. Since the RAM is accessible for read and write from the buss, the displayed characters are under program control. The user can tailor a character set to the application. Some examples are:

- APL and other programming languages using non standard characters.
- Special mathematical and scientific symbols.
- Foreign alphabets and specially accented letters.
- *Italics and other forms of emphasis.*
- Special editing symbols for word processing.
- Music and speech notation.

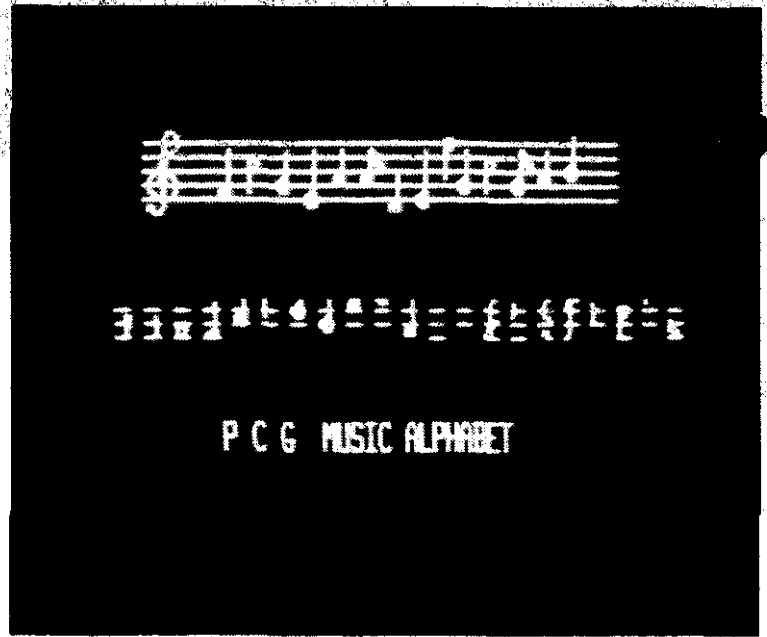
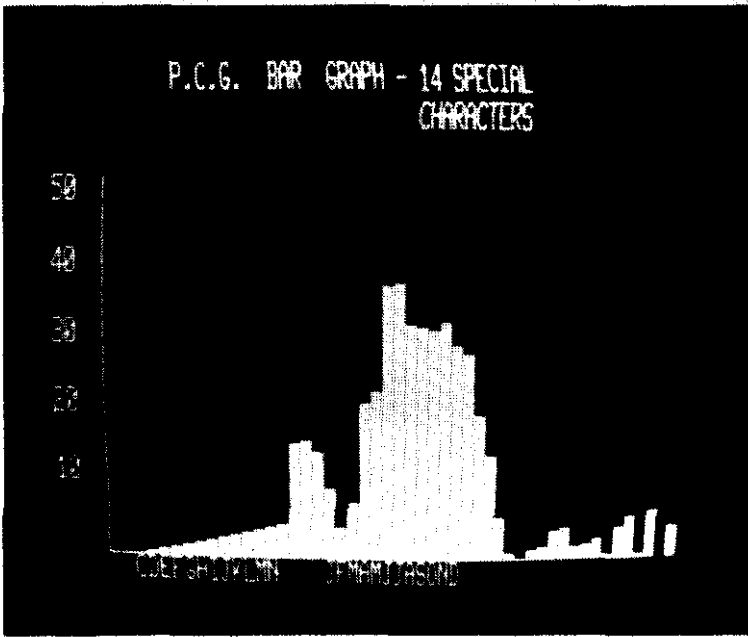


PCG Graphics

Programmable characters can be used to create high resolution graphic images up to 512 x 256. Unlike bit mapping devices, the PCG does not control screen points on an individual basis. For many graphic applications, it will prove unsuitable, while for others it is far superior — requiring the minimum amount of software overhead. The PCG will excel in graphics generation in those cases in which the screen can be broken into a basic set of images that are repeated any number of times. And take note: the PCG achieves its high resolution without the use of external system memory or any DMA activities.



P.O. BOX 20325 TALLAHASSEE, FL 32304
(904) 224-5545 ORDERS • 576-6002 ENGINEERING



Note on Graphics: some video boards inject blank spaces between characters which will affect the graphics, but not the special character images produced by the PCG. For the most part, this image breakup occurs in the horizontal direction. It can often be wholly or partially defeated by hardware adjustments.

PCG Hardware Specifications

Printed circuit board:
S-100 compatible, double-sided, plated-through, solder mask.

Power requirements:
+8 volts, -16 volts (optional, required only for keyboard using -12 volts)

Memory address:
2K, selectable on 2K boundaries.

I/O address:
8 of both in and out, on 8 I/O address boundaries.

Connector for keyboard/joysticks:
16 pin IC socket on board. (IC connector and cable are not supplied.)

Wait:
1 wait, always used. The standard version is intended for use with a 2mhz or slower 8080. The high speed version will allow operation with faster processors, to 4mhz cycle times.

Buss control signals used:
SINP(46), SOUT(45), PSYNC(76), POC(99), PDBIN(78), PWAIT(27), PRDY(72), PWR(77), MEMR(47).

Standby power:
On board pads are available for connection to external power source. Power required is 3-5 volts at 1/2 amp.

Memory protect:
Software controlled by I/O commands (not connected to the S-100 protect and unprotect lines). The power on condition (protected or not) is set by a jumper.

Joystick interface:
Four 555 timers operating on monostables, with external resistors (100K nominal) included in the timing chain.

SOL users:
Requires 36" connector. Please specify SOL with order.

Limited Warranty:

1. Kit or assembled unit may be returned within 14 days if the purchaser is not satisfied with this product for any reason, provided it has not been abused. This does apply to kits which have been assembled and used by the purchaser.
2. Kit: all parts guaranteed for 6 months. Repairs necessitated by incorrect assembly will be charged for parts and labor. In such cases, a binding estimate of repair cost will be sent before any repair work is begun.
3. Assembled and tested: six months guarantee. No charge for parts and labor.

Manual Only (will credit to later PCG purchase)	\$ 5.00
Complete Kit	\$149.95
Assembled/Tested	\$199.95
High Speed Version — add to above prices	\$ 16.00

SHIPPING: 5% U.S., 10% CANADA, 15% FOREIGN

OBJECTIVE DESIGN, INC.

**PROGRAMMABLE
CHARACTER
GENERATOR
(PCG)**

Programmable Character Generator

Introduction

The truly critical interface in any computer system is the one which joins man and machine. Usually this is the CRT screen to human eye connection. It is here that the computer will output information at the maximum rate. One means of improving this interface is to provide for maximum flexibility in the choice of characters displayed. The PCG provides circuitry for adding software generated characters to existing video display devices.

For many applications, the ability to create the characters which appear on the terminal screen can be considered a tremendous convenience, if not an absolute necessity. Computers serving as calculators in advanced mathematical applications require special symbols, Greek letters, sub- and super-scripts, and variable sized fonts. The best example is APL, an advanced programming language which requires many special symbols. Simpler computing programs, which transform the general purpose computer into the equivalent of a hand-held calculator, suffer from the lack of a specialized keyset and the corresponding display format. Financial and scientific applications call for their own special symbols. In fact, there are very few computer applications which cannot benefit from easy access to a specialized character set.

General Theory of Operation

The typical video display device (VDD) works with a raster scan CRT and produces screen characters a full line at a time. Each character is constructed on a matrix of screen dots, usually in a seven column by nine row configuration. The pattern for each character is stored, a row at a time, in ROM, i.e. a Character Generator Rom or CGR. Before scanning past any row of a character, the CGR is accessed using the row number and character type as an address. The latter is the now familiar ASCII code. The data which is produced by the CGR is shifted out serially to create the visible dots and spaces of one row of a character.

There is nothing unique about the CGR: it is merely memory. In fact, it is becoming common for manufacturers to use large PROMS in the place of the CGR in order to provide a programming service to customers. The size requirements of the CGR are easy to calculate. There are a maximum of sixteen eight bit rows which could be useful for each character. There are 128 characters in the full ASCII set. The result is a memory which has eleven address bits (four for rows and seven for characters) and is eight bits wide. Two 'K' (2048) bytes will hold all of the information in any CGR. In theory then, it is possible to substitute 2K bytes of RAM for the CGR.

Practice

The requirements of the replacement RAM are that it can be accessed by the main processor via the standard bus and by the dot generating

circuitry of the VDD. Retention of access to the CGR from the VDD is also a desired feature so that the terminal will be useable at system start-up or after a failure.

It is not difficult to gain access to address and data lines in the system -- one need only construct an S-100 card interfaced to the bus. The only tricky part is obtaining access to the same lines on the VDD. This is done by substituting a 24 pin connector from the memory board for the CGR on the VDD board. The CGR is then installed in a socket on the memory board. Through hardware and software switches, the user is able to choose between the CGR character set and a programmed set residing in RAM.

General Circuit Explanation

The circuit consists of 2K bytes of RAM which are addressed by the character and row lines that normally go to the character generator. These lines are multiplexed with the normal CPU address bus lines. The data out lines of the memory are available to the CPU bus and are multiplexed with the character generator data lines. This data set is sent via the 24 pin connector to the video board.

