GTEK

Model 7344 User's Manual

Document Number 7344V103.MAN

Date 27 February 1987

*** * * * Read This... If Nothing Else! * * * ***

The Model 7344 can program certain parts made by MMI, National, and Texas Instruments. You MUST be certain that before you try to program one of these chips, that you have selected the proper manufacturer and part number. The part must be oriented so that pin 1 goes toward the LED. You MUST install a 20 pin part in the 20 pin socket to attempt to read or program it. You MUST be certain, that even if you apparently have the right selection for the part, that the 7344 is capable of programming the part, IN PARTICULAR the Texas Instrument parts.

APPLY AC POWER BEFORE PUTTING DEVICES INTO THE PROGRAMMER!

SEE CHAPTER ABOUT BAUD RATES AND CABLES IF PROGRAMMER FAILS TO COMMUNICATE!

Please take a moment and write this information here; You need this information if (in the unlikely event) you need to call GTEK for information on this product!

7344 PROGRAMMER SERIAL NUMBER / Version : ____________________________
DISK SERIAL NUMBER :____
PALX2 Version :____
GPC Version :____

TABLE OF CONTENTS

INTRODUCTION

Congratulations. You now have, what we believe to be, the most cost effective and advanced PAL/PLD programmer on the market today. The design philosophy used on the 7344 allows for simple future expansion of capabilities. All communications with the 7344 is in printable ASCII characters and it supports JEDEC and AHS hex formats.

Resident features include facilities for making source to pal content comparisons, blank checks, formatted device listings, menu driven device selection, and more.

The 7344’s interrupt driven type ahead buffer allows it to program and verify in real time, while data is being typed in from the keyboard. The model 7344 programs and verifies in real time transparent to the user, whose sole responsibility is to send and receive data. The standard algorithm prereads cells prior to programming, skips the cell if it is not necessary to program it, and post verifies the cells to as sure that it is properly programmed. Extended diagnostics pinpoint the cause of errors.

The Model 7344 may be used without handshaking, or with XON/XOFF or hardware CTS/DTR handshake. Baud rate selection is done automatically through your inter face program or PALX2 and defaults to 2400 baud on power-up. Used in conjunction with any terminal or computer with an RS-232 port, the 7344 is capable of programming and reading the devices listed in the appendix of parts supported.

All voltages and pin configurations are set up by the onboard microprocessor and no personality modules are required.

Note on Syntax Errors. A **–Syntax Error** will result from the selection of an invalid command, and you will be returned to the command prompter. If you generate 4 syntax errors in a row, you will find that the 7344 has apparently “Locked Up”. The 7344 did not “crash”, but it is looking for a new baud rate to send at. You can “Unlock” it by sending spaces to it until it responds with a prompter.

There is no reason for a reset button or on–off switch. The programmer will never “crash”. If you do get into a situation where you can’t get the programmer to respond to you (usually due to persons not using the PALX2 program), remove all parts from the programming sockets and unplug the programmer, wait about 10 seconds and then plug it back in and start over.

Any of the following commands that apply voltages and currents that could damage an improperly inserted or selected part will ask that you confirm its selection by answering (Y/N). To execute the command simply type Y or N.

B—Blank check command

The B command causes the programmer to check the part in the selected socket for being Blank. This is done by comparing it to what a blank part looks like in RAM. If you have data in the RAM it is not destroyed. Blank check will not check an empty socket with a simple pal (one that has phantom fuses). With selected parts like 16R8, the socket will appear to be blank if there is no PAL inserted.

Example:

```
MMI–10L8>Blank (Y/N) Y
Check Sum = 00
MMI–10L8>_
```

If the part is not blank, an error message will say so, otherwise the check sum of the RAM buffer is printed (not the socketed part).

C—Checksum Command

The C command causes an 8 bit checksum of the RAM buffer to be displayed. Most commands do an automatic checksum when they are invoked such as the Verify, Program and Load commands. This checksum has no relation to any other (externally generated) checksum, and is proprietary to the GTEK programmer.

Example:

```
MMI–10L8>Checksum
Check Sum = 00
MMI–10L8>_
```

D—Display Command

DB – The DB command causes the RAM buffer to be displayed on the screen in a formatted HEX dump. Each HEX character represents four input bits. It is read left to right across the screen. Every 8 characters represents 1 product term. There are 32 characters on each line, representing 4 product terms. In the following representation, the input and product lines have been numbered, the number being the first input line or product term for the character eg:

```
              0            1            2            3
   00 01 12 22  00 01 12 22  00 01 12 22  00 01 12 22
   04 82 60 48  04 82 60 48  04 82 60 48  04 82 60 48
00 5F FF BF FF  EB FF BF FF  FF F7 FF FF  00 00 00 00
04 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
08 B7 FB FF FF  9F FB FF FF  E7 FB FF FF  FF FF 7F FF
12 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
16 FF 5F BF FF  FF AF BF FF  FF F7 FF FF  00 00 00 00
20 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
24 FF 9B FF FF  FF 6B FF FF  FF FF 7F FF  00 00 00 00
28 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
32 FF FF B7 FF  FF F7 FF FF  00 00 00 00  00 00 00 00
36 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
40 FF FB FB FF  FF FF 7F FF  00 00 00 00  00 00 00 00
44 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
48 FF FF BF 7F  FF FF BF EB  FF F7 FF FF  00 00 00 00
52 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
56 FF FB FF B7  FF FB FF 9F  FF FF 7F FF  00 00 00 00
60 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
```

Use the numbers along the top and side in bold type for X and Y Cartesian coordinates. As a reference, 3 nibbles have been printed in bold face type.

Example 1: The B represents input numbers 16 (1), 17(1), 18 (0) and 19 (1) of product term number 00 (1101).

Example 2: The 7 represents input numbers 16 (0), 17 (1), 18 (1) and 19 (1) of product term number 26 (0111).

Example 3: The 9 represents input numbers 24 (1), 25 (0), 26 (0) and 27 (1) of product number 57 (1001).

Any bit that is 0 (zero) represents an INTACT fuse. A 1 represents a BLOWN fuse.

Example 4: The entire product term for product number 40 for instance will translate to (in terms of blown and intact fuses):

```
(input #s) 0000 0000 0011 1111 1111 2222 2222 2233
           0123 4567 8901 2345 6789 0123 4567 8901
(fuses)    1111 1111 1111 1011 1111 1011 1111 1111
            F    F    F    B    F    B    F    F
           where 0 represents an intact fuse
           and 1 represents a blown fuse
```

DH – The DH command causes an Ascii Hex Space format of the fuse buffer to be displayed to the screen. You shouldn't have to know the following in formation, but it is here for those inquiring minds. Each nibble of the display below represents four product terms for one input line. The first nibble (lsb through msb) represents product terms 0, 8, 16 and 24 for the first input line (#0). The second nibble (lsb through msb) represents the second input line (#1) for products 0, 8, 16, and 24, and so on through the last nibble (lsb through msb), which represents product lines 0, 8, 16 and 24 for the 32nd in put line (#31).

The next line, first nibble (lsb–msb) represents product lines 1, 9, 17 and 25, input line #0, the second (lsb–msb) represents product lines 1, 9, 17 and 25, input line #1 and so on to the last nibble (lsb–msb) on that line (input line #31 products 1, 9, 17 and 25). The product line numbers advance down to the line that contains product 7, 15, 23 and 31, and the next line changes to products 32, 40, 48 and 56, and advances as above. In other words, each nibble represents one input line and 4 product terms.

```
   input   0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3   Products
numbers 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 (bit positions)
                 E D E F D F F F B 7 3 F F 5 F F F A F F F F F F F F F F F F F F
24 16 08 00
                 F D D E F E F F 7 B F 3 F 5 F F F A F F F F F F F F F F F F F F
25 17 09 01
                 F F F D D F F F F F F F A D F F 7 F F F F F F F F F F F F F F F
26 18 10 02
                 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
27 19 11 03
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
28 20 12 04
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
29 21 13 05
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
30 22 14 06
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
31 23 15 07
                 F F F F F F F F F F F F F 5 F F F A F F E D F F B 7 F F 7 F F F
56 48 40 32
                 F F F F F F F F F F F F E 7 F F D B F F F F F F F 7 7 B F B F F
57 49 41 33
                 C C C C C C C C C C C C 8 C C C 4 C C C C C C C C C C C C C C C
58 50 42 34
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
59 51 43 35
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
60 52 44 36
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
61 53 45 37
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
62 54 46 38
                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
63 55 47 39
```

In the above representation (which is the same data as in the DB command), to locate the same information as in the previous examples, you must take one bit from four input lines from the bit position that represents that product term. For input lines 16, 17, 18 and 19 in product term 0, that is the first bold face block above.

Example 1:
```
input numbers:   16   17   18   19
      product:  2100 2100 2100 2100
      numbers:  4680 4680 4680 4680
         data:  1111 1010 1111 1111 (bit representation of first marked line)
 Signif. Bits:  1    0    1    1    is B for product line 24
```

Example 2:
```
input numbers:   16   17   18   19
      product:  2110 2110 2110 2110
      numbers:  6802 6802 6802 6802
         data:  0111 1111 1111 1111 (bit repr. of second marked line)
 Signif. Bits:  0    1    1    1    is 7 for product line 26
```

Example 3:
```
input numbers:   24   25   26   27
      product:  5443 5443 5443 5443
      numbers:  7913 7913 7913 7913
         data:  1111 0111 0111 1011 (bit repr. of FOURTH marked line)
 Signif. Bits:  1    0    0    1    is 9 for product line 57
```

Example 4:
```
       0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
input #0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
data   F F F F F F F F F F F F F 5 F F F A F F E D F F B 7 F F 7 F F F
PL 56  1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 (signif.
bits)
PL 48  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1       ″
PL 40  1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1       ″
PL 32  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1       ″
Notice how the binary lines add up to be what the HEX representation is like on
input 13, 0101 = 5. Input 17 adds up to 1010 = A.
Also see the chapter on Upload Formats.
```

DV - The Display Vectors command causes the current list of test vectors to be displayed. If you haven’t uploaded any, of course, you will not see any.

Example:
```
MMI–16L8>DV
V0000 XXXXXXXXXN1HZZZZZZHN
... up to end of vectors
V0127 111111110N0LLLLLLLLN
MMI–16L8>_
```

F Functional Test command

The F command will functionally test a PAL in the selected socket if you have uploaded test vectors to the programmer with the JEDEC command. Test Vectors may be manually entered using the TB command, following the JEDEC format. If tests are passed, no Error message is issued and control is returned to the programmer command state. If there are any errors, an error is issued for each output pin that is in the incorrect state. The vector line is displayed with the complaint to the side of it, and a BEEP is issued. Any “X” that is found on an input line is treated as if it were a “0”. The following example shows a test of 22 vectors with several errors in one of them (V0001) and one error on the last one (V0022).

Example:
```
MMI–12H6>Functional testing
V0001 11XXXXXXXNXXXXHXHHXN* – Level error on pin 15
V0001 11XXXXXXXNXXXXHXHHXN* – Level error on pin 17
V0001 11XXXXXXXNXXXXHXHHXN* – Level error on pin 18
V0022 XXXXXX10XNXXXHXXXXXN* – Level error on pin 14
MMI–12H6>_
```

If you are getting a lot of errors and wish to stop and return to the command prompter, hit the ESC key. See Appendix B and Compiler chapter.

L LOAD COMMAND.