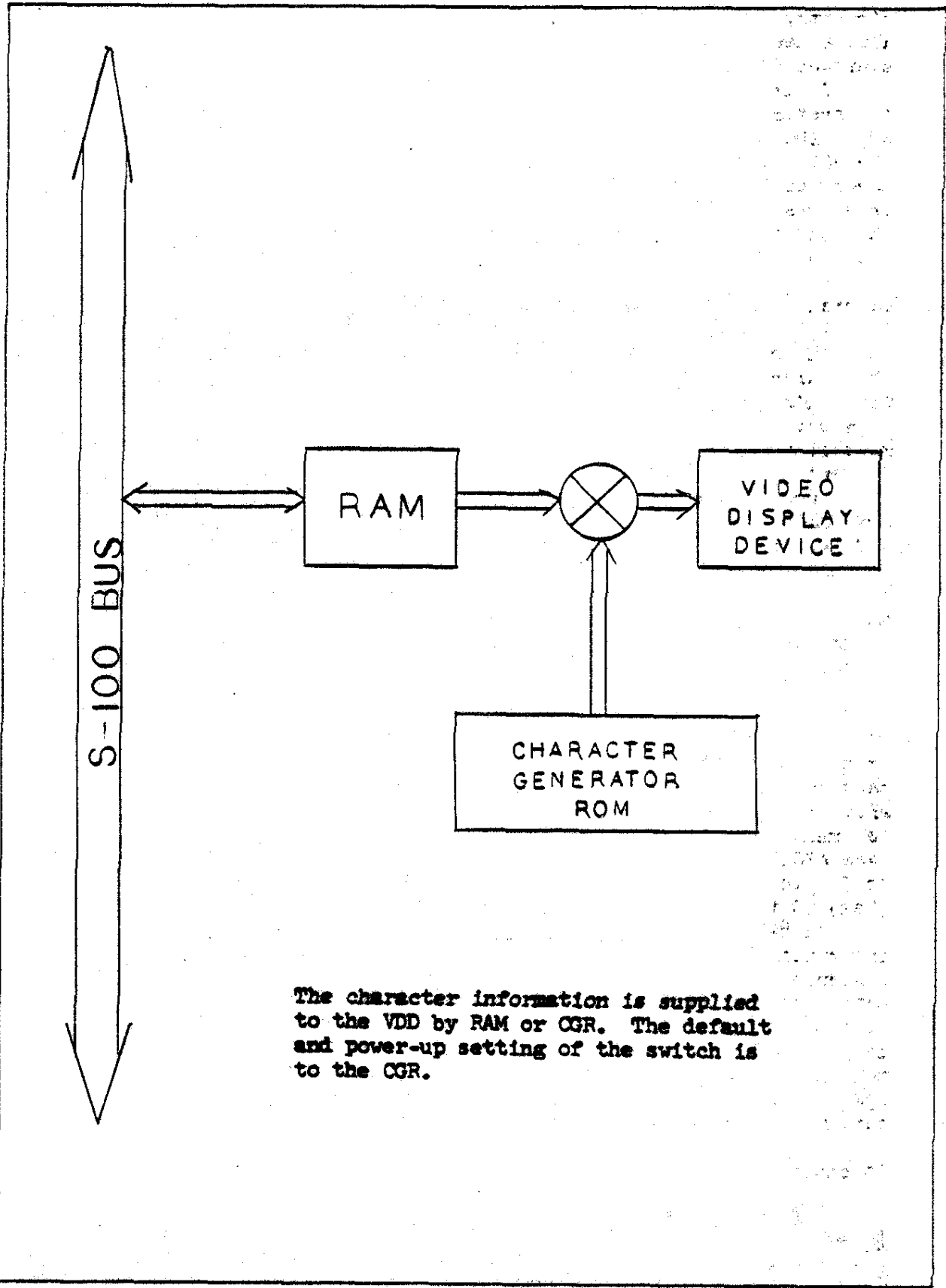
During normal operation, both the memory and the character generator are addressed by row and character selection lines. Hardware and software switches determine which data set is sent to the video board for display. If the CPU should require access to the memory for read or write, the CPU address lines are switched in to the memory. If, during a CPU memory access, one is also attempting to read a programmed character byte, the system causes the data switch to blank the output to the 24 pin connector.

The circuit has several modes of operation. It can be hardwired via an onboard switch to use only the character generator (normal mode) or only the memory (graphics mode). It can be hardware set to flip back and forth from graphics to normal mode by software commands. In another setting, the state of Bit 7 of the character select lines will determine the mode. The final operation setting selects the 64 upper case ASCII characters (letters, numbers, and symbols: 20 (hex) through to 5F) to come from the ROM and all others (control and lower case: (hex) 00 to 1F and 60 to 7F) to be taken from the RAM.

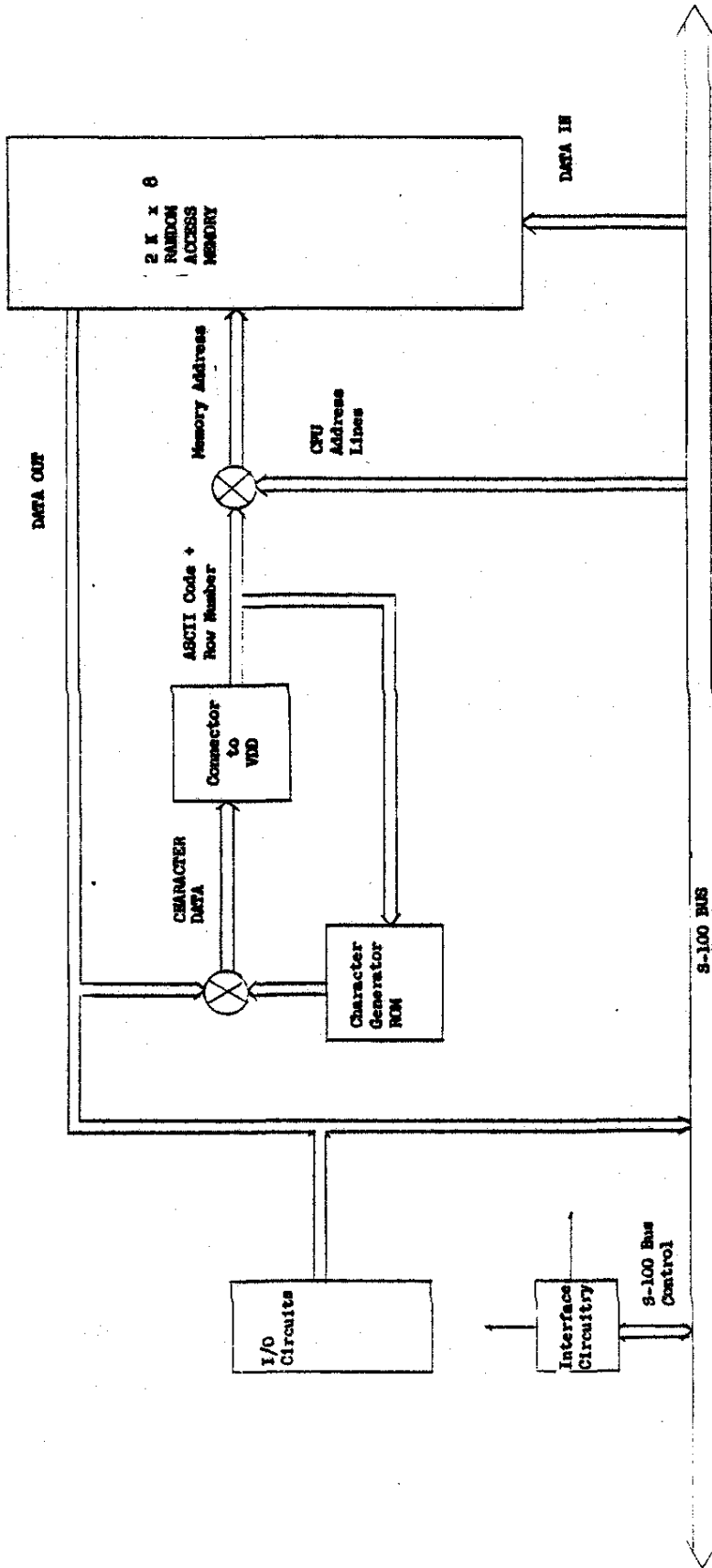
An 8212 on the board provides an 8 bit parallel input port for interfacing a keyboard. In addition, four monostable circuits provide a simple interface for two joysticks in two dimensions. Following an output command, the monostables all fire. The length of each pulse is determined by the associated external potentiometer. A software loop is required to monitor these and produce a value for the joystick position. The remainder of the circuitry is concerned with address decoding for the CPU bus. All components are standard and may be obtained from many distributors and wholesalers.

Compatibility

The PCG is intended to work with video display devices using the Motorola family (MCM 6574-6, and others) of 7x9 matrix character generators. With a few minor changes in the connector the circuit can be applied to display devices with other types of CGRs, e.g. the popular 2513 which works with a 5x7 matrix. However, the onboard socket



The character information is supplied to the VDD by RAM or CGR. The default and power-up setting of the switch is to the CGR.



PROGRAMMABLE CHARACTER GENERATOR -- BLOCK DIAGRAM

OBJECTIVE DESIGN, INC.

is designed to accept only the Motorola type. Using a changed connector to replace another CGR will mean that the ROM must be discarded. It will not be available at system start up, etc.

The PCG can be used with computer terminals whose circuitry is external to the computer. Four foot cable lengths have been tested with no sign of difficulty. (Contact Objective Design about special applications which involve longer cable lengths.)

The best results are obtained with the memory mapped VDDs such as the Processor Technology VDM-1, which reside in the computer. These devices also present the special advantage of having total control over screen addressing. This feature, in combination with programmable characters, can produce advanced graphics capabilities.

The Operation Modes

Each character to be portrayed on the CRT is represented by an eight bit binary code, labeled B0 to B7. B7 is not used for character selection, but rather serves a special hardware function -- to key in reverse video, for example. The remaining 7 bits are used to select from 128 possible characters. These 128 characters are shown in the ASCII chart. The PCG gives the ability to substitute, for some or all of these, characters created by the user. In each case, the seven bit code is used as an address in the character memory.

Circuitry on the PCG will select the character data according to information present in the code itself and the operation mode. There are five such modes: fixed normal, fixed graphics, command, programmed, and automatic. (Review PCG I for details on the selection process.) In fixed normal, the RAM is never accessed. For fixed graphics, command, and programmed modes there is a simple connection between the character code and the RAM locations used to describe the characters.

The RAM is divided into 128 blocks of 16 bytes each, proceeding from low memory to high. The 7 code bits, from 00 to 7F (hex), locate the block of 16 bytes used to describe the character. This is expressed by the simple formula:

Address in memory of first byte of character block =

Base address of RAM on PCG + [16 x code],

where the code is interpreted as a number.

Each byte will represent one row of the screen dots which form the character. Row 0 is the first byte (lowest address) in the block, and Row 15 is the last byte (highest address.) We are tempted at this point to say that Row 0 appears at the top of a character space on the screen and Row 15 at the bottom. But this is not always true, due to some hardware tricks in the different Video Display Devices. For instance, in the Processor Technology VDM-1, Row 15 is on the top, followed by Row 0 and the other Rows in order to 12. Since Row 15 is always blank when generated by the CGR, this isn't normally apparent. With PCG characters, it will have an effect and must be dealt with in the hardware or software. We will

low 4 bits of ASCII code

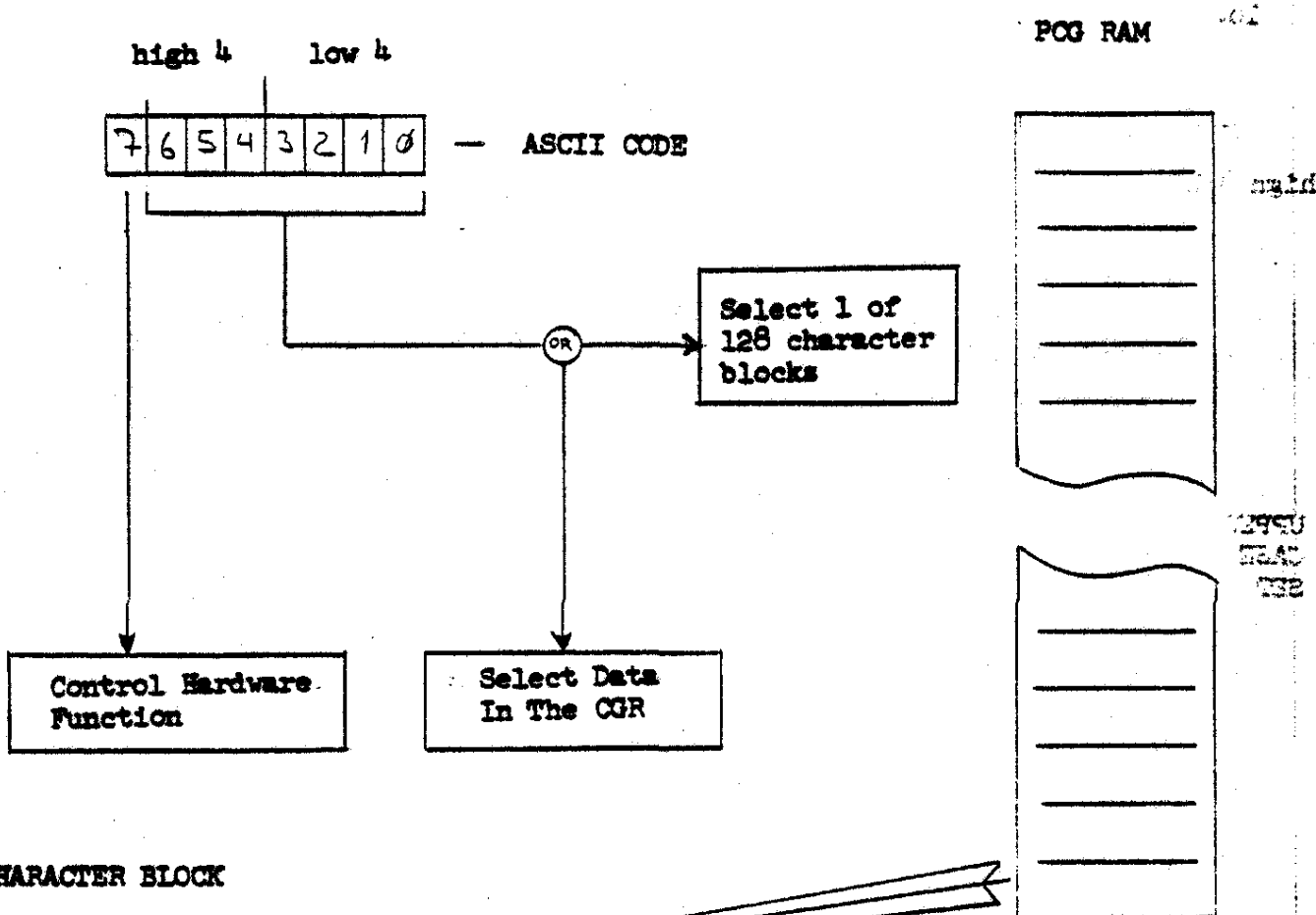
0 1 2 3 4 5 6 7 8 9 A B C D E F

high 4 bits	0	N	S	S	E	E	E	A	B	B	H	L	V	F	C	S	S
		U	H	X	X	T	Q	K	L	S	T	F	T	F	R	O	I
		CONTROL															
1		D	D	D	D	D	N	S	E	C	E	S	E	F	G	R	U
		L	1	2	3	4	K	Y	B	N	M	B	C	S	S	S	S
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
3		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
		P	Q	R	S	T	U	V	W	X	Y	Z	[/]	^	_
4		'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
		LOWER CASE															
5		p	q	r	s	t	u	v	w	x	y	z	{		}	~	/

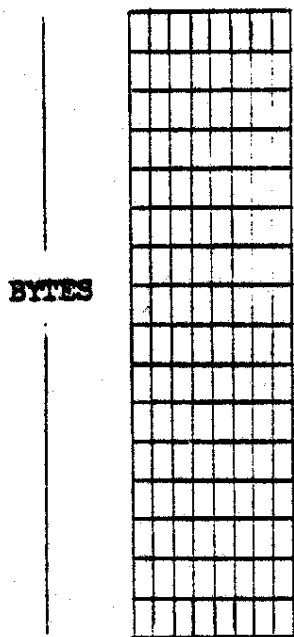
UPPER CASE SET

ASCII CHART

The seven bits of the Ascii code indicate which of the 128 characters in the chart will be generated. The data used for producing these characters is stored in the Character Generator ROM. The ROM functions in the same manner as ordinary computer memory.



CHARACTER BLOCK



*E
 DFGH LIXCVBQRWEYU IOP/*

The code bits are decoded to a value from 0 to 127. This number indicates which block of information, either in the CGR or POG RAM, will be used to generate a character. In the POG RAM, the data block consists of sixteen 8 bit bytes. The ones and zeroes in these bytes are converted to screen dots.

assume, for now, that the Rows are projected in numerical order.

The division of character RAM is slightly altered for the automatic mode. In automatic the 128 codes are split between the CGR and the RAM, with 64 characters being generated from each. The CGR will produce all of those characters in what is normally known as the 'Upper Case Set.' The remaining characters will come from RAM. It is unfortunate that the split does not take place in the center of the code numbers, as can be seen from the ASCII chart. So as to neatly separate the RAM into special character and general purpose groups, the codes normally located in the top 512 (1/2 K) locations can be moved to the second 512 spaces. In this case the low address 1K of memory is devoted to special characters and the upper 1K is free for system use. Automatic mode is probably the most convenient, since the normal character set used by most software is available, along with 64 special characters.

Creating the Characters

The 16 bytes which form a character are chosen in a very straightforward manner. The bytes are projected onto a simple grid, 16 X 8, where each bit equals one space. The spaces are roughly equivalent to dot spaces on the screen. The programmer constructs his character from filled spaces in the grid. When converted to data bytes, the filled spaces are ones and the empty spaces are zeroes.

In many cases, the circuitry in the VDD will allow only a subset of the 16 rows to appear. Also, the eighth bit in each row is often left unused -- although this can generally be 'corrected' by an alteration. It is not necessary to dig through VDD schematics to learn which rows and columns are projected. A study of CGR characters on the screen plus some trial and error work with the PCG will quickly reveal character limits.

The horizontal to vertical ratio of screen dot size will not be one to one in most cases. For this reason a grid composed of square spaces will not accurately portray the characters. The ratio of vertical to horizontal screen dot size can be calculated by:

$$V/H \text{ size} = V \text{ dots}/H \text{ dots} \times 4/3$$

where 4/3 is the standard screen aspect ratio and V and H dots are the number of dots across the screen vertically and horizontally. For example, if each character is drawn on a 12 X 8 grid with a video display of 16 rows by 64 characters per row, one has:

$$\begin{aligned} V/H \text{ size} &= (16 \times 12) / (64 \times 8) \times 4/3 \\ &= 192 / 512 \times 4/3 \\ &= 8/15 \\ &-- 1/2 \end{aligned}$$

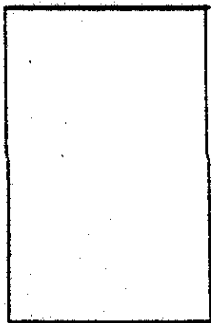
The grid should then be drawn with the vertical dimension of each space twice as large as the horizontal one.

Character Set Examples

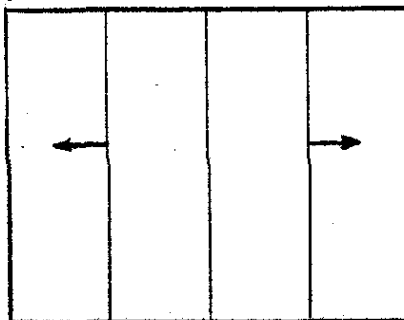
There has been a great deal of interest lately in APL for microcomputers. The two great stumbling blocks have been writing the interpreter and

DYNAMIC GENERATION

Filling
Single
Screen
Space

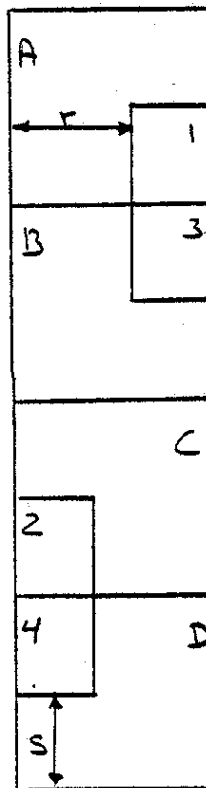
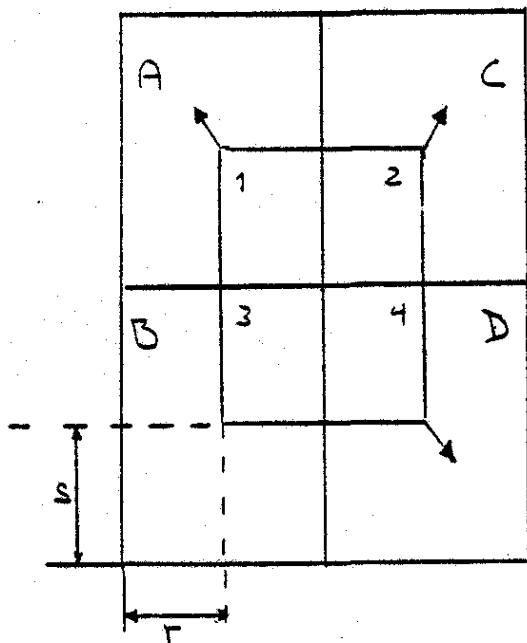


Overlapping
Two Spaces



Overlapping
Four Spaces

Four Character
Blocks In The
PCG Memory



A figure the size of a single character will overlap one, two, or four screen spaces. The characters can be dynamically created by mapping a 'master design' into a four byte block. The mapping function is determined by the relative location of the figure on the screen.

displaying the unique APL character set. With the PCG, producing the special character set is a trivial task.