The L command causes the selected part data to be loaded from the PAL into the RAM buffer. You must set the part type and manufacturer properly first before executing this command, because even though the part will load, chances are that the data that gets loaded is incorrect due to the fact that some parts have phantom locations. The programmer will ask you to confirm that you want to execute this command, so that you have to answer Yes or No. A checksum is issued after the part is loaded.

Example: MMI–10L8**L**oad (Y/N) **Y**

Check Sum = FC

MMI–10L8_

M MENU COMMAND.

The Menu command is used to select the Manufacturer and device type you intend to work with. The current Manufacturer and device type always be comes part of the command prompter.

Sending an M causes a menu to be output, from which the desired Manufacturer and device is then selected. If the code character for the device is already known, then just send M and the device will be selected. Selecting a device establishes the programming algorithm to be used, as well as the device pinout, proper programming voltage and prompter.

Every time that you make a menu selection, the prompter will default to MMI parts. If you are programming a series of Texas Instrument parts for in stance and change the part number, you will also have to reselect the manufacturer to TI again (M5).

Menu Example starting from the DOS prompter. Commands typed in from your keyboard are in Bold face type. <cr> means to strike the Return or Enter key:

Example from DOS :
```
C>PALX2<cr>
Pal Programmer Com. Package Version 3.03
Copyright 1983, 1986  GTEK, INC.
I/O Hardware Driver Vers 1.04 – IBM PC/XT/AT
Serial port  – COM22:,  2400 bps
Printer port – LPT1:
```

Initializing....

Z

GTEK Corp

Model 7344 V1.02

```
Copyright 1984, 1986
MMI–XXXX M<cr>
PAL  MENU
A – 10L8    I – 16C1    1 – MMI–    P – 12L10   W – 20X4
B – 12L6    J – 16L8    2 – MMIB–   Q – 14L8    X – 20X8
C – 14L4    K – 16R8    3 – NAT–    R – 16L6    Y – 20X10
D – 16L2    L – 16R6    4 – TI24–   S – 18L4    Z – 20L8
E – 10H8    M – 16R4    5 – TIB20–  T – 20L2    ! – 20R4
F – 12H6    N – 16X4                U – 20L10   @ – 20R6
G – 14H4    O – 16A4                V – 20C1    #  – 20R8
H – 16H2
;note these are 20 pins (above)     and these are 24 pins (above)

Enter Selection –M

;Select MMI 16R4 (default MMI)

MMI–;Select MMIB parts (valid selection)
MMIB–16R4 ;Select National parts
NAT–16R4 M4*– Select Error

;Error– 16R4 not 24 pin part, beeps...
NAT;Select TIB 20 pin part
TIB20–16R4 MB

;Select 12L6– defaults to MMI
MMI;Select B part. This is an invalid
;selection, but there is no error message
MMIB–12L6 M4;Select National part
NAT–12L6 M4*– Select Error
```

```
;Error, not a 24 pin part
NAT–12L6 M5*– Select Error

;Error, Not a TIB selection
NA;Select MMI 20R8
MMI;Select MMIB part
M;Select National part
NAT–20R8 ;Select Texas Instrument 20R8
TI24–20R8 M5*– Select Error

;Error– not a “B” part or 20 pins
TI24–20R8_



;done
```

See the selection chart in the appendix to select parts that are not on the programmer’s menu but can be programmed using this programmer.

P PROGRAM COMMAND.

Sending a “P” puts the 7344 into the program mode. Once in the program mode, the 7344 asks for you to also strike a Y for Yes or an N for No to confirm that you really mean to program the selected part, or not. If Y is struck, the content of the RAM buffer is programmed into the PAL. Once the process is started, you can’t stop it, so be careful. The type–ahead buffer allows you to issue commands, even though the programmer is busy, so you can enter a sequence of commands to set up the next process. When the part is programmed and verified, a checksum is issued, and you are returned to the programmer command prompter.

Example: MMI–10L8**P**rogram (Y/N) **Y**

Check Sum = FC

MMI–10L8_

Error messages can be “**Overblown Error**” and “**Underblown Error**” for any part. If there are no errors then the “**Check Sum =**” is displayed.

S SECURE COMMAND

Sending an “S” puts the 7344 into the secure mode. Once invoked, the 7344 asks you to confirm the use of this command with a Y. Any other response will return you to the command mode. Remember to program the part before you secure it, because you will not have access to the array after the part is secured.

Some parts use different algorithms and programming voltage levels from other parts. Selection of the correct manufacturer also selects the correct algorithm, so even though you can program parts like a National 10L8 with the MMI algorithm, you MUST SELECT National FOR THE MANUFACTURER BEFORE YOU TRY TO SECURE THE PART! If you try to secure a part with the wrong manufacturer selected, you will probably damage the part beyond use. The programmer cannot tell what part you have inserted and consequently it cannot know if the part is just damaged beyond use or if it is merely secured.

All of the 20 and 24 pin PALs can be secured by blowing the “Last Link” in side the PAL.

To secure simple gate PALs such as:

10H8 12H6 14H4 16H2 16C1 10L8 12L6

14L4 16L2 12L10 14L8 16L6 18L4 20L2

20C1

On simple PALs you must “use up” all of the unused product terms before you secure the part. This can be done in 2 ways. The first way, you must program all of the fuses except one input pair (low and high true) on all of the unused product lines of every output. This can be done by putting in “dummy” OR lines in your equation like the example below :

PAL14L4

last fuse programming

example for gate type PALs

A B C D E F G H I GND

J K L M N O P Q R VCC

```
/N =  A * B * /C  ; First Product Line

+  B * G * /H  ; Second Product Line

+  J * /J     ;  Dummy Product Line

+  J * /J     ;  Dummy Product Line
```

These two dummy product lines will blow all of the fuses on each product line except one (low and high) input pair to keep the product line from going true.

The input pair used for the “Dummy” should be an unused input, which should be tied either to ground or Vcc on your project. If all of the inputs are used you may use an input you are already using for something else. This, however, can cause glitches on that product line, so make sure that your PAL works in the project with the dummies you specified in your equations before you blow the security fuses. Also, if you have used up all the inputs and don’t have any left for using up unused terms, instead of using just any input, just duplicate one of the product terms you have used for the output to use them up. This is the same as tying inputs on an OR gate together.

```
/N =  A * B * /C  ; First Product Line

  +  B * G * /H  ; Second Product Line

  +  B * G * /H  ; Repeat Second Product Line

  +  B * G * /H  ; Repeat Second Product Line
```

This procedure is necessary on all of the unused OR lines of the simple gate PALs. Make sure that you have put only enough dummy product terms per output or GPC will complain. In other words don’t use product terms you don’t have available for an output. Of course if you use up all the OR terms, you will not have to follow that procedure.

All other PALs do not have to use this procedure. One reason you might want to use the procedure on other parts is that you may be able to cause lower power consumption by selective burning of unused product terms. You must still try to avoid glitches in the output the same way as described before, by using an unused input pin or the best way being by duplication of previously used terms for that output pin.

Example: MMI–16L8**S**ecure? (Y/N) **Y** Failed !

MMI–16L8_

The above example failed because the RAM buffer (before securing) looked the same to the 7344 as the part did after the secure procedure. The 7344 checks to be sure the content of the RAM buffer is the same as what's in the PAL before attempting to secure the part. In other words, before you try to secure the part, you have to have uploaded to the RAM buffer or loaded the RAM buffer from a programmed part.

MMI–16L8 **S**ecure? (Y/N) **Y**

Verify error

MMI–16L8_

The above example failed because the 7344 attempted to verify that the RAM buffer was the same as the part before trying to secure it. The verify procedure is the one specified by the manufacturer for the part (typically low/ttl/high Vcc).

MMI–16L8 **S**ecure? (Y/N) **Y**

MMI–16L8_

The above is an example of a part that secured properly without errors (no complaints means it secured OK). If you try to secure a PAL that has already been secured, without changing the content of the RAM buffer, you will get a Verify error, because the content of the PAL now looks different from what's in RAM. However, if you have test vectors loaded the PAL will test functionally of course. Remember that once you secure a PAL you can't use it for a master part any more.

TB JEDEC LOAD COMMAND

When in the command state, receipt of a TB is interpreted as meaning that a JEDEC file is about to be sent. Normally, you would never have to use this command, as the PALX2 program issues this command automatically to cause the programmer to accept the following JEDEC file about to be sent. If you can't use PALX2 because you don't use a PC/XT/AT type computer, see the chapter on writing interface software. When the buffer is loaded, a checksum is issued and you are returned to the programmer command prompter.

You could use this command to set or reset particular fuse locations in the RAM buffer. Certain things won't print on our printer here, so for clarity we will say that anything within a set of BRACKETS [] means one action, like pressing and holding the control key while pressing the letter B, eg: [control–B].

Example: MMI–10L8**TB[control–B]*L0 1*[control–Z]**

Check Sum = 01

MMI–10L8_

The above command example has the effect of setting the first real fuse in the real first product line on the first real input line in the buffer. Remember, if you want to do this for some reason, that Link addresses refer to Real Fuses. You can't just count over the number of positions you desire in the representation of our fuse buffer and know the correct Link address; you have to skip the Phantom locations within the buffer. Refer to the blank part pattern and the Logical representation of the PAL to figure the real Link address. Every place a real input line crosses a real product line is a real fuse which gets counted in figuring the Link address.

Also see the chapter on Upload format and the DB command.

TC AHS LOAD COMMAND

This command functions precisely the same way that the JEDEC LOAD COMMAND does, except the format is the AHS (ASCII HEX SPACE) format.

Also see the chapter on Upload format and the DH command.

V VERIFY COMMAND.

The V command checks the cells in the PAL for comparison with the data that is in the RAM buffer. If they are different, then an error message is issued, otherwise a checksum is displayed.

Example: MMI–10L8**V**erify (Y/N) **Y**

Check Sum = FC

MMI–10L8_

The Error messages for this command are "**Verify Error**" for TTL verification, "**Vcc Low Verify Error"** for low Vcc verify, and "**Vcc High Verify Error"** for high Vcc verify. No error, of course causes a "**Check Sum =**" to be printed.

X Calibrate command

The X command will cause the programmer to begin a process of calibration. Remove the screws on the bottom of the case and the 2 on the side and look for the rows of POTS. Each pot is marked, but the numbers are NOT IN ORDER, so you have to look at them to see which pot the programmer is as king you to adjust. Insert your digital meter probe into the locations requested and perform the required adjustment when it asks you to. Pressing the space bar will make the programmer skip to the next adjustment.

Before you try to calibrate the unit, you should turn it on for a few minutes prior to removing the case to allow the temperature to stabilize.

Example: C:\palsPALX2

Pal Programmer Com. Package Version 3.03

Copyright 1983, 1986 GTEK, INC.

I/O Hardware Driver Vers 1.04 – IBM PC/XT/AT

Serial port – COM22:, 2400 bps

```
Printer port – LPT1:

Initializing....
Z
GTEK  Corp
Model 7344 V1.02
Copyright 1984, 1986


;ALLOW  PROGRAMMER  to  WARM  5
MMI–20;MINUTES   BEFORE   CALIBRATION
CALIBRATION  –  REMOVE  PAL


;(beep) measure at the 24 pin socket
;Adjust to within .05 volts
At pin 24, adjust
VR1 for 4.5v
VR2 for 5.0v
VR3 for 5.5v
VR4 for 6.0v
VR5 for 10.5v
VR6 for 11.75v
;Beeps  here
At pin 1, adjust
VR7 for 4.5v
VR8 for 10.2v
VR9 for 11.75v
;Beeps  here
VERIFY
```

Pin 1 is at 20v.