Several of the characters are plotted here with a simple grid. The automatic mode is used for the entire alphabet, since the upper case character set is still needed. Placement in PCG RAM determines the keyboard code (and therefore, which keys) the characters will match with -- so that using 'shift' and 'ctrl' with the upper case set will cause special characters to be displayed. The user could easily add labels to a key set to produce a full APL terminal (minus all of the programming, of course!)

From this example, it should be obvious that any character set which will fit on the 9X7 (or larger, to 16X8) matrix used by the CGR can be generated. Also, multiple character sets can be stored in memory and swapped in as needed. In cases where more than 128 different characters are required -- but do not have to appear on the screen at one time -- it is possible to dynamically create them. In this instance a subset, possibly with only one element, of the set is swapped as required.

Graphics

The ability to create character sets implies some graphics capabilities. The simplest form of graphics will use an unusual character set -- but still handle it as an alphabet. A good example would be a set of musical notes. Each character will be a single note, projected on the musical staff. The software overhead to create such a picture is very small -- taking up less than half of the PCG memory. Using a high resolution point-by-point graphic display, the amount of time and software overhead required to generate such a picture would be enormous.

At the next level of graphics the hardware restrictions on the VDD begin to tighten. The ideal display is one in which there are no forced blank sections between the characters. This will enable us to build pictures from individual character elements. The next example is a timing diagram. In this case it is one for the 8080. The VDD used forces a single blank dot column between characters, which goes almost unnoticed.

The diagram uses very few characters, but repeats them many times. Those who are applications oriented should note that storing this picture, or any picture constructed from this character set, requires only 1K for the screen memory and less than 256 bytes for the characters. Even a limited mass storage system could hold all of the 8080 timing diagrams and information on the signals!

The bar graph picture is similar. However, its hardware restriction is mainly in the vertical direction. In this case it is best to eliminate the blanks between rows of characters as much as possible. Finally in the pictures we call 'pipes' and 'logic set' the restrictions are tight in both horizontal and vertical directions. It should be remembered that while the human eye is capable of noticing even the smallest flaw in a picture, the human brain will 'integrate' the available information and make up for most imperfections.

Dynamic Generation

The previous examples have utilized a simple character set with frequently repeated elements. Now let us consider a case where this is not sufficient. We have a figure - it could be a spaceship - must move smoothly across the screen. Assuming it is the same size as a single letter, it may overlap as many as four screen spaces at one time. Although the total number of different characters used to portray parts of the ship as it moves on the screen is quite large, only four are needed at one time. The total information on the screen appearance of the ship is contained in a single picture, requiring perhaps eight to twelve bytes.

The trick then is to dynamically create the 1, 2, or 4 characters needed at any one time from the basic ship picture bytes. This is done by mapping the image into the PCG RAM -- with the mapping function determined by the screen position. For a changing object which also moves across the screen, it is only necessary to maintain different sets of picture 'masters' from which to perform the mapping. If pictures larger than a single character space are required, they can be created by a repetition of the same process.

The applications of the PCG are as varied and numerous as the characters it creates. Once its use is mastered, there is virtually no character set or graphic which cannot be portrayed -- with very low overhead in cost, memory, and programming effort.

PCG Circuitry

Bus Interface

An 8131 comparator seeks for a match to the high 5 address lines to indicate a PCG memory or I/O action. The single 8131 can match different memory and I/O addresses by offering the compare circuit a choice of +, IO, IO not, or ground. Where the address line match for both memory and IO are the same, either 1 or 0, a + or ground connection is used. Where they are different, the IO line is chosen if IO is 1 and memory is 0. The IO not line is used if memory is 1 and IO is 0.

The 8131 latch is opened during SYNC. By the time SYNC ends, and the latch is closed, the match condition will have been determined. DECD is low on a match. If both INP and OUT are low, indicating that this is not an IO operation, the MEM signal will go high. This will cause the READY line to be pulled low, forcing a wait cycle from the processor. With the PWAIF signal, READY is returned high. All memory operations (and only memory) will have a single wait state. The MEM line is combined with DBIN and WR to produce 'memory read' and 'memory write' actions.

Memory

The PCG memory consists of 16 21L02 RAMs arranged as two 1Kx8 banks. The memory is accessed via 10 address lines (1K bytes) and an eleventh line which activates the Chip Enable of one bank or the other. Data input is taken from the S-100 bus, buffered by 3-state drivers which are always on. Data output is available to the bus via 3-state drivers, gated by MEM and DBIN. A memory protect flip flop provides a gating signal for memory write actions. A jumper determines whether the F.F. comes up in protect (jumper to ground) or unprotect mode with power up. The F.F. can be switched by INPUT commands to the PCG. (Data received on these commands will be OFFH.) The F.F. is not affected by the 'protect' and 'unprotect' bus lines.

The address lines to the memory are selected by 74LS157 multiplexers from the bus address lines (0 through 10) or the character type and row information which comes from the VDD. The multiplexers switch with the MEM signal.

Character Generation

The information which would normally go to the CGR on the VDD (7 character select lines and 4 character row lines) is brought to the PCG via a cable and 24 pin connector which is placed in the CGR socket. The CGR is removed from the VDD and placed on the PCG board. The 11 lines are sent to the PCG memory by the address multiplexers. These lines are also presented to the CGR just as they are in a VDD. The output from the memory and the CGR are both input to a multiplexer. The output of which is returned to the VDD via the cable and socket. When the CGR is selected, the results are the same as they would have been if there were no PCG involved. When the memory supplied data is selected, the

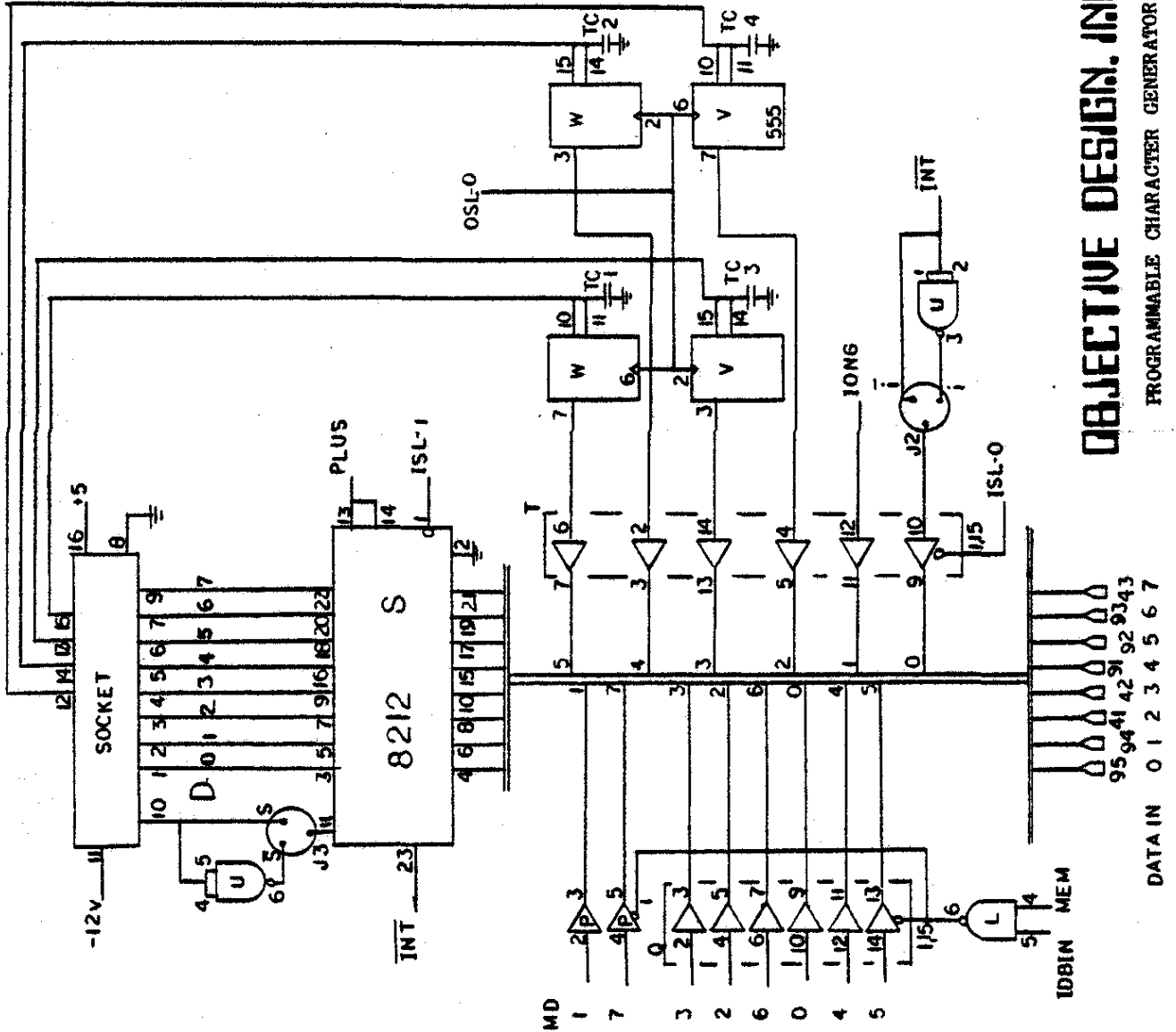
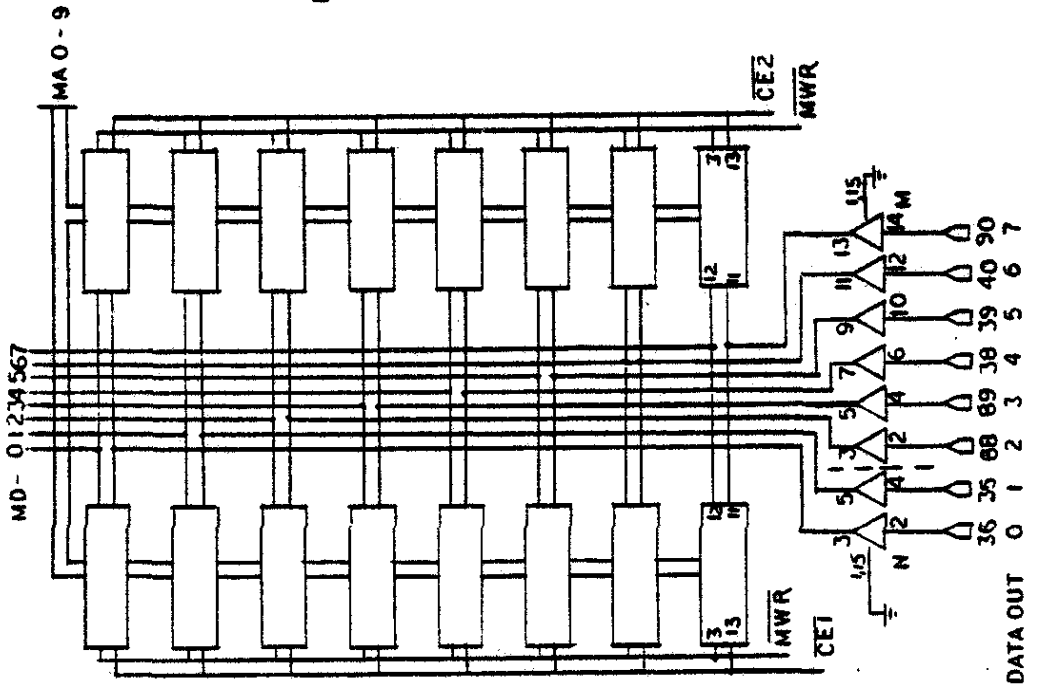
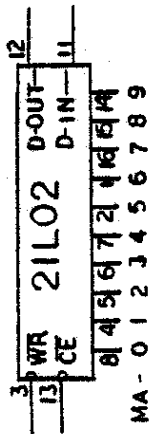
VDD will produce characters whose design has been programmed.

Selection Process

There are four data select modes available on the PCG: fixed, programmable, command, and automatic. In the fixed mode, the data select lines for the multiplexers are set high for normal characters and low for programmed characters. In programmed mode, the high order bit from the VDD, usually used for reverse video internally, is brought out via pin 14 on the connector and controls the data select process. In command mode, the output of a F.F. determines the selection process. The F.F. can be switched by output commands. OSL-1 brings programmed characters; OSL-2 switches to normal characters. In automatic mode, the multiplexers switch to the CGR for all of the upper case set and to the RAM for the control and lower case characters. An Exclusive Or operation on address lines (code lines) 5 and 6 creates the switching information.

PCG Switch Settings

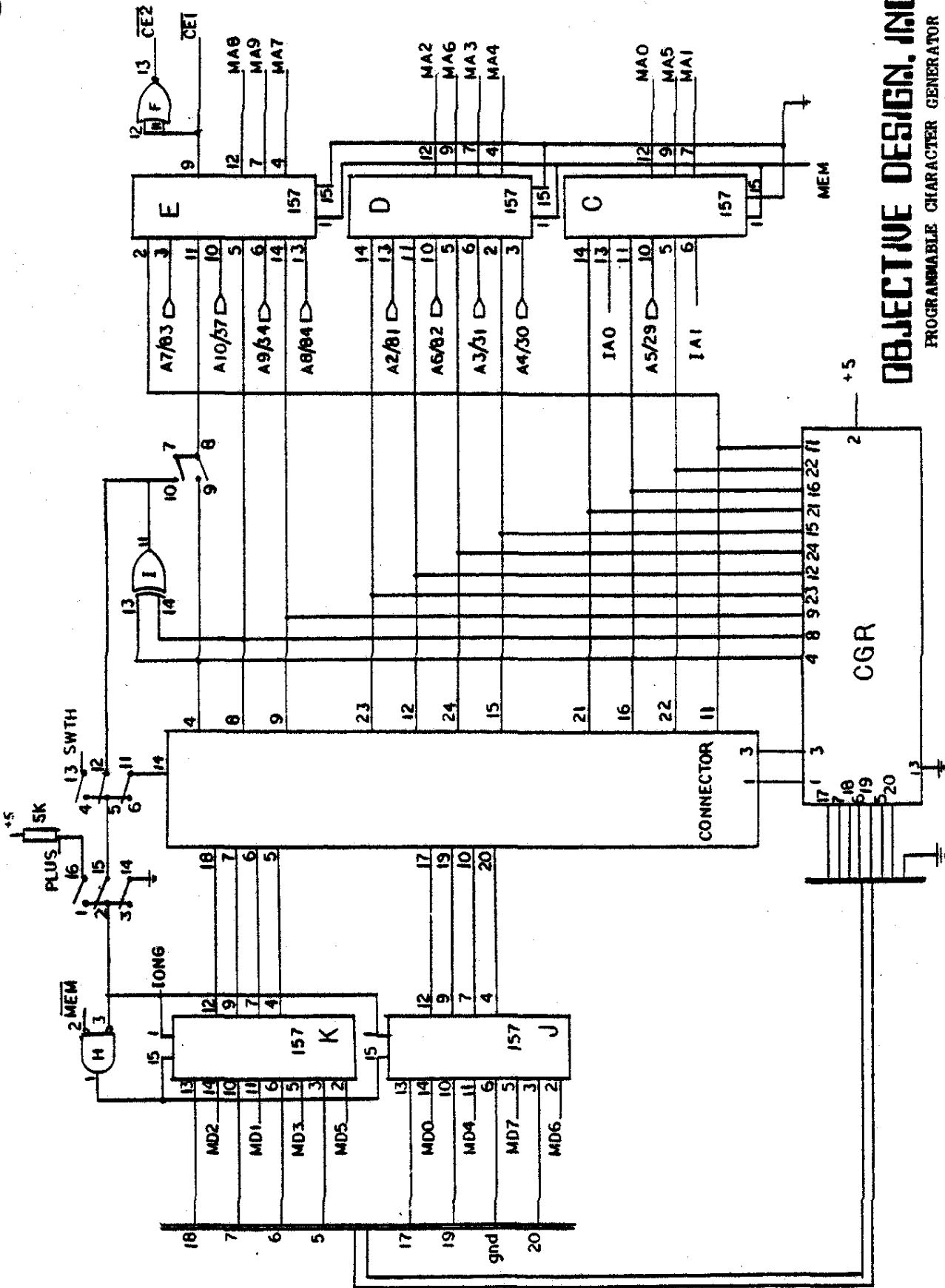
Mode	Switches Closed (On)	Switches Open (Off)
✓ Fixed Normal	1,8	2,3,4,5,6,7
Fixed Special	3,8	1,2,4,5,6,7
Programmable	2,6,8	1,3,4,5,7
Command	2,4,8	1,3,5,6,7
Automatic	2,5,7	1,3,4,6,8