;æ .25 volts

Pin 2 is at 20v.

;æ .25 volts

GND on all pins..

;check each pin on 24 pin socket

MMI–20R8_

;for ground (æ.20 volts)

Z Restart command

The Z command is a command that will cause the prompter to be reissued after displaying the Log in message. This information is necessary if you are going to call GTEK for answers to any questions about the 7344 or GPC.

Example: MMI-16L8**Z**

GTEK Corp

Model 7344 V1.02

Copyright 1984, 1986

MMI-16L8_

SPACE COMMAND (ASCII character 32)

Sending a space (ascii 32 char) to the programmer causes it to reissue the command prompter. The space command is also used to determine the baud rate during a baud rate seek.

Example: MMI-20R8_

MMI-20R8 **A** - Syntax Error

MMI-20R8 **A** - Syntax Error

```
MMI-20R8 A - Syntax Error
MMI-20R8 A ;Now the 7344 appears to be locked up...
;apparently, no key has any effect
;(except up to 6  in a row!)
;The 7344 has now determined
GTEK Corp ;the baud rate and reissues
Model 7344 V1.02;command prompter
Copyright 1984, 1986

MMI-20R8_
```

ƒ Carriage Return command

The command has the effect of causing the prompter to be reissued when done from the command prompter. A carriage return is not used for most of the programmer commands. They take immediate effect when the proper selection is made. A
has no effect on any command except when you are using the Menu command. A M
will cause the menu of generic part numbers to be displayed.

General

1. Error Code Display

Error codes may be issued in 2 ways:

A. After command eg:

```
>Verify (Y/N) Y
Verify Error
>_
```

which indicates a fatal error with the part, and

B. At the prompter eg:

```
>A- Syntax Error
>_
```

which indicates a fatal error with the command.

2. Error Code Return

Errors cause the programmer to return to the command state.

3. Error Code Timing

Errors are output on a real time basis, i.e., they are output as soon as they are detected.

Prompter Error Messages

Errors are preceded with a BEEP to alert you that an error has occurred.

- Syntax Error

Syntax error means you have typed an invalid command letter(s). You are returned to the command prompter to reissue or correct the command that you issued. Four Syntax Errors in a row without a valid command being issued (like a <space> or ↵) will cause the programmer to assume that you want to change baud rates. Issuing up to 6 ace commands in a row will cause the programmer to lock onto the new baud rate and reissue the command prompter.

- Select Error

Select Error means you have tried to select a Manufacturer / part combination that is invalid.

Command Error Messages

Overblown Error

Means that during the Verify process, fuses were detected blown, that are not supposed to be blown. A part can verify good at TTL levels, while not verifying correctly at reduced or elevated Vcc levels. This error is usually associated with the PROGRAM command.

Underblown Error

Means that during the Verify process, fuses were detected un-blown, that are supposed to be blown. A part can verify good at TTL levels, while not verifying correctly at reduced or elevated Vcc levels. This Error is usually associated with the PROGRAM command.

Verify Error

Means that during the Verify process, the RAM buffer did not compare with the PAL, with Vcc at TTL levels. Fuses might be blown that are not supposed to be (blown), or un-blown when they are supposed to be (blown). This error message is usually associated with the VERIFY command.

Verify Low Error

Means that during the Verify process, the RAM buffer did not compare with the PAL, with Vcc at the low limit. Fuses might be blown that are not sup posed to be (blown), or fuses might be un-blown when they are supposed to be (blown). This message is usually associated with the VERIFY command. Most MMI parts do not use a low level Vcc during verification.

Verify High Error

Means that during the Verify process, the RAM buffer did not compare with the PAL, with Vcc at the high limit. Fuses might be blown that are not sup posed to be (blown), or fuses might be un-blown when they are supposed to be (blown). This message is usually associated with the VERIFY command.

INTERFACINGNOTES

The Model 7344 comes supplied with a program called PALX2 to operate the 7344 on IBM type PC/XT and AT's. If you want to run the programmer on a different type computer (CP/M, VAX) you will probably use a simple communications program.

There are 2 requirements necessary to operate the 7344. Number 1 is for the program to have a "Dumb Terminal" mode. That's easy, nearly all communications programs have dumb terminal modes. The second requirement is that you be able to Upload (send data from the computer to the programmer). In the event that you do not have a program like this, one is easily written.

The Model 7344 is surprisingly easy to interface and there are several methods of handshaking which can be utilized if it is desired to operate at the higher baud rates. The following section describes some of the methods. The FIFO contains 16 characters to allow you to "type ahead" by at least that many characters during the programmer operation.

1. Software handshake

This is perhaps the easiest method of all. When you begin to send data to the programmer, simply wait for the character that you sent to be sent back. In some cases, the programmer wants to send more characters back to you than what you sent, so you can display any other characters that are sent back also. When the programmer is operated from the command mode, you will always get characters back from the programmer and the last character that is always sent is the greater–than sign ">". You can wait for that symbol to come back before sending any more characters to the programmer.

2. CTS/DTR hardware handshaking

The Model 7344 is configured as data terminal equipment, which means that the CTS (clear to send) line is an input to the programmer which when pulled low forces the programmer to stop sending. On the other hand, the DTR (data terminal ready) line is an output from the programmer, which will go low when the buffer contains 2 or more characters and high again when there are less than 2 characters in the FIFO. If you are using hardware handshake and the DTR line goes low, you should stop sending to the 7344. The RTS line is pulled high whenever the programmer is plugged in. See Specifications for Cable.

3. Xon/Xoff software handshaking

If you do not monitor the DTR line, the 7344 will transmit an XOFF character if there gets to be 4 characters in the FIFO. When the FIFO level drops below 4 characters, an XON will be transmitted. Likewise, when the programmer is sending you data, you may send an XOFF character, which will stop the programmer from sending until it receives an XON character. XON's and XOFF's, are not put into the FIFO, but are processed as soon as they are received. Even if you don't use XON/XOFF handshaking, you will find it useful when using the display commands, to stop and start the data flow to your screen. XON and XOFF are the keyboard equivalents of control-Q and control-S respectively.

4. Establishing Baud Rate

The 7344 defaults to a 2400 baud rate on power up. It expects that the next characters you send will be valid commands. Even if your program is going to run at 2400 baud, you should send at least 6 <space> commands to the programmer every time you invoke your communications program, to insure that you will be able to communicate with the programmer when you return to the command state. See the flow chart for interface software examples.

HINTS

When you automate the transfer of data from your computer to the 7344, you should examine the echoed characters to see if a dash, “-” has been sent. If you receive one, it means that an error message will follow and that the programmer will return to the command state. Any automation software should take this into account.

You don’t need to compare the characters that are echoed to what you sent. The characters are echoed to the host as they are removed from the FIFO, and would not reflect an error. However, the 7344 will detect any programming error and the host need only trap the error message. Remember that since the JEDEC and AHS files are echoed as they are received, they may contain dashes (–) in them in any comments.

The programmer is in the command state after the prompter is sent. The prompter always ends with a “>”. You can use this character to let your program know when a command has finished.

You have to have one mode of operation that turns your computer into a dumb terminal so that you can operate the programmer. Or once you have uploaded the data to the programmer, you can disconnect the RS-232 cable from the computer and attach it to a terminal for production operation. As cheap as IBM compatibles are today, you can probably buy a compatible much cheaper than a good terminal and turn the 7344 into a complete development system with that computer.

SEE APPENDIX FOR FLOWCHART EXAMPLES

Dimensions (H x W x D):
3.2” x 8.3” x 8.8”
(81mm x 211mm x 224mm)

Power Requirements:
120VAC, 60HZ, 10VA
(240VAC, 50/60Hz optional)

Interface Connector:
DB25P configured as Data Terminal Equipment.

Data word size:
1 Start bit, 8 data, 1 stop bit, no parity

Auto Select Baud Rate:
300, 600, 1200, 2400

Weight:
5 lbs
2.22 Kg

Operating Environment:
55 - 85° F.
12 - 29° C.
5% to 95% non-condensing relative humidity.

MAKINGACABLE

(for those people who can’t use our cable or want another one)

PROGRAMMER INTERFACE

The Model 7344 has a DB25P connector configured as Data Terminal Equipment (DTE).

Programmer:

Pin#	Abbreviation	Direction	Function
1	EG	Ground	Equipment Ground 0v
2	TXD	Output	Transmit Data. ±12V
3	RXD	Input	Receive Data. ±12V
4	RTS	Output	Request To Send. Always +12V
5	CTS	Input	Clear To Send. +12V enables
6	DSR	Input	Data Set Ready. Not used.
7	SG	Ground	Signal Ground. 0V
20	DTR	Output	Data Terminal Ready. +12V when programmer Ready

Note that the CTS line on the programmer is pulled up internally, so that if you use a program or a cable that does not use CTS, then you should be able to communicate properly without making any jumpers in your cable. The RTS line is always +12 Volts making it easy for programming software to determine if the programmer is turned on (as long as you have connected it). GTEK software and cables make use of these features.

Refer to the previous page for information on making a cable for other than an IBM PC/XT or AT.

Connections for IBM PC/XT 25 pin DB to 25 pin DB

IBM PC DB25 Female		Programmer DB25 Female	
EG	1	1	EG
TXD	2	3	RXD
RXD	3	2	TXD
CTS	5	20	DTR
SG	7	7	SG
DTR	20	5	CTS
DSR	6	4	RTS
RTS	4	6	DSR

Connection for IBM PC-AT 9 pin DB to 25 pin DB

AT DB9 Male:		Programmer DB-25 Female	
CD	1	6	DSR
RXD	2	2	TXD
TXD	3	3	RXD
DTR	4	5	CTS
SG	5	7	SG
DSR	6	4	RTS
CTS	8	20	DTR
RD	9	nc	

JEDEC Format (as used by the 7344 and GPC)

GTEK supports the JEDEC format on its 7344 programmer, but some features of JEDEC are not supported. GTEK also adds a command to enable the programmer to automatically select the part number and manufacturer.

JEDEC Format (commands in approximate order):

[control-B] —The first character in a JEDEC file. EVERYTHING is ignored that is sent to the programmer until it encounters an asterisk (*).

comment You are allowed comments up to an asterisk, before or after the control-B.

Mx or Mn Menu command. There may be 1 or 2 Menu commands in a JEDEC file compiled by the GTEK GPC compiler. The first (Mx) is the part selection where x means a letter A-Z. The second (Mn) is the Manufacturer selection. The command is terminated with an asterisk (*). This is an added command to GTEK's JEDEC files to allow the 7344 to make the menu selection automatically.

Example:	***MA***	which causes an MMI–10L8 to be selected, RAM cleared.
	MA*M3	causes a National 10L8 to be selected, RAM cleared.
	No Mx	No part selection done, RAM not cleared.
	No Mn	No manufacturer selection done, RAM not cleared.
	M3	;National Part selected, RAM not cleared.

G0 or G1 This command is usually used to operate the security fuses of the part being programmed. It is terminated with an asterisk. The 7344 ignores this command.

Example: ***G0*** ;No effect on programmer

F0 or Fn This command will cause the fuse buffer to be cleared before loading data into it. The GPC (GTEK PAL Compiler) will always issue this command (F0*) for the 7344. It is terminated with an asterisk (F0*). The Menu command (Mx), as a side effect however, will do the same thing.

Example: ***F0*** ;Causes RAM buffer to be cleared

L This command will cause the 7344 to look for 1 to a number of ASCII–decimal characters terminated by a space (or a number of spaces). The 7344 then causes its Link counter to count over to the specified location in the RAM buffer.

The 1's and 0's that follow the space are then put into the RAM locations specified by the Link counter. The Link counter counts only the real input fuses in the RAM buffer and skips the phantom locations so that each character of information is stored in the proper place. Any number of 1's and 0's could be sent to the programmer after the first Link address is specified, up to the number of fuses in the part. This command is terminated by an asterisk (*). See also the Commands chapter under the TB command.

Example: `L0000 1111111111110111111111111011111*` ;The 4 0's after the L in the above line causes the programmer to set its Link address counter to 0 and point to the first real fuse in the selected part. This Link address could simply be one 0 or ten 0's in this case. A space, or a number of spaces, follows the Link address, which tells the programmer that the Link data, composed of 1's and 0's is about to follow. The command is terminated by an asterisk (*).

C This command is not used on the 7344 programmer. It is ignored.

V This command tells the programmer that Test Vectors are coming next. The V is followed by a number of ascii-decimal characters, up to a space (or a number of spaces). After the spaces, follows a number of characters making up a test vector, to tell the 7344 how to test the part. The programmer will store these vectors to be used with the Functional Test command. This command is terminated by an asterisk.

Example: `V0000 XXXXXXXXXNXXXXXLXXXN*`
The above test vector is stored in the RAM buffer to be used as instructions for testing the part with the Functional Testing command.

See some of the examples of JEDEC files in other chapters and the Appendix.

AHS Format (as used by the 7344 DH command)

The 7344 supports the AHS (Ascii-Space-Hex) format. The AHS format consists of a string of single ascii-hex characters separated by a space. The complete syntax is as follows:

Start Sequence

(the number pairs represent a single byte) 12 07 07 0D 0A comments and part number here... notice that the AHS for mat does not have any command to set the part type in the 7344, consequently, you have to set the part number IN ADVANCE! 0D 0A 01 02

Body (20 pin)

The following representation contains single ascii-hex characters (nibbles) separated by a space. Each of the nibbles contain information for 1 input line and 4 product lines. The input lines start at #0 and go through #31 (32 inputs), so there are 32 nibbles on each line.
The first line of nibbles represents product lines 00, 08, 16 and 24. The second line represents product lines 01, 09, 17 and 25, and so on 6 more times to product lines 07, 15, 23 and 31. The ninth line starts with products 32, 40, 48 and 56, which increment with each line to the last, down to product lines 39, 47, 55 and 63. The products are presented as from the least significant nibble up through the most significant nibble, which means on the first line, input number 0 (E in hex or 1110 in binary), represents product number 24 (1), 16 (1), 8 (1) and 0 (0).

The following is the same example from the DH command:

```
input    0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 Products
numbers  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 (bit psns)
         E D E F D F F F B 7 3 F F 5 F F F A F F F F F F F F F F F F F F 24 16  8 00
         F D D E F E F F 7 B F 3 F 5 F F F A F F F F F F F F F F F F F F 25 17  9 01
         F F F D D F F F F F F F A D F F 7 F F F F F F F F F F F F F F F 26 18 10 02
         2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 27 19 11 03
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 28 20 12 04
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 29 21 13 05
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 30 22 14 06
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 31 23 15 07
         F F F F F F F F F F F F F 5 F F F A F F E D F F B 7 F F 7 F F F 56 48 40 32
         F F F F F F F F F F F F E 7 F F D B F F F F F F F 7 7 B F B F F 57 49 41 33
         C C C C C C C C C C C C 8 C C C 4 C C C C C C C C C C C C C C C 58 50 42 34
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 59 51 43 35
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 60 52 44 36
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 61 53 45 37
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 62 54 46 38
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 63 55 47 39
```

For Example... for product line number 58:

```
         0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
input #  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
data     C C C C C C C C C C C C 8 C C C 4 C C C C C C C C C C C C C C C
PL 58    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 (signif. bits)
PL 50    1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1       ″
PL 42    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0       ″
PL 34    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0       ″
```

Notice how the binary lines add up to be what the HEX representation is like on input 12, 1000 = 8. Input 16 adds up to 0100 = 4.

Also see the Command chapter on the DH command.

Body (24 pin)

The following representation contains paired ascii-hex characters (bytes) separated by a space. Each pair of nibbles (bytes) contain information for 1 input line and 5 product lines. The input lines start at #0 and go through #40 (40 inputs), so there are 40 bytes (80 nibbles) on each line. There are so many characters that the display "wraps around" and continues to display on the next line. Your display is somewhat different from the following presentation because the printer can print more characters on a line than the display can.

The first line of bytes represent product lines 00, 08, 16, 24 and 32. The second line represents product lines 01, 09, 17, 25 and 33, and so on 6 more times to product lines 07, 15, 23, 31 and 39. The ninth line starts with products 40, 48, 56, 64 and 72, which increment with each line to the last, down to product lines 47, 55, 63 and 79. The products are presented as from the least significant nibble up through the most significant nibble, which means on the first line, input number 06 (1F in hex or 0001 1111 in binary), represents product number 32, (0), 24 (0), 16 (0), 8 (1) and 0 (1). The last 3 bits are not used (bits 2, 1 and 0). The following is the same example from the DH command:

```
MMI-12L10>dh                                    ;INPUT LINE NUMBERS
0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2
0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7
00 00 00 00 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F

2  2  3  3  3  3  3  3  3  3  3  3
8  9  0  1  2  3  4  5  6  7  8  9
00 00 1F 1F 00 00 1F 1F 00 00 00 00                                        ;00, 08, 16, 24, 
00 00 00 00 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F
00 00 1F 1F 00 00 1F 1F 00 00 00 00             ;01, 09, 17, 25, 33
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00             ;02, 10, 18, 26, 34
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00             ;03, 11, 19, 27, 35
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00                                        ;04, 12, 20, 28, 
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00                                        ;05, 13, 21, 29, 
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00             ;06, 14, 22, 30, 38
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00             ;07, 15, 23, 31, 39
00 00 00 00 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F
00 00 1F 1F 00 00 1F 1F 00 00 00 00             ;40, 48, 56, 64, 72
00 00 00 00 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F 00 00 1F 1F
00 00 1F 1F 00 00 1F 1F 00 00 00 00             ;41, 49, 57, 65, 73
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00             ;42, 50, 58, 66, 74
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 0              ;43, 51, 59, 67, 75
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 0              ;44, 52, 60, 68, 76
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 0              ;45, 53, 61, 69, 77
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 0              ;46, 54, 62, 70, 78
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 0              ;47, 55, 63, 71, 79
Check Sum = 40
MMI–12L10>_
```

End Sequence (the number pairs represent a single byte)

`14 0D 0A 1A` This ends the AHS format file. The `1A` is necessary for the 7344 to be able to tell when you don't want to send any more characters.

7344 RAM buffer format

20 pin format

The 7344 can also display a non-standard format to display the buffer that is somewhat easier to read, once you know what you are looking at. The following is a display of a 16L8 RAM buffer, followed by a explanation of what you are looking at. The notes to the right of the 0's are not displayed when you execute the DB command.

(next page)

```
MMI-XXXX>MJ
MMI-16L8>DB
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  1
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  2
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  3
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  4
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  5
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  6
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  7
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  8
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line  9
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line 10
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line 11
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line 12
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line 13
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line 14
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line 15
 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00 line 16
Check Sum = 00
MMI–16L8>_
```

In the above presentation each 0 represents 4 input lines. Each pair of 0's represents 8 input lines. Each group of 4 pairs represents 32 input lines, or 1 product term. The 4 pair groups (product terms) are separated by an extra space. Each line represents 4 product terms.

The product terms are numbered from left to right starting at 0, on the first line so that the first line contains product terms 0, 1, 2 and 3. The second line contains product terms 4, 5, 6 and 7, and so on down to line 16 which contains product terms 60, 61, 62 and 63.

24 pin format

The following is a representation of a 24 pin part buffer. The display does not contain the characters to the right of the 0's.

```
MMI-16L8MZ
MMI-20L8>DB
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  1
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  2
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  3
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  4
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  5
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  6
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  7
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  8
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line  9
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 10
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 11
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 12
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 13
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 14
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 15
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 16
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 17
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 18
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 19
 00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00 line 20
Check Sum = 40
```

In the above presentation each 0 represents 4 input lines. Each pair of 0's represents 8 input lines. Each group of 5 pairs represents 40 input lines, or 1 product term. The 4 pair groups (product terms) are separated by an extra space. Each line represents 4 product terms.

The product terms are numbered from left to right starting at 0, on the first line so that the first line contains product terms 0, 1, 2 and 3. The second line contains product terms 4, 5, 6 and 7, and so on down to line 20 which contains product terms 77, 78, 79 and 80.

7344 RAM buffer blank part format

Introduction

If you were to load a blank PAL, most would look as though it contained all 0's. The PALs that look like this are:

20 Pin
```
PAL16L2   PAL16L8/B  PAL16R8/B  PAL16R6/B  PAL16R4/B  PAL16X4
PAL16A4   PAL16P8    PAL16RP8   PAL16RP6   PAL16RP4
```

These 20 pin PALs will load and have a check sum of 00.

24 Pin
```
PAL20L2    PAL20L10   PAL20X10   PAL20X8   PAL20X4   PAL20L8/B
PAL20R8/B  PAL20R6/B  PAL20R4/B  PAL20RA10
```

These 24 pin PALs will load and have a check sum of 40.

PAL numbers not mentioned above have phantom fuses.

Some of the smaller 20 and 24 pin PALs have fuses in areas of the PAL that are not used. These fuses can show up in the RAM buffer as either blown (1's) or un-blown (0's).

PALs with blown phantom fuses are:

20 pin
```
PAL10H8   PAL12H6   PAL14H4   PAL16H2   PAL16C1
PAL10L8   PAL12L6   PAL14L4   PAL16L2
```

24 pin
```
PAL12L10  PAL14L8   PAL16L6   PAL18L4   PAL20C1
```

Other PALs having phantom un-blown fuses (intact) never show in the fuse buffer because they are 0's to begin with.