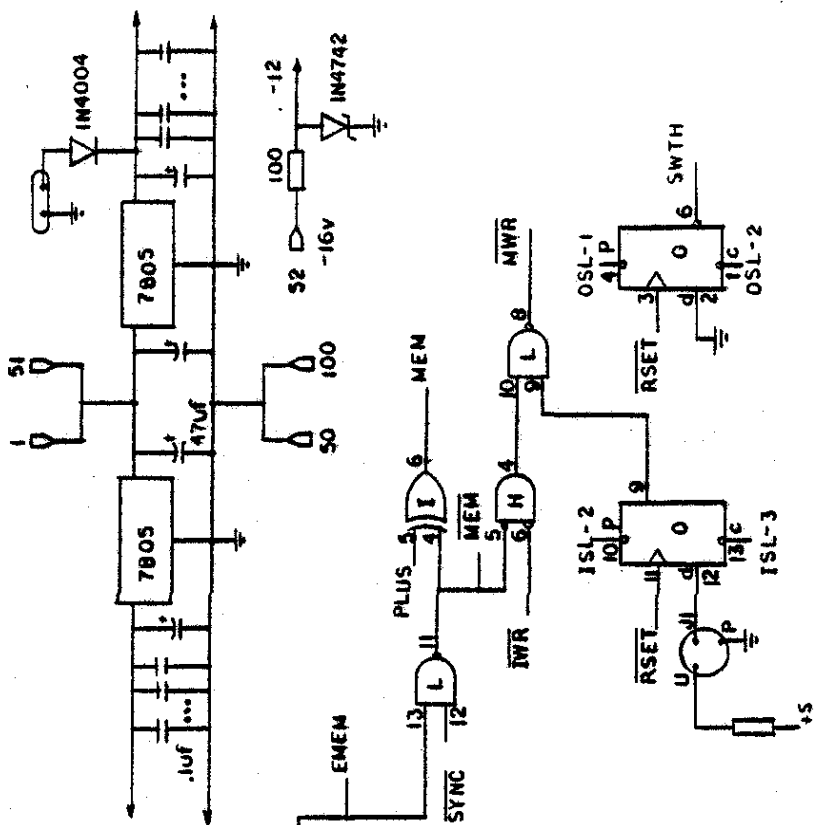
OBJECTIVE DESIGN, INC.
PROGRAMMABLE CHARACTER GENERATOR

DATA IN 0 1 2 3 4 5 6 7
95 94 41 42 91 92 93 43

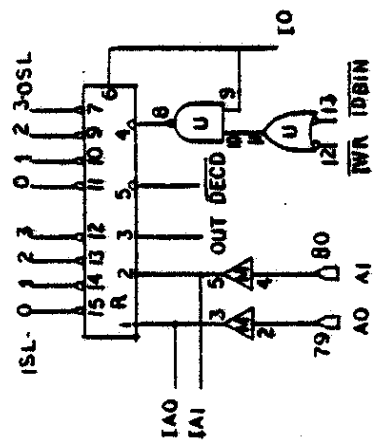
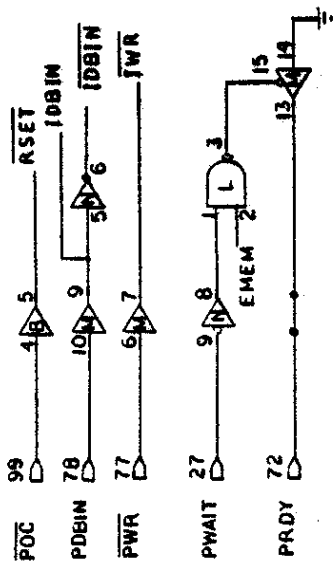
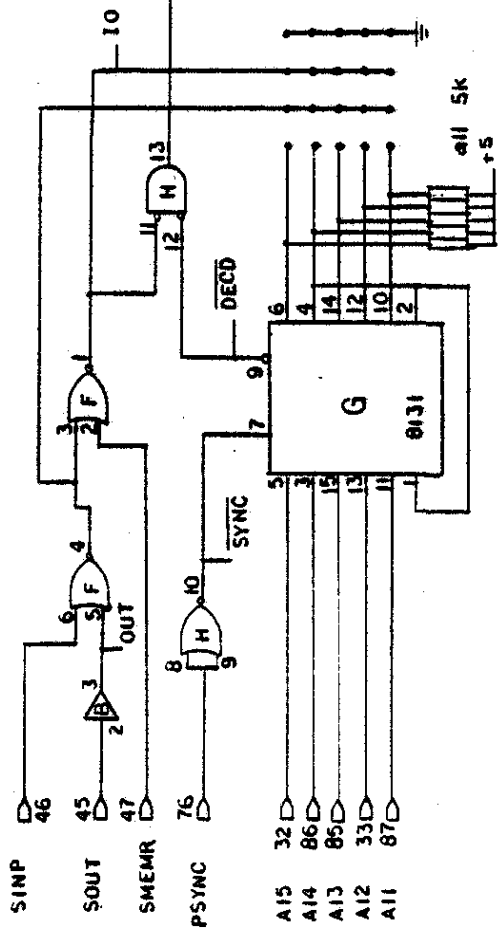
DATA OUT 0 1 2 3 4 5 6 7



OBJECTIVE DESIGN, INC.
PROGRAMMABLE CHARACTER GENERATOR



OBJECTIVE DESIGN, INC.
PROGRAMMABLE CHARACTER GENERATOR



M1, B1, B15 → 7

Assembly

Those who are experienced in the construction of S-100 or similar printed circuit cards will consider the assembly of the PCG to be a 'piece of cake.' It is. However, if you have never done soldering of small electronic components, then please don't start with our board. As usual, if all else fails: Read The Instructions!

A)

✓ Spend at least five minutes carefully studying the board. Look for the very small breaks that can easily appear in PCB lines. Make any necessary correcting connections. Also, use a fine knife blade (X-ACTO or similar) to insure that any lines which inadvertently touch pads or other lines are clear. Test these lines with an ohm-meter. Place the board in the computer and check the alignment and fit of the connector.

B)

✓ Locate the COMPONENT side of the board. This is the side with 'PCG' and 'OBJECTIVE DESIGN, INC.' printed on it. Orient the board so that the component side is up and the connector fingers are nearest to you. The top of the board is the edge opposite the connector.

C)

✓ Now the sockets... There are three sizes: 24 pin, 16 pin, and 14 pin. Be careful to avoid placing a 14 pin socket in a 16 pin position. Take note: do not use a socket in the 16 pin position labeled 'switch'. This is where the dip switch will go. Take your time here and work carefully.

D)

✓ The eight position DIP switch is next. Orient the switch so the writing on it is readable (not upside down) when the board is in the 'connector finger toward you' position. Solder it in place.

E)

✓ The seven 1/2k resistors can now be placed on the board. Solder and trim the leads. Be sure to hold each lead with one hand while cutting. Eyes tend to come in matched pairs, so replacements are difficult to obtain. Be careful.

F)

✓ Locate the 100 ohm resistor and zener diode positions on the board. They are near the regulators, at bottom left. Solder the parts in place. The black band of the diode (1N4742) is 'pointed' toward the bottom of the board.

G)

✓ The backup power supply diode goes on now. It is located at the top right-hand corner. The black band of the diode (1N4004) goes on the right.

H)

The filter capacitors are next. These 12 .1 uF ceramic capacitors have a plastic coating on their leads which must be scraped off with a knife edge before soldering. Run the blade edge along the leads a few times.

I)

Now solder on the monostable capacitors, which are the same as the filter caps. The same cleaning process is required. The timing caps are all in the top left-hand corner of the board.

J)

The four 47uF electrolytic capacitors go in the regulator area at the bottom left. All have the plus lead pointed down.

K)

The regulators and their heat sinks will now be placed on the board. Put the regulators in position with their mounting holes over the large holes in the board and their leads pointed down. Note where their leads will have to be bent to match the PCB holes. Bend the leads and test them. Make adjustments as necessary. It is a good idea, but not essential, to place some thermal compound on the heat sinks where they will contact the PCB and regulators. Use only a thin film and avoid a mess. Fit the regulator and heat sink assemblies onto the board and bolt them down with the #6 hardware. The washer is first placed on the screw which is then inserted from the back of the board. Solder the leads in place.

L)

Jumpers... Refer to the special section on jumper selection. If you believe that some of the jumper selected features will require constant change, then some form of connector is a reasonable idea. We recommend using a small pin from a multi-pin connector. Also, there are some small single pin sockets designed to be soldered or pressed into a PCB which will do a fine job. Molex pins will not do a good job.

M)

Before placing the integrated circuits in their sockets, test the resistance between the 5 volt test points (near the regulators) and ground. At the upper t.p., resistance should measure from 200 to 1000 ohms. A very low reading implies a short. This will most likely be due to a solder splash or similar problem. The other t.p. should measure approximately three thousand (3000) ohms. Again, a very low reading implies a solder related problem. In both cases, a bad component is possible.

O)

Carefully place the board into a slot on an S-100 computer. Turn on the power and check the voltages at the test points. Both should be within a range of 4.8 to 5.2 volts. Any other reading (or smoke) implies a problem. Check the voltage at the high end of the zener diode. It should be near -12 volts. A very low, but not zero, reading means that the diode is in backwards!

N)

Now place all of the chips on the board. Be especially careful when handling the memories, as they are sensitive to static charges. For

the most part, this means taking the memories from their container and immediately placing them in their sockets. Note that the 555s are placed two to a socket.

P)

Resistance checks should be made again. The high and low test points should measure about 200 and 1500 ohms, respectively.

CONSTRUCTION IS COMPLETE. Initiate testing procedures.

PCB PARTS LIST

Printed Circuit Card and Manual

.....Integrated Circuits.....

Type	Quantity	Circuit I.D.	Function
74LS00	2	L,U	Nand gates
74LS02	2	F,H	Nor gates
74LS04	1	N	Inverters
74LS74	1	O	Flip flop
74LS86	1	I	Exclusive-Or
74LS138	1	R	3 to 8 decoder
74LS157	5	C,D,E,K,J	Multiplexer
74367	6	A,B,M,P,Q,T	Hex 3-state drive
DM8131(NAT)	1	G	6 bit comparator
8212	1	S	3-state 8 bit latch
555	4	V,W	Timer
21L02	16	mO-15	1Kx1 static memory
7805	2	--	5 volt regulator

.....Sockets.....

Pins	Quantity
14	7
16	32
24	3

.....Resistors.....(all 1/4 watt, 5% tol.).....

Value	Quantity	Description
1000 ohms	7	Brown/Black/Red
100 ohms	1	Brown/Black/Brown

.....Capacitors.....

Value	Type	Quantity	Description
.1uF	ceramic disk	12	small disk, for noise filtering
.1uF	ceramic disk	4	small disk, for timing

47uF electrolytic 4 axial, used in power supply

.....Diodes.....

Number	Type	Quantity	Use
1N4742	Zener/12 volt	1	for -12 volt supply to keyboard
1N4004	Diode/1 amp	1	to switch in backup supply

.....Miscellaneous.....

Description	Quantity
8 position DIP switch	1
Heat sinks (for regulators)	2
#6 machine screws	2
#6 nuts	2
#6 washers	2

.....Plus 1 24 pin to 24 pin, 12 inch connector.....

Note: Longer connectors, to 48", may be purchased from Objective Design or substituted (at no additional cost) for the 12" version normally supplied. Cable lengths available are: 12,18,24,36, and 48 inches -- all costing \$6.00 each.

PCJ Joystick Interface Timing

This interface is intended primarily for cursor (or other character) control on the screen or as a simple directional input. If the processor's time is not valuable and the application does not call for response time below 100 milliseconds, then the potentiometer setting to digital value conversion can be very accurate. The interpretation software and the component values used (which set the timing) determine the speed and accuracy of the interface.

The timing equation for the monostable is:

$$T = 1.1 R C$$

where R is the external resistance in ohms (potentiometer on a joystick in our case) and C is the value, in Farads, of the onboard capacitor. Resistor values may vary from 10K to 10M ohms. The capacitor may range from .001 to 10 uF. With the supplied .1 uF capacitor and a 100K potentiometer, the maximum pulse time is 11 milliseconds.

It may be that for some real-time applications this will cause the processor's attention to be diverted for an unacceptably long time.

Substitution of .01 uF capacitors will give a pulse time of 1.1 milliseconds or less. However, the user must not forget that such action will reduce the resolution of the device, since the software loop will not be able to make as many counts during the shorter pulse.

PCG TESTING

There are three phases to the user checkout of the PCG: Memory test, External Interface test, and Character Generation. Objective Design recommends that even those persons receiving a factory assembled board go through these tests -- both as a check on our work and for device familiarization.

Memory Test

Before attempting connection with the Video Display Device in a system, the PCG should be tested as though it were a 2K memory board. A short test which the user can adapt to his system is included. It is a good idea to simply enter and dump a few bytes into memory before the test takes place. This will insure that the address setting is correct and that Memory Protect is off.

In the event that the test cannot be successfully completed, here are some possible causes:

- Total inability to read or write---incorrect addressing or failure of address circuitry, memory protect on.
- Large number of randomly located failures---insufficient access time due to problem with the WAIT/READY circuits, crossed data or address lines.
- Repeating failures at address increments---crossed address lines. Note: the mem test given is inappropriate for locating this problem. Simple ENVR and DUMP operations should be used.
- Constant fail on one bit or at one address---bad memory chip.

External Interface Test

The joystick and keyboard interfaces are best checked by attempting to use them. Be alert to problems caused by incorrect connection to the board or mismatching of the hardware jumpered conditions to the software. As with the memory, total failure can be due to incorrect addressing. The 8212 used in the keyboard interface is sensitive to low voltage conditions. If it drops bits on an irregular basis, check the Motherboard voltage (should be greater than or equal to 8 volts) and the onboard voltage (+5 volts).

For the joysticks it is best to write a program which will interpret the external resistance values of all four inputs and display them on the system terminal.

Character Generation

As with the interfaces, this feature is best tested by attempting to use it. The connection between the PCG and VDD is made by removing

the character generator rom from the VDD and placing it in the CGR socket on the PCG. Place one end of connector in its socket on the PCG, and be sure to orient the flattened corner to the left. Place the other end in the socket formerly occupied by the ROM on the VDD. Be certain to align it so that the flattened corner is in the Pin # 1 position on the VDD. (The orientation on the board will vary with the VDD used.)

Set the PCG switches to Fixed Normal and place both boards in the computer. Perform the usual start-up; the video board and display should work as though there were no PCG. If not, you have a problem. It is time to recheck all of the settings.

Once you are sure the video board will work as before, change the PCG settings to Automatic Mode. Enter a series of random bytes into the very beginning of the PCG memory (not all zero.) Now enter several bytes of 00 into your video memory. These entries can be made via a front panel or by operating system commands.

At the screen positions that correspond to these video memory locations, the randomly created character should now appear. Every entry into the first bytes of the PCG ram should show up in these screen positions. Entries into the PCG ram at Base Address + 0210H will appear on the screen wherever the letter 'a' is placed. For the letter 'b', the graphic block begins at Base + 0220H, etc.

The user can proceed to check out the other modes of operation, using the appearance of randomly created characters as an indicator for the switch from CGR to PCG RAM. Running the memory test when in a graphic displaying mode will produce dynamically changing characters.

It is unfortunate that the various video devices are sufficiently different to make a universal checkout procedure impossible. Each display will have a different character matrix and will handle the character generation in its own way. All the devices should be capable of displaying at least the 7 low bits of each graphic byte and the first 12 bytes in a graphic block.

In the event of a failure in a factory assembled board, notify Objective Design for immediate return and correction. Purchasers of the kit or printed circuit board should examine the board to the limits of their hardware and software experience. Repair work is not charged if the product is within the warranty period and the problem is not due to faulty assembly.

The PCG IO Ports

- IN-0 Status
- IN-1 Data for keyboard input
- IN-2 Set UNPROTECT MODE (data all ones)
- IN-3 Set PROTECT MODE (data all ones)

- OUT-0 Trigger all four monostables
- OUT-1 Set to GRAPHICS (if in command mode)
- OUT-2 Set to NORMAL (if in command mode)

Note: the given IO port numbers must be added to the IO base address set by J4. Since this address has only 5 significant bits, the PCG will actually be taking up 8 input and 8 output ports.

Example: the 5 J4 bits are, from high to low, 11000. This implies that IN 0 for the PCG, which will read the status, is given by the command: IN 11000000B, or IN COH. OUT COH will trigger the monostables.

As you may have already noticed, if we use the base address of our example, then IN 11000100B (IN C4H) will also read the status port.

Status Port ISL-0

- Bit 0 Data Ready for keyboard input port. Selection of Data Ready on 1 or 0 is made with J3. The Data Ready flip flop is internal to the 8212. It is automatically reset when the Data Port (ISL-1) is read.
- Bit 1 Normal or Graphics Status. A 1 indicates normal characters (i.e. the Character Generator Rom is selected.) A 0 indicates graphic characters.
- Bit 2-5 A 1 indicates that the monostable pulse is active and timing should continue. Each bit is related to a different monostable, which in turn is tied to a particular external resistor.

Status Bit	Socket Pin
2	12
3	13
4	14
5	15

Bit 6-7 Will always read as high (bus floating.)

Altering VDDs For Better Operation

The various Video Display Devices all have their own unique method of putting characters on the screen. For the most part, there is little that can be done to produce changes in them which will affect the PCG performance.

The CGR will only output 7 bits of information for each row of a

character. These 7 bits are input to a parallel loading shift register -- which is almost always 8 or more bits wide. It is the eighth bit which interests us, since the PCG will generate 8 programmed bits of information. The shift register's eighth input will be tied to ground in most cases. To make the change it is necessary to break this connection and tie the pin to pin 10 on the CGR socket. There are two important cautions here. First, the ground connection to the shift register may not be easily accessible on a VDD which is already assembled. (It may be on the component side under a socket.) Also, it may be tied to surrounding pins which are also to be grounded. You can get around this problem with the none too neat solution of lifting the input pin and soldering a small wire directly to it. The second problem involves the use of multiplexers between the CGR and the shift register. (As in the Polymorphics Systems VDD.) In this case it is a pin on the multiplexer which must be changed from ground to pin 10 on the CGR socket. (On the 'Poly' it is pin 3 of IC 32, a 74LS157.)

X In the typical VDD, there is the option to use Bit 7 to turn on a type of 'block graphics' or to cause the character to be displayed in reverse video. The user may route this line to pin 14 on the CGR socket for use on the PCG. When the PCG switch is set to 'Programmable Mode,' Bit 7 will cause switching between normal and graphics characters. (A 1 is normal, a 0 is graphics.) Users making this change should take care to see that pin 14 on the CGR socket is not connected to any part of the circuit on the VDD. Normally it is a free pin. The reverse video and block graphics features are nice to have. Unless there is a very definite need to use the programmed mode of the PCG, we suggest this change be avoided.

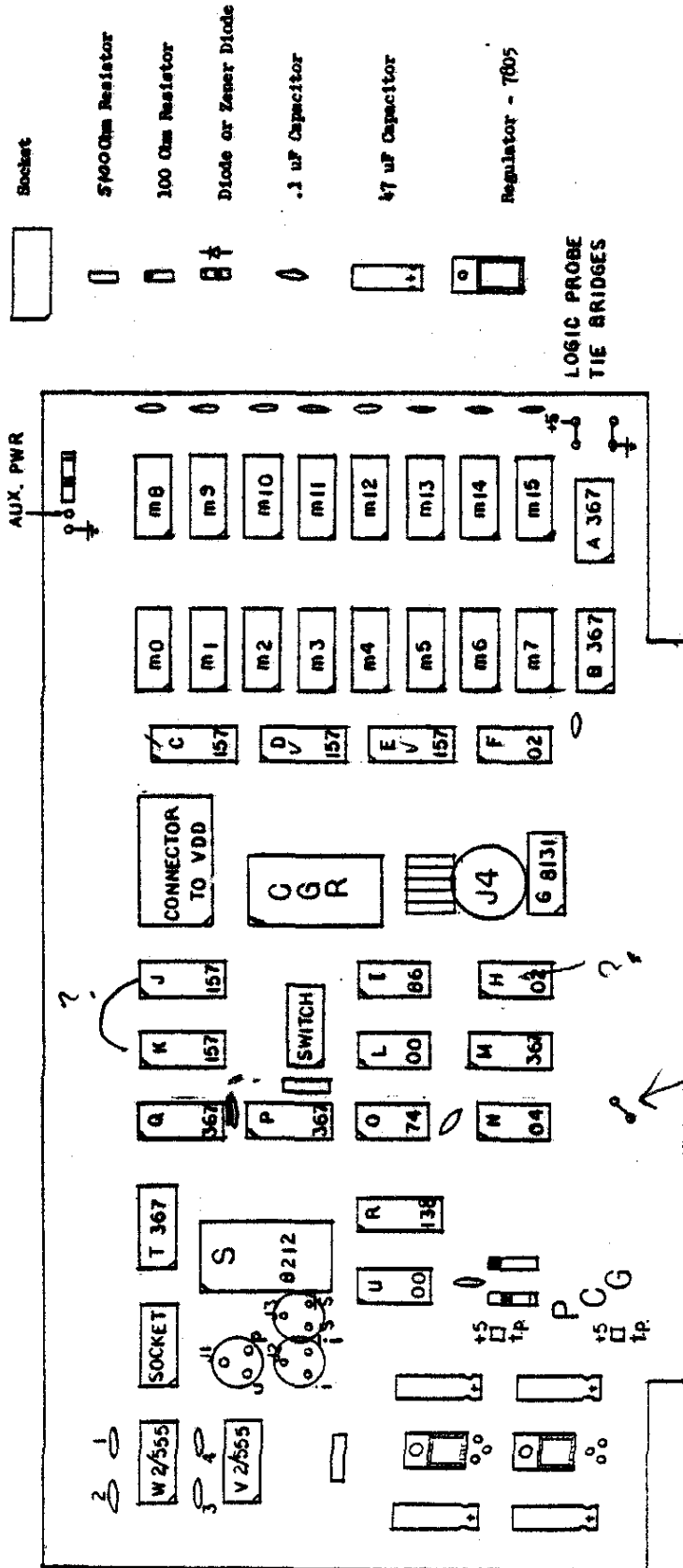
Getting the 8th dot on popular video boards

✓ VDM - Remove gnd from IC3, Pin 14. Connect this pin to IC 4, Pin 10. Note: when the PCG isn't connected, screen will have vertical lines. Do not remove gnd from pins 1,6 of IC 3. VDM uses 9 wide matrix, even after fix one dot remains blank.