Non–Zero Blank Fuse Patterns

```
PAL10L8  PAL10H8
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
03 33 33 30  03 33 33 30  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
Check Sum = 00

PAL12L6
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 33 33 00  00 33 33 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 33 33 00  00 33 33 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
Check Sum = 00

PAL14L4
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
Check Sum = 00
```

```
PAL16L2  PAL16L8/B  PAL16R8/B  PAL16R6/B  PAL16R4/B
PAL16A4  PAL16X4
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
Check Sum = 00

PAL12H6
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
00 33 33 00  00 33 33 00  00 33 33 00  00 33 33 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 33 33 00  00 33 33 00  00 33 33 00  00 33 33 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
Check Sum = 00

PAL14H4
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 03 30 00  00 03 30 00  00 03 30 00  00 03 30 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
Check Sum = 00
```

```
PAL16H2
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
Check Sum = 00

PAL16C1
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
Check Sum = 00

PAL12L10
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
03 33 33 33 30  03 33 33 33 30  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
Check Sum = 40
```

```
PAL14L8
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 33 33 33 00  00 33 33 33 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 33 33 33 00  00 33 33 33 00  00 33 33 33 00  00 33 33 33 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
Check Sum = 00

PAL16L6
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 03 33 30 00  00 03 33 30 00  00 03 33 30 00  00 03 33 30 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 03 33 30 00  00 03 33 30 00  00 03 33 30 00  00 03 33 30 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 03 33 30 00  00 03 33 30 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 03 33 30 00  00 03 33 30 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 03 33 30 00  00 03 33 30 00  00 03 33 30 00  00 03 33 30 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 03 33 30 00  00 03 33 30 00  00 03 33 30 00  00 03 33 30 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
Check Sum = 00
```

```
PAL18L4
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 33 00 00  00 00 33 00 00  00 00 33 00 00  00 00 33 00 00
00 00 33 00 00  00 00 33 00 00  00 00 00 00 00  00 00 00 00 00
00 00 33 00 00  00 00 33 00 00  00 00 33 00 00  00 00 33 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 33 00 00  00 00 33 00 00  00 00 33 00 00  00 00 33 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 33 00 00  00 00 33 00 00  00 00 33 00 00  00 00 33 00 00
00 00 33 00 00  00 00 33 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
Check Sum = 40
```

```
PAL20L2   PAL20L10   PAL20X10/A  PAL20X8/A  PAL20X4/A  PAL20L8/B
PAL20R8/B PAL20R6/B  PAL20R4/B
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
Check Sum = 40
```

```
PAL20C1
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF  FF FF FF FF FF
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
00 00 00 00 00  00 00 00 00 00  00 00 00 00 00  00 00 00 00 00
Check Sum = 20
```

Source File Description

The PAL Design Specification is the outline specified to enable the GPC program to create a JEDEC file from Logic equations. This file has the PAL type, pin list, logical equations, a Description, and a Function Table to test the PAL after it is programmed. The format of this file is similar to the one that is used by Monolithic Memories, Inc. This file is called a PALASM source file.

The GTEK PAL COMPILER (GPC) is used to compile this Source code into a file on your disk in the JEDEC format. The format for the PAL Design specification is below :

<u>Line 1 Pal Part Number</u>

Line 1 is the PAL part number, left justified.

<u>Line 2 User Part Number</u>

Line 2 is the user’s part number, originator’s name, and date.

<u>Line 3 Device application name</u>

Line 3 is the title of your application, eg. Basic Gates

<u>Line 4 Address</u>

Line 4 is the user Company’s name, City, State etc.

<u>Line 5,6 Pin List</u>

The pin list is a sequence of symbolic names separated by one or more spaces (or tabs) on one or two lines. The first line will have the symbols for pins 1-10 of the 20 pin PAL or pins 1-12 of the 24 pin PAL. The second line will have the symbols for pins 11-20 on the 20 pin PALs or pins 13-24 on the 24 pin PALs. Each symbolic name is unique and must be different, unless it is not used in the equations.

All pins including power and ground must be defined. Pin 10 on 20 pin PALs or Pin 12 on 24 pin PALs must have the name “GND” for ground and pin 20 on 20 pin PALs or pin 24 on 24 pin PALs must have the name “VCC”. Any unused pins could be named with their pin number like “PIN4” or “PIN12” for example. Unused pins can also be called NC. NC can be used more than once.

NC should never be used as a real symbol. Pin names may use any printable character except the operators : ” = * + / ( ) . Pin names may be as long as necessary, although it will slow the compiler down slightly. Typing the lengthy names can be a chore, and will introduce errors when the symbol is spelled differently later in the equations.

You SHOULD NOT use the prefix “/” to logically complement the symbol name in the PIN LIST or the FUNCTION TABLE PIN LIST. Use it only in your equations.

<u>Line 7 Blank Line</u>

Line 7 must be blank. (carriage return/line feed)

Line 8 Equations

The logic functions desired from the PAL are defined in logic equations starting at line 8. These equations can be expressed in one of three forms :

1. Symbol = Expression
2. IF (Product) Symbol = Expression
3. Symbol := Expression

Definitions:

A **symbol** is a **pin name** with the optional **operator** “/”.

A **product** is a sequence of **symbols** separated by the **and** operator. (*)

An **IF(expression)** is a conditional equality to evaluate the tri-statable product in question to be either logically true (output) or logically false (tri-state output). **IF(VCC)** can be used to make the product true all the time.

An **expression** is a sequence of **symbols** separated by **operators**.

Operators

;	means that a **comment** follows.
/	means that the **symbol** is logically complemented.
*	means that the preceding **symbol** is logically **AND**ed with the **symbol** following.
+	means that the preceding **symbol** or **expression** is logically **OR**ed with the **symbol** or **expression** following.
:+:	means that the preceding **symbol** or **expression** is logically **XOR**ed (exclusive OR) with the **symbol** or **expression** following.
()	means that the contents are a conditional (tri-state) **expression** or fixed **symbol** to be evaluated for the tri-state control product line of the output pin.
=	means equality between the output side and the input side of the equations.
:=	means the output becomes what the equation is evaluated to after the low to high transition (or high to low in certain cases) of the clock pin. This is used on “Registered” parts.

Line nn FUNCTION TABLE

After the last equation, an optional Function Table is used to test the PAL after it has been programmed, or simulated after compilation. To use this feature, the keyword “FUNCTION TABLE” must be left justified in the file. Beginning on the next line, comments may be inserted anywhere afterward as long as they have the operator “;” in front.

The pin list should follow soon afterward. It should occupy only one line—reguardless of how long it is. Versions of GPC later than 1.06 will allow multiple line pin lists. This “pin list” is NOT the same as the first pin list you made. It contains only the names of the pins to be tested, and can NOT contain VCC or GND. DO NOT USE ANY LOGICAL OPERATORS LIKE “/” in the pin list. Following that is a dashed line, function table symbols, and then another dashed line. See the Appendix and Examples on making Function Tables for a complete description.

Line nn+n DESCRIPTION

After the Function Table, the keyword DESCRIPTION is used to show where the file ends. Notes about the PAL and its operation can be listed here. The compiler will completely ignore anything in the description lines.

Example PAL Source file :

```
PAL12H6 PAL DESIGN SPECIFICATION                                            ;LINE  1
GTEKP0007 WILLIAM C. EDMONDS 02/20/87                                       ;LINE  2
BASIC GATES                                                                 ;LINE  3
GTEK, INC., BAY ST. LOUIS, MS                                               ;LINE  4
CHARLIE DELTA FOX GOLF MIKE NOVEMBER POPPA QUEBEC INDIA GND                 ;LINE  5
JULIET KILO LIMA ROMEO OSCAR HOTEL ECHO BRAVO ALPHA VCC                     ;LINE  6
                                                                            ;LINE  7
BRAVO = /ALPHA                                  ;INVERTER CIRCUIT           ;LINE  8
ECHO  =  CHARLIE *  DELTA                       ;AND GATE                   ;LINE  9
HOTEL =  FOX                                    ;FIRST TERM OF OR GATE      ;LINE 10
      +  GOLF                                   ;SECOND TERM OF OR GATE     ;LINE 11
LIMA  = /INDIA                                  ;first term of nand gate    ;LINE 12
      + /Juliet                                 ;second term of nand gate   ;LINE 13
      + /Kilo                                   ;third term of nand gate    ;LINE 14
oscar = /Mike    * /November                    ;Nor Gate                   ;LINE 15
Romeo =  Poppa   * /Quebec                      ;first term of XOR          ;LINE 16
      + /Poppa   *  Quebec                      ;second term of XOR         ;LINE 17
                                                                            ;LINE 18
FUNCTION TABLE                                                              ;LINE 19
alpha bravo charlie delta echo fox golf hotel india juliet kilo             ;LINE 20
lima mike november oscar poppa quebec romeo                                 ;LINE 21
;                                     N                                     ;LINE 22
;      C                              o                                     ;LINE 23
;      h                   J          v        Q                            ;LINE 24
;A B   a D           H   I u          e  O   P u  R                         ;LINE 25
;l r   r e E      G  o   n l K  L   M m  s   o e  o                         ;LINE 26
;p a   l l c    F o  t   d i i  i   i b  c   p b  m                         ;LINE 27
;h v   i t h    o l  e   i e l  m   k e  a   p e  e                         ;LINE 28
;a o   e a o    x f  l   a t o  a   e r  r   a c  o                         ;LINE 29
---------                                                                   ;LINE 30
 L H   X X  X   X X  X   X X X  X   X X  X   X X  X                         ;LINE 31
 H L   X X  X   X X  X   X X X  X   X X  X   X X  X                         ;LINE 32
 X X   L L  L   X X  X   X X X  X   X X  X   X X  X                         ;LINE 33
 X X   L H  L   X X  X   X X X  X   X X  X   X X  X                         ;LINE 34
 X X   H L  L   X X  X   X X X  X   X X  X   X X  X                         ;LINE 35
 X X   H H  H   X X  X   X X X  X   X X  X   X X  X                             ;LINE 36
 X X   X X  X   L L  L   X X X  X   X X  X   X X  X                             ;LINE 37
 X X   X X  X   L H  H   X X X  X   X X  X   X X  X                             ;LINE 38
 X X   X X  X   H L  H   X X X  X   X X  X   X X  X                             ;LINE 39
 X X   X X  X   H H  H   X X X  X   X X  X   X X  X                             ;LINE 40
 X X   X X  X   X X  X   L L L  H   X X  X   X X  X                             ;LINE 41
 X X   X X  X   X X  X   L L H  H   X X  X   X X  X                             ;LINE 42
 X X   X X  X   X X  X   L H L  H   X X  X   X X  X                             ;LINE 43
 X X   X X  X   X X  X   H L L  H   X X  X   X X  X                             ;LINE 44
 X X   X X  X   X X  X   H H H  L   X X  X   X X  X                             ;LINE 45
 X X   X X  X   X X  X   X X X  X   L L  H   X X  X                             ;LINE 46
 X X   X X  X   X X  X   X X X  X   L H  L   X X  X                             ;LINE 47
 X X   X X  X   X X  X   X X X  X   H L  L   X X  X                             ;LINE 48
 X X   X X  X   X X  X   X X X  X   H H  L   X X  X                             ;LINE 49
 X X   X X  X   X X  X   X X X  X   X X  X   L L  L                             ;LINE 50
 X X   X X  X   X X  X   X X X  X   X X  X   L H  H                             ;LINE 51
 X X   X X  X   X X  X   X X X  X   X X  X   H L  H                             ;LINE 52
 X X   X X  X   X X  X   X X X  X   X X  X   H H  L                             ;LINE 53
---------                                                                       ;LINE 54
                                                                                ;LINE 55
DESCRIPTION                                                                     ;LINE 56
This example is used to illustrate the use of a 12H6 to implement              ;LINE 57
basic gate functions.   ;LINE 58
```

You might want to obtain data books from MMI or National to obtain more information about writing equations for a PAL file. The GTEK PAL COMPILER (GPC) can compile programs written in this format with some slight changes to their definitions of how the "/" operator works in the Pin List and the Function Table. (We think our way is more logical!)

Compiler Operation

See the chapter on using the GPC and PALX2 for a demonstration on how to compile and upload a PAL file. This compiler builds a "fuse buffer" from the PAL equations that are specified in the PAL file, and then uses that "fuse buffer" to generate a JEDEC file (with or without Vectors) and an XPLOT representation.

If there are any problems with your equations the GPC will complain, and depending on whether or not it is a FATAL error, it will either issue an "Error" message or a "Warning" message. An Error will terminate with a message right away and a Warning will generate a message and not terminate until the error becomes fatal.

There are some differences between the GPC (GTEK PAL COMPILER) and some other PALASM Compilers on the market.

(1) The Error checking in GPC is much more sophisticated and complete than other compilers.

(2) All tri-state outputs that have their tri-state control coming from the fuse array must have complete IF (symbol or expression) OUT = IN type equations.

(3) All outputs that have registers must have OUT := IN type equations.

Compiler Invocation Syntax

From the DOS command line (typically C>_), type the name of the compiler (GPC) and optionally the name of the file you want to compile. If you don't want the JEDEC output file to have the same name as the input file, then specify that also. You may also not use any options and let the program ask you for a command line once it's invoked. Parameters within brackets in the following examples are optional. Extensions are also optional as well as the drive and path specifications.

C> **GPC↵**
;this will cause the compiler to ask for a command line...

Enter Command Line :**palfilename.PAL [jedfilename.JED]↵**

;if there are no errors (see following section) then compiler commences execution.

C> **GPC palfilename.PAL [jedfilename.JED]↵**

;commences execution

C> **GPC palfilename.PAL↵**

;commences execution

A:\ **GPC b:\pals\palfilename c:\pals\jedfile\jedfilename↵**

Remember the .PTM files (PAL part number parameter files) must be on the same drive and in the same directory as the GPC program. GPCV107 will automatically use the environment variable GPC=pathname (eg. GPC=C:\GTEK\PTMS), if it is present, to find the .PTM files. GPC will automatically supply any extension. Also remember that you must have correctly supplied the Manufacturer and Part number on the first line of the PAL specification file (filename.PAL).

Compiler Error Messages

Notes about the Error Messages. The following error messages are current. At some time in the future GTEK will be adding, deleting or modifying these error messages in some way. Please look for an addendum sheet to correct or add or delete any of the following error messages. Typically the error message will become more descriptive of the problem, while still being listed in the same way as in this manual.

The words "symbol" and "source code line" and "pin name" (same as "symbol") are used as variables in the following messages. Whatever symbol, source code line, or operator the compiler was working on will be put there.

General Error Messages

Error... No Input File Name Was Specified!

This error will occur when you don't specify a file name on the DOS command line. It is not a fatal error, the program will then ask you for a command line or type "END" to stop.

Error... Bad File Name ---> filename.PAL (file not found or illegal file name)

This is an error that occurs when you specify a bad file name for the .PAL input file, either bad extension or file name. File names that are illegal (contain bad characters) are trapped also. Program allows you to go back and specify another filename.

Error... Disk Drive Door Open or Disk Not In Drive!

This an error that occurs when you have forgotten to put the disk in the drive and left the door open, or just left the door open. The error allows you to recover by making you close the door on the disk and starting over.

Unknown Error, Number ##

This is an error that occurs when something unresolvable and unidentifiable happens during the operation of the compiler. If you see this error, carefully check your source file for illegal usage of operators and/or make sure that your title/pin list/description/function table occurs on the correct lines. If it is still unresolvable, call GTEK at the number listed in the Warranty section to report the error. GTEK may not be able to help you on the telephone with this error, because it may require that you send your source code, demonstrating the error, for GTEK to examine. This error occurs only during parsing the command line or opening disk files.

Error... (partfilename.PTM) Is Not On Current Drive!

This is an unrecoverable error causing an abort to DOS. GPC has tried to find the file partfilename.PTM on the current drive or the same drive\path as the source file and was unsuccessful. This is because you either did not put the part file name correctly into the source file, or do not have the part file name specified on the disk or in the same drive or directory as the compiler or the source file.

GTEK can add new parts easily to your compiler by giving you new specification files, or you can do that yourself by reading the section in the appendix on PART FILES.

Warning... DESCRIPTION and/or FUNCTION TABLE Not Found!

This error is caused by a problem with your PAL specification file (palfilename.PAL) not containing a "DESCRIPTION" or "FUNCTION TABLE". Even if you do not use either, you should at least put "DESCRIPTION<cr><lf>" at the end of your file to show that there are no more equations (or any thing else for that matter) to follow. This is not a FATAL error, but the compiler assumes that you didn't put any thing else in the file after it found EOF. This could mean that the .JED file is incorrect, but not usually; if you only forgot to put the DESCRIPTION word in the source file, everything should be OK.

Error... Pin List is Invalid! Press Space Bar to Step Through.

This is a FATAL error. The compiler can't use the pin list you specified. This error occurs when you have put the pin list in the wrong place in the file, or didn't put one in the file at all, or used incorrect syntax. The compiler found what it thought was a pin list, but then realized it was wrong, and wants to show you what it got. This is helpful in determining what you did wrong. You are returned to DOS so that you can reedit your source code.

Error... Number --> # # in Area # # # #

This is a FATAL error. The compiler has an unresolvable error similar to the previous "UNKNOWN Error". The area is in the rest of the program, during expression evaluation. Call GTEK if you are unable to resolve the error by rewriting your source code. This may require that you send GTEK an example of code that causes the error, for this error to be resolved. You are returned to DOS so that you can reedit your source code.

Error... The Pin List Is NOT In The Proper Location!
It Should Begin On Line 5 Of The Source Code.

This is a FATAL error. You are returned to DOS so you can reedit your source code to resolve the conflict of where you put the pin list. It should begin on the 5th line of the code. The compiler actually starts looking for it on the 1st line and will go all the way down to the 10th line before it decides it can't find it. Your pin list MUST start on the 5th line to conform to the specification though.

Error... In Pin List, Pin (10 or 12) Is Not "GND"
But is symbol Instead!

This is a FATAL error caused by an incorrect pin list. It could be that the pin list doesn't begin on line 5 of the code or is simply the fact that you didn't put the fixed symbol "GND" in the pin list at position 10. You are returned to DOS to reedit the file after you type spaces to look through the pin list.

Error... In Pin List, Pin (20 or 24) Is Not "VCC"
But Is symbol Instead!

This is a FATAL error similar to the previous error for "GND".

Finding Output Pin In -->source code expression line

This is not an error, but simply the compiler telling you that it is compiling the line that is displayed here. If an error does occur, especially an UNKNOWN Error, it is while the GPC is working on this line.

Error... Ran Out of Characters in the Source Code Line!
source code line

This FATAL error occurs while the compiler is looking for symbols and operators to the right (input side) of the equals operator. The GPC did not find any symbols or operators. You are returned to DOS so you can fix your source code line that GPC complained about.

Error... Output Pin —>symbol<—WAS NOT FOUND!

This is a FATAL error that occurs when GPC discovers that the symbol you specified in the expression is not in the pin list you specified. You are returned to DOS to correct the error in your source file.

Error... You Can’t Use This Pin For an Output! -->symbol

This is a FATAL error that occurs when you try to use an input pin as an output pin (that cannot also be used as an output pin). You are returned to DOS to correct the error in your source file. The pin in question is not specified can be an output in the part file.

Error... Your Equation is Incorrect for This Part, Output Low True!
Before :symbol after: /symbol

This is a WARNING error (not fatal) that occurs when you specify in your expression that the output is high true (no / in front of symbol). The compiler assumes you meant it to be low true, since if the part is not programmable polarity, the equation is incorrect. If cannot specify a / in the pin list to make the output low true, because the / in the pin list is only used if the specified pin is an input. A / in the pin list is ignored if the pin is not an input. The GPC continues to compile the line after changing the sense of the output symbol.

Error... Equation Invalid... Missing Colon Before Equals...

This is a WARNING error (not fatal) that occurs when you forget to specify a : in front of the equals sign of a registered part. The colon is supplied for the clock. You may also have a FATAL error if you meant for the pin to be only an input. This is IMPOSSIBLE for a registered part, since the input to the array comes from the internal register and not from the output pin itself.

Error... Parentheses Can’t Be Used Here!

This is a FATAL error for any time that you use a parenthesis (open or close) in an expression and it is not used with the IF operator. The IF() operator may only be used on pins that are tri-statable and not registered. Information about the error is printed after the error message, and control is returned to DOS so that you may modify your source file.

Error... Nothing Found to the Right of the Equals Sign!

This is a FATAL error similar to the error “Ran out of characters in the Source Code line”. This time there was no expression on the input side, just comments or trash characters, and the line probably contains a conditional expression to evaluate a tri-statable product line. Control is returned to DOS so you can edit the source file.

Error... symbol Was Not Found In The Pin List!

This is a FATAL error caused either during the processing of a tri-statable product line or during the processing of symbols on the right side (input side) of the expression (to the right of the equals sign. The currently processing input pin is not found in the pin list. You are returned to DOS so you can examine the line displayed after “Finding Output Pin in-- source code line”.

Error... You Can’t Use This Pin As an Input!
symbol

This is a FATAL error caused by trying to use an output pin (or other pin like VCC, GND, tri-state control, or clock) as an input pin in the expression. It displays certain data about it and then control is returned to DOS so you can edit that line.

Error... Unknown Operator in the Source Code Line!
source code line
pointer into the line ↑
operator that caused the error

This is a FATAL error caused by using an operator (which may be perfectly valid), that can't be used in this position. Control is returned to DOS so you can edit your source file.

Error... Bad Operator Found!
—>source code line<—
pointer into the line (up arrow)
operator that caused the error
—>character before the operator<—

This is a FATAL error caused by GPC not finding the proper character to the right of a colon (:) where it was expecting either a plus (+) or an asterisk (*). You are returned to DOS so that you may edit that line in your source file. This happens when you are operating on an XOR (:+:) and it is improperly specified... like : + : for instance instead of :+:.

Error... Bad Part File or Source Code Line!
source code line

This is a FATAL error caused when you specify too many XOR operators (:+:) and there are no more products left. You are returned to DOS to correct the error. The first source code line is shown by "Finding Output Pin in -->" and the line that caused the error is after the error message. It could also be that the part you are using does not have enough XOR products.

Error... Found an Exclusive NOR!

This is a FATAL error caused when you specify an XNOR operator. GPC does not support XNOR. You are returned to DOS to correct the error.

Errors Generated During Processing of Conditional Equation

Error... No Open Parenthesis Found In Conditional Equation! source code line

This is a FATAL error that occurs when the IF operator has been found to the left (output side) of the equals sign and there is no conditional equation present (delineated with parenthesis on either side of it), or there is parenthesis around the conditional equation. You are returned to DOS to correct the problem in the source code.

Error... No More Characters in Conditional Equation Line! source code line

This is a FATAL error that occurs when GPC runs out of characters to process for the conditional equation before it encounters a closing parenthesis on the line. You are returned to DOS to correct the problem in the source code.

Error... Bad Symbol In Conditional Equation!
source code line

This is a FATAL error that occurs when GPC encounters a symbol that is not in the pin list during evaluation of the conditional expression. You are returned to DOS to correct the problem in the source code.

Error... symbol Was Not Found in the Pin List!
source code line

This is a FATAL error that occurs when GPC is looking up an input pin symbol in the pin list during the processing of a conditional expression. You are returned to DOS to correct the problem in the source code.

Error... Can't Use This Symbol For an Input!
symbol
source code line

This is a FATAL error that occurs for the same reason as above except the symbol was found, but it's not an input, with the same results.

Error... Missing Characters in File or Ran Into End of File Unexpectedly!
source code line

This is a FATAL error that typically occurs when your source file is bad (bad expression), or there is a disk I/O error while you were processing a conditional expression (with the partfilename.PTM). You are returned to DOS to correct the error in the source code or the part file (only if you made your own part file!).

Error... Unknown Operator in Conditional Equation Found!
source code line
pointer to character that caused the error ↑

This is a FATAL error that occurs when you have some operator that is not legally used in a conditional expression. You are returned to DOS to correct the source code.

Errors During Generation of the Output Files

Note that except for the following error, most of the errors following it are not FATAL, although most times, not all of the JEDEC file is generated. Most times it will leave off the Vectors or generate only an XPLOT file.