Polymorphic - Remove gnd from Pin 3, IC 33 (32 on old units) and connect to IC 37(36), the CGR socket.

SSM - Remove gnd from V7, Pin 13 and connect to V16, Pin 10. The CGR. Note: on Poly and SSM connection is from CGR to 74LS157.

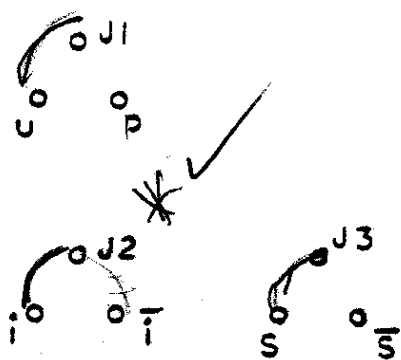
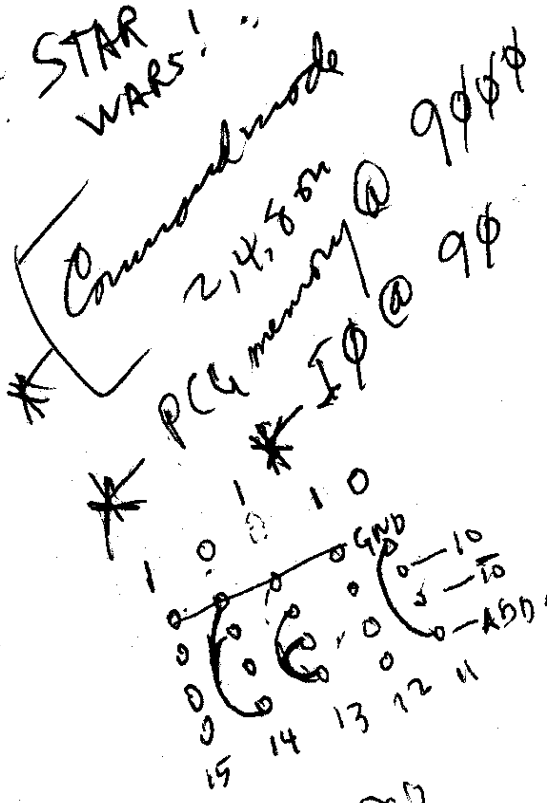
Attention: The video board from ObjDesInc is designed to work with the PCG. Characters are contiguous (no space between) and no board modifications are required. This video board is compatible with existing 64x16 boards, but will also do 80x24 displays as well as several others.



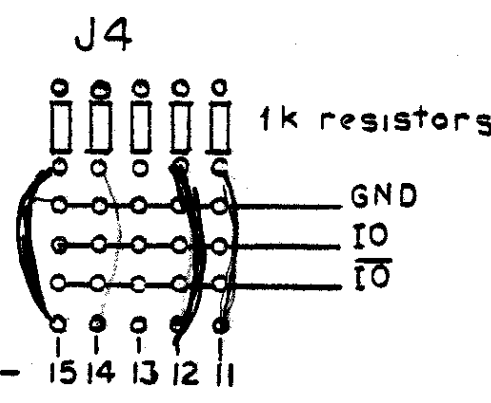
PROGRAMMABLE CHARACTER GENERATOR
PARTS PLACEMENT

OBJECTIVE DESIGN, INC.

PCG JUMPERS



B = 1:11



address

J1--u : Memory is not protected following power up or reset.
 J1--p : Memory is protected following power up or reset.

J2--1 : Bit 0 of ISL-0 will be 1 when data is ready.
 J2--I : Bit 0 of ISL-0 will be 0 when data is ready.

J3--s : Used when keyboards provide a positive going strobe.
 J3--S : Used when keyboards provide a negative going strobe.

J4
 This patch area will determine the memory and IO addresses of the PCG in the system memory map. There are four possible cases for each address line.

- A) Both IO and address bits are 1: no jumper required.
- B) IO bit is 0 and address bit is 1: jumper from adrs to \overline{IO} .
- C) IO bit is 1 and address bit is 0: jumper from adrs to IO.
- D) Both IO and address bits are 0: jumper ^{addr} to gnd.

*

*

**** PCG Memory Test ****

*

* This test is adapted -- with little change --

* from the program by T. E.

* Travis which appeared in the December,

* 1976 (vol. 2, issue 1) edition

* of INTERFACE AGE.

*

*

*

DONE EQU 0000H * AFTER TEST, GO HERE

*

* The user must supply an address for 'DONE' in the operating

* system to return control when the test is over.

*

PCGAD EQU 0C000H * THIS MUST BE THE AD OF THE PCG MEM

*

*

START LXI B,PCGAD+0800H * C=0, B IS PAGE PAST PCG MEM

REP LXI H,PCGAD * POINTER TO MEM

FILL MOV A,L * LOW BYTE OF ADDRESS

XRA H * EX-OR WITH HI BYTE

XRA C * EX-OR WITH PATTERN MODIFIER

MOV M,A * PLACE TEST PATTERN IN MEM

INX H * POINT TO NEXT MEM BYTE

MOV A,B

CMP H * HAVE WE GONE BEYOND MEM?

JNZ FILL * NZ--NO, CONTINUE FILLING MEM

*

* The entire 2K memory is filled with a special pattern.

* Now we check each byte for the correct data.

*

TEST LXI H,PCGAD * START OVER

MOV A,L * LOW BYTE AGAIN

XRA H * FORM THE SAME PATTERN

XRA C * MODIFIER AGAIN

CMP M * COMPARE WITH MEM

CNZ ERR * NZ--DATA IS INCORRECT!!

INX H * GO TO NEXT BYTE

MOV A,B

CMP H * HAVE WE CHECKED ALL MEM?

JNZ TEST * NZ--NO, CONTINUE.

*

*

* A complete check has been made with one pattern. We now

* go to the next pattern by changing the pattern modifier.

*

INR C * CHANGE MODIFIER

JNZ REP * NZ--DO NEXT PATTERN LOOP

JMP DONE * TEST IS OVER

*

* When the modifier is back to zero, we will have done 256 pattern

* checks--all this test will do without repeating a pattern.

*
* The user must add the ERR subroutine. It can simply stop the
* test or indicate the error, report the address and data, and
* return to the test.

*
ERR PUSH H * HL HOLD THE ADD OF ERROR
 PUSH B * WE NEED IT
 NOP * USER ROUTINE ???
 NOP * A REG. HOLDS PATTERN THAT
 NOP * WE SHOULD HAVE FOUND
 POP B
 POP H
 RET

*
*
*
* END PROGRAM

1

```

0000
0000
0000
0000
0000
0000
0000 01 00 00
0003 11 00 00
0006 D3 90
0008 CD 2B 00
000B DB 90
000D E6 3C
000F CA 32 00
0012 OF
0013 OF
0014 OF
0015 D2 19 00
0018 04
0019 OF
001A D2 1E 00
001D 0C
001E OF
001F D2 23 00
0022 14
0023 OF
0024 D2 28 00
0027 1C
0028 C3 08 00
002B
002B 3E 0C
002D 3D
002E C2 2D 00
0031 C9
0032
0032
0032
0032
0032 21 00 90
0035 70
0036 23
0037 71
0038 23
0039 72
003A 23
003B 73
003C 76
003D
003D
READY
INIT 0000      CHK 0008      DELAY 002B      REPRT 0032

```

```

*
***** JOYSTICK TEST PROGRAM *****
*
* JOYSTICK VALUES RETURNED IN MEM 9000-9003
* EXAMPLE FOR PCG IO AT 90H
*
INIT      LXI      B,0      * CLR COUNT
          LXI      D,0      * CLR COUNT
          OUT      90H      * TRIGGER ALL MONOS
CHK       CALL     DELAY     * 100 US
          IN       90H      * IN MONOS AND PCG STATUS
          ANI      3CH      * ASSUMES ALL POTS CONN.
          JZ       REPRT     * ALL DONE
          RRC
          RRC
          RRC      * MONO AT PIN 12 IN CRY
          JNC      $+1      * NC -- THIS MONO DONE
          INR      B        * STILL HIGH, ADD TO COUNT
          RRC      * MONO AT PIN 13
          JNC      $+1
          INR      C
          RRC      * MONO AT PIN 14
          JNC      $+1
          INR      D
          RRC      * MONO AT PIN 15
          JNC      $+1
          INR      E
          JMP      CHK      * CONTINUE COUNTING
*
DELAY     MVI      A,12D    * APPROX. 100 US
          DCR      A
          JNZ      $-4
          RET
*
* THE USER ROUTINE CALLED REPRT SHOULD DISPLAY OR PLACE
* INTO MEMORY THE CONTENTS OF THE FOUR REGS. FOR EXAMPLE,
* WE ASSUME THE PCG MEM IS AT 9000H
*
REPRT     LXI      H,9000H
          MOV      M,B
          INX      H
          MOV      M,C
          INX      H
          MOV      M,D
          INX      H
          MOV      M,E
          HLT      * BETTER--JMP TO SYSTEM
* EXAMINE 9000-9003 FOR COUNTS
*

```

PCG Manual Errata:

P. 21 The seven (7) resistors on the PCG are now 5100 ohms, not 1000 ohms as listed in the parts list and etched on the board.

P. 21 The four (4) .1 MFD ceramic disc capacitors used for joystick timing are the same as the 12 used for filtering.

P. 22 24 pin to 24 pin connector - A length of two (2) feet will work in the SOL. We will begin supplying these on orders specified for the SOL as soon as our supply of 36 inch connectors is depleted.

P. 26 Ready Line - Users should cut the short PC line between the two component size holes at bottom cent on the back of the board. (Just above S-100 connector finger 72.) This removes the Ready line and prevents Wait states. If random memory fails occur, reconnect the Ready line with a jumper across the two holes.

P. 27 The filter capacitor shown between ICs P and Q may not fit when certain makes of DIPs and capacitors are supplied. In this case, leave the cap off the board. We have noted no failures resulting from this practice. Do not force the parts into position!