UNKNOWN ERROR ##

This is a FATAL error caused when an unresolvable error occurs during the generation of a JEDEC file, either in the Link area or the Vector area. If you get an error like this, you may have to send an example of the source code to GTEK before the error can be resolved. In general, if you will look at your source code and Function Table, to make it as compatible as possible with the examples given, you will correct your error. You are returned to DOS to correct the error.

NO Test Vectors In File...

This is not actually an error, unless you did put test vectors in the file. Maybe you did not use the proper format for a FUNCTION TABLE? A JEDEC file is created (without test vectors), along with an XPLOT file.

Error... Improper Function Table! No Pin List? last source code line used

This error occurs when the proper format for a Function Table pin list is not followed. Typically you have specified VCC or GND in it, or pins that don't match the original pin list, or a Dash (-) on a line somewhere before the pin list. A / is not used with the Function Table pin list. A JEDEC file is generated without the Test Vectors along with an XPLOT file.

Error... Found No Pin List?

Error... Problem with Pin List???

Error... Operator not allowed in function table pin list.

Error... No Pin List In Function Table!

No Test Vector Pin List Found!

The above errors are not FATAL, but the net result is that no Test Vectors are generated, only a JEDEC and XPLOT file. The typical problem is one of those from the previous "Improper Function Table!" error.

Error... Operator not allowed in function table pin list.

This is a non-FATAL error that occurs when you have used a legal operator in the pin list. See the previous section where it defines what a legal operator is. This is usually caused by a "typo" when you are typing in the function table pin list. A / is allowed, but has no effect.

Warning... Pin Number ## (symbol) Is Not Used In Any Of The Equations!

This is a non-FATAL error that occurs generating test vectors when you discover that a symbol you used in the test vectors was not used in ANY of the input expressions used for an output symbol. No test vectors are generated since GPC doesn't know how to handle that pin (no expression) in the JEDEC test vectors, and an XPLOT file is created.

Error... Pin Is Used As symbol. This Is Illegal In The Function Table!

This is a non-FATAL error that occurs when GPC cannot determine how you want to use the symbol in the function table, as an input or an output, when you specify H or L instead of 1 or 0 for an input or other combinations. You should always specify a 1, 0 or X when you want to use a pin as an input (as long as you CAN use it for an input) and H, L, Z or X if you want to use the pin as an output (as long as you can use that pin for an output). A display is made of what GPC was working on at the time the error occurred so that you might be able to modify the function table to correct the error. You must press a key to continue after you examine the data. Some part of the test vectors may be generated and then an XPLOT file is created.

Pin Added to Input List: ## symbol

This is not an error as such, fatal or otherwise, but what has happened is that a tri-state pin or clock pin was discovered in the pin list for the function table. Typically, the clock pin and the tri-state pin is not mentioned in the equation except as the := for a registered part or when you control the output to be tri-state on a part like a 16L8. The symbol is not mentioned in the expressions to be added to the list of pins used as output or input, so they are then added to that list here.

Error... Did Not Find a Dashed Line to Begin FT!

Error... Test Vectors Invalid or Don't Exist!?

This is a non-FATAL error message that occurs when there is a problem with the Function Table. You did not use the correct format to write your function table or it is not present. No test vectors are generated in the JEDEC file, and an XPLOT file is created.

Error... No More Than 127 Test Vectors...

This is a non-FATAL error message that occurs when you try to create more test vectors than the 7344 can handle at one time. If you need to generate more, then you must recompile with a different function table to test another 127.

Error... Illegal Symbol in Function Table! -->symbol
For Pin symbol
Number # #
Index # #
Symbol That Caused The Error Is symbol
display of symbols in pin list

This is a non-FATAL error that is caused by GPC finding a symbol that was allowed in the pin list that it cannot use in the test vectors for some reason. The information above is printed and then the pin list from the function table is displayed. You must strike a key to continue. Some or most of the test vectors may have already been generated, but no more after this message. The JEDEC file is completed and the XPLOT file is made.

Error... Pin Not Found In Equations!

This is a non-FATAL error that occurs when you have specified a pin in the Function Table pin list that is not used in the expressions. This error is similar to a previous error, except this one occurs later in the processing of the vector line, generally due to the way you specified its condition as 1, 0, H, L, C, K, P, or Z. The net result is the same though. You may have already generated some test vectors by the time this point is reached, but no more are generated. The JEDEC and XPLOT files are created and you are returned to DOS. If other data is displayed below the error message you have to strike a key to continue.

Error... Clock Can't Be Used Here!

Error... Improper Use of Function Table!

These are a non-FATAL errors that occur when you have specified a C (or an illegal character) on vector test line. A display is made of the items that caused the error (including the pin list) and no more test vectors are generated. You have to strike a key to continue after displaying the pin list. The JEDEC and XPLOT files are created and you are returned to DOS.

Internal Errors

The following errors can occur when the compiler has an unresolved internal error. The message describes what the compiler was doing when the error occurred. Some internal errors are described as UNKNOWN errors and are explained in previous sections.

DID NOT FIND INPUT PIN... FAILED...

This is a FATAL error that occurs when GPC is getting information from the partfilename.PTM file (eg: PAL12H6.PTM). It could be caused by incorrect data in your part file (if you created it yourself) or if the file is zero length or so forth. This error can also occur if you have used the wrong type PAL with your equations (like PAL10L8 instead of PAL16R8, or PAL16L8 instead of PAL16P8). You are returned to DOS to fix the problem.

Error... Never Found an Output Pin Name in This Source Code Line.
source code line

This is a FATAL error that occurs if the compiler loses track of which line it is working on. This can be caused by the expressions starting before or long after line 8. The last source code line the compiler thought it was working on is printed below the error message and you are returned to DOS to fix the problem.

Error... Part File Damaged, or Fatal Program Error!
source code line

This is a FATAL internal error that occurs when the partfilename.PTM is damaged (like when you created it yourself or it is now 0 length for some reason). GPC tried to access a byte in a location beyond the end of the file. You are returned to DOS to correct the error (probably by copying the correct part file from your backup copy or the MASTER disk).

INSTALLATIONofPALX2

PALX2, sold separately, is a communication program which runs on IBM PC/XT and AT's. It allows you to transmit (upload) JEDEC and AHS format files and then communicate in a dumb terminal mode with the 7344.

On the PALX2 program disk you will have 3 programs: PALX2.COM, PINSTALL.COM and GPC.EXE (plus some example files). PALX2 is the program used to communicate with your programmer. PINSTALL is the program that you must run to install the serial drivers in PALX2 so that you can communicate with the programmer. GPC (GTEK PAL COMPILER) is the program that you will use on your PAL Boolean equation files to create JEDEC files to upload to the 7344.

If you try to run the PALX2 program without installing the serial drivers, it will tell you to run the PINSTALL program. Remember that the PALX2 license is a single user license.

Insert GTEK program disk in drive A: and copy the programs to your hard disk with:

```
C>COPY  A:*.*
```

This will copy all the programs on the GTEK disk over to the subdirectory that you are logged on to on your hard disk. If you don't have a hard disk, use DISKCOPY or COPY to the B: drive. Refer to the DOS manual for specific instructions on using the COPY command. The desired end result is a backing up of the original GTEK copy. Store the original program disk in a safe place.

Now you should insert the backup copy in the drive A: and/or go to the subdirectory where PINSTALL and PALX2 are located. You must first run the PINSTALL program to install the serial drivers for PALX2.

Example installation of PALX2. Bold type is where you have typed a command or response.
 Means to strike the enter or return key.

Notes about the program are put to the side of the line, separated by a semi-colon (;). Example begins on next page...

```
C>pinstall↵

Serial Driver Installation Program Version 9.05
Copyright 1986, 1987  GTEK, INC.
All Rights Reserved,  worldwide.


Enter name of program to be installed --palx2

IBM PC/AT or compatible COM1 (IRQ4)
A - 57600 bps (AT only)
B - 28800 bps (AT only)
C - 19200 bps
D - 9600 bps
E - 2400 bps;Use this selection for (COM1:)
F - 1200 bps;or this selection
G - 300 bps;or this selection

IBM PC/AT or compatible COM2 (IRQ3)
H - 57600 bps (AT only)
I - 28800 bps (AT only)
J - 19200 bps
K - 9600 bps
L - 2400 bps;OR use this selection (COM2:)
M - 1200 bps;or this selection
N - 300 bps;or this selection

Z - no changes;or this selection


Enter selection --L
Select LPT# (1,2,3 or return for lpt1:) --
;not used on PALX2


Do you have a GTEK Super Serial Card on COM2:  (Y/N)? - y

OK! What port do you want the programmer on (0-7) ? --2
DONE
C>_
```

LPTn: is not used with PALX2. The GTEK PCSS-8 (Super Serial Card) is a serial communications card with 8 software selectable serial ports on it. If you have one (y) then you have to tell PALX2 what port to select to communicate with the 7344. After you type N or 2 in the above example (last 2 questions) you are returned to DOS with PALX2 having been set up for the specified parameters, in this case COM2: and 2400 baud on port 2 of the PCSS-8 serial card.

IRQ4 is used in conjunction with an interrupt service routine for COM1: when PALX2 is invoked, if you installed it for COM1:. This is a hardware line on your PC to give the system an interrupt whenever a character is received. If you know that something else in your computer is using this hardware interrupt line, then you should use the other com line, which uses IRQ3 (COM2:).

IRQ3 is also used in the same manner for COM2: when PALX2 is invoked, if you in stalled it for COM2:. If you know something in your system uses IRQ3 for interrupts, then you must use the other comport.

See the example for **C>PALX2** ↵ later in the manual.

Operation of PALX2

PALX2 is a “command driven program” as opposed to a “MENU driven program” which means that everything you do is done by entering a “command” on the command line instead of “selecting” the command from a menu. This makes the program very fast when you have learned what the commands are. All “commands” while running PALX2 are directed towards the programmer except for one, which is control-C. Control-C (pressing the control key and while holding it down, typing a C) will return control to DOS.

EXAMPLES:

Dumb Terminal example:

C> **PALX2**
Enter PALX2 and establish communication (dumb terminal mode) with the programmer (assuming everything is hooked up properly). This example, from the DOS command line, establishes communication with the programmer, and after log-on displays the Programmer Command Prompter, which is the currently selected Manufacturer and PAL type.

Upload example:

C> **PALX2FILENAME.JED**
C> **PALX2FILENAME.AHS**
Results in communication being established with the programmer and sending FILENAME.JED (JEDEC format) or FILENAME.AHS (ASCII-SPACE-HEX format) When PALX2 is through, you are returned to the 7344 command mode. The net result is, the RAM buffer is loaded with the data to burn the PAL. If select a part number at any time after you upload a file, you will have to upload the file again. GTEK JEDEC files automatically select a part number during upload, AHS files do not. Be sure to select a part number (in the dumb terminal mode) before uploading an AHS file.

Log on message example:

```
C>PALX2
Pal Programmer Com. Package Version 3.03
Copyright 1983, 1986  GTEK, INC.
I/O Hardware Driver Vers 1.04 - IBM PC/XT/AT
Serial port  - COM22:,  2400 bps          ;COM22: is GTEK PCSS-8 card COM2, PORT2
Printer port - LPT1:                      ;not supported in PALX2
Initializing....                          ;establishing communications here
Z                                         ;command used to display log-on message
GTEK Corp
Model 7344 V1.02                          ;model and version number
Copyright 1984, 1986
MMI-XXXX>_                                ;default power-on prompter...
                                          ;this stays set until you change it or
                                          ;remove power from the unit!
```

Example Compiler and PALX2 Session

The programmer is ready and waiting for a command at this point. This is the source file. It was created using EDLIN on an AT. See the chapter on GPC to interpret what each line means.

```
PAL12H6 PAL DESIGN SPECIFICATION                              ;LINE  1
GTEKP0007 WILLIAM C. EDMONDS 02/20/87                         ;LINE  2
BASIC GATES                                                   ;LINE  3
GTEK, INC., BAY ST. LOUIS, MS                                 ;LINE  4
CHARLIE DELTA FOX GOLF MIKE NOVEMBER POPPA QUEBEC INDIA GND   ;LINE  5
JULIET KILO LIMA ROMEO OSCAR HOTEL ECHO BRAVO ALPHA VCC       ;LINE  6
                                                              ;LINE  7
BRAVO = /ALPHA                     ;INVERTER CIRCUIT          ;LINE  8
ECHO  =  CHARLIE *  DELTA          ;AND GATE                  ;LINE  9
HOTEL =  FOX                       ;FIRST TERM OF OR GATE     ;LINE 10
      +  GOLF                      ;SECOND TERM OF OR GATE    ;LINE 11
LIMA  = /INDIA                     ;first term of nand gate   ;LINE 12
      + /JULIET                    ;second term of nand gate  ;LINE 13
      + /KILO                      ;third term of nand gate   ;LINE 14
OSCAR = /MIKE    * /NOVEMBER       ;Nor Gate                  ;LINE 15
ROMEO =  POPPA   * /QUEBEC         ;first term of XOR         ;LINE 16
      + /Poppa   *  Quebec         ;second term of XOR        ;LINE 17
                                                              ;LINE 18
FUNCTION TABLE                                                ;LINE 19
alpha bravo charlie delta echo fox golf hotel india juliet    ;LINE 20
kilo lima mike november oscar poppa quebec romeo              ;LINE 21
;                                     N                       ;LINE 22
;      C                              o                       ;LINE 23
;      h                   J          v        Q              ;LINE 24
;A B   a D           H   I u          e  O   P u  R           ;LINE 25
;l r   r e E      G  o   n l K  L   M m  s   o e  o           ;LINE 26
;p a   l l c    F o  t   d i i  i   i b  c   p b  m           ;LINE 27
;h v   i t h    o l  e   i e l  m   k e  a   p e  e           ;LINE 28
;a o   e a o    x f  l   a t o  a   e r  r   a c  o           ;LINE 29
---------                                                     ;LINE 30
 L H   X X  X   X X  X   X X X  X   X X  X   X X  X           ;LINE 31
 H L   X X  X   X X  X   X X X  X   X X  X   X X  X           ;LINE 32
 X X   L L  L   X X  X   X X X  X   X X  X   X X  X           ;LINE 33
 X X   L H  L   X X  X   X X X  X   X X  X   X X  X           ;LINE 34
 X X   H L  L   X X  X   X X X  X   X X  X   X X  X           ;LINE 35
 X X   H H  H   X X  X   X X X  X   X X  X   X X  X           ;LINE 36
 X X   X X  X   L L  L   X X X  X   X X  X   X X  X           ;LINE 37
 X X   X X  X   L H  H   X X X  X   X X  X   X X  X           ;LINE 38
 X X   X X  X   H L  H   X X X  X   X X  X   X X  X           ;LINE 39
 X X   X X  X   H H  H   X X X  X   X X  X   X X  X           ;LINE 40
 X X   X X  X   X X  X   L L L  H   X X  X   X X  X           ;LINE 41
 X X   X X  X   X X  X   L L H  H   X X  X   X X  X           ;LINE 42
 X X   X X  X   X X  X   L H L  H   X X  X   X X  X           ;LINE 43
 X X   X X  X   X X  X   H L L  H   X X  X   X X  X           ;LINE 44
 X X   X X  X   X X  X   H H H  L   X X  X   X X  X           ;LINE 45
 X X   X X  X   X X  X   X X X  X   L L  H   X X  X           ;LINE 46
 X X   X X  X   X X  X   X X X  X   L H  L   X X  X           ;LINE 47
 X X   X X  X   X X  X   X X X  X   H L  L   X X  X           ;LINE 48
 X X   X X  X   X X  X   X X X  X   H H  L   X X  X           ;LINE 49
 X X   X X  X   X X  X   X X X  X   X X  X   L L  L          ;LINE 50
 X X   X X  X   X X  X   X X X  X   X X  X   L H  H          ;LINE 51
 X X   X X  X   X X  X   X X X  X   X X  X   H L  H          ;LINE 52
 X X   X X  X   X X  X   X X X  X   X X  X   H H  L          ;LINE 53
----------                                                   ;LINE 54
                                                             ;LINE 55
DESCRIPTION                                                  ;LINE 56
This example is used to illustrate the use of a 12H6 to      ;LINE 57
implement basic gate functions.                              ;LINE 58
```

Send the source file to the compiler by typing the following:

```
C>GPC P0007 P7OUTS↵          ;the following is the output of the compiler...
```

```
GTEK, INC. Pal Compiler Version 1.0 January 1, 1987
Copyright 1986, 1987 by GTEK, INC.
 All Rights Reserved, World Wide.

MICROSOFT QUICK-BASIC COMPILER, COPYRIGHT 1982-1986

The Input File Name is :P0007.PAL  ;Default Drive\Path
     The Output File Name is :P7OUTS.JED;Default Drive\Path, Spec. Fn
   The PAL Part File Name is :PAL12H6.PTM;Default Drive\Path
Finding Output Pin In --BRAVO = /ALPHA;Compiling line
Finding Output Pin In --ECHO = CHARLIE * DELTA;compiling
Finding Output Pin In --HOTEL = FOX;compiling
Finding Output Pin In --LIMA = /INDIA;compiling
Finding Output Pin In --OSCAR = /MIKE * /NOVEMBER;compiling
Finding Output Pin In --ROMEO = POPPA * /QUEBEC;compiling

PAL12H6 PAL DESIGN SPECIFICATION
GTEKP0007 WILLIAM C. EDMONDS 02/20/87

BASIC GATES
*MF*G0*F0*                          ;Set 12H6, Leave Security Fuse, Fill 0’S
L0000 111111101111111111111111*    ;Link Address/Data
L0096 010111111111111111111111*
L0144 111101111111111111111111*
L0168 111111110111111111111111*
L0192 111111111110101111111111*
L0240 111111111111110110111111*
L0264 111111111111111001111111*
L0288 111111111111111111111011*
L0312 111111111111111111111110*
L0336 111111111111111111101111*
V0001 XXXXXXXXXNXXXXXXXH0N*        ;Vector/number/data
V0002 XXXXXXXXXNXXXXXXXL1N*
V0003 00XXXXXXXNXXXXXXLXXN*
V0004 01XXXXXXXNXXXXXXLXXN*
V0005 10XXXXXXXNXXXXXXLXXN*
V0006 11XXXXXXXNXXXXXXHXXN*
V0007 XX00XXXXXNXXXXXLXXXN*
V0008 XX01XXXXXNXXXXXHXXXN*
V0009 XX10XXXXXNXXXXXHXXXN*
V0010 XX11XXXXXNXXXXXHXXXN*
```

```
V0011 XXXXXXXX0N00HXXXXXXN*
V0012 XXXXXXXX0N01HXXXXXXN*
V0013 XXXXXXXX0N10HXXXXXXN*
V0014 XXXXXXXX1N00HXXXXXXN*
V0015 XXXXXXXX1N11LXXXXXXN*
V0016 XXXX00XXXNXXXXHXXXXN*
V0017 XXXX01XXXNXXXXLXXXXN*
V0018 XXXX10XXXNXXXXLXXXXN*
V0019 XXXX11XXXNXXXXLXXXXN*
V0020 XXXXXX00XNXXXLXXXXXN*
V0021 XXXXXX01XNXXXHXXXXXN*
V0022 XXXXXX10XNXXXHXXXXXN*
V0023 XXXXXX11XNXXXLXXXXXN*

PAL12H6 PAL DESIGN SPECIFICATION   ;XPLOT of P0007.PAL
GTEKP0007 WILLIAM C. EDMONDS 02/20/87
BASIC GATES                        ;X Represents an intact fuse!
                                   ;- Represents a blown fuse
8 ---- ---X --   --   --   --   ---- ---- ;Lines that are totally
                                   ;intact are not printed,
                                   ;they can never be
16 X-X- ---- --   --   --   --   ---- ---- ;true


24 ---- X--- --   --   --   --   ---- ----
25 ---- ---- X-   --   --   --   ---- ----


32 ---- ---- --   -X   -X   --   ---- ----


40 ---- ---- --   --   --   X-   -X-- ----
41 ---- ---- --   --   --   -X   X--- ----


48 ---- ---- --   --   --   --   ---- -X--
49 ---- ---- --   --   --   --   ---- ---X
50 ---- ---- --   --   --   --   ---X ----


NUMBER OF BLOWN FUSES = 306

Done Compiling...
PAL12H6 PAL DESIGN SPECIFICATION
GTEKP0007 WILLIAM C. EDMONDS 02/20/87
BASIC GATES
GTEK, INC., BAY ST. LOUIS, MS
C>_                                ;returned to dos here...
```

The invocation of the above compiler results in two files on the default drive (C:P7outs.JED and C:P7outs.PLT). Remember that if you don’t specify where to find the filename.PAL file or to put the filename.JED file, they will be put on the cur rent drive. The GPC program also expects that the files necessary to compile the part (.PTM files) are on the same drive and in the same directory as the GPC program.

Operation of PALX2

PALX2 is a "command driven program" as opposed to a "MENU driven program" which means that everything you do is done by entering a "command" on the command line instead of "selecting" the command from a menu. This makes the program very fast when you have learned what the commands are. All "commands" while running PALX2 are directed towards the programmer except for one, which is control-C. Control-C (pressing the control key and while holding it down, typing a C) will return control to DOS.

EXAMPLES:

Dumb Terminal example:

```
C>PALX2↵
```

Enter PALX2 and establish communication (dumb terminal mode) with the programmer (assuming everything is hooked up properly). This example, from the DOS command line, establishes communication with the programmer, and after log-on displays the Programmer Command Prompter, which is the currently selected Manufacturer and PAL type.

Upload example:

```
C>PALX2 FILENAME.JED↵

C>PALX2 FILENAME.AHS↵
```

Results in communication being established with the programmer and sending FILENAME.JED (JEDEC format) or FILENAME.AHS (ASCII-SPACE-HEX format) When PALX2 is through, you are returned to the 7344 command mode. The net result is, the RAM buffer is loaded with the data to burn the PAL. If select a part number at any time after you upload a file, you will have to upload the file again. GTEK JEDEC files automatically select a part number during upload, AHS files do not. Be sure to select a part number (in the dumb terminal mode) before uploading an AHS file.

Log on message example:

```
C>PALX2↵
Pal Programmer Com. Package Version 3.03
Copyright 1983, 1986  GTEK, INC.
I/O Hardware Driver Vers 1.04 - IBM PC/XT/AT
Serial port  - COM22:,  2400 bps   ;COM22: is GTEK PCSS-8 card COM2,
PORT2
Printer port - LPT1:               ;not supported in PALX2
Initializing....                   ;establishing communications here
Z                                  ;cmd used to display log-on message
GTEK Corp
Model 7344 V1.02;model and version number
Copyright 1984, 1986
MMI-XXXX>                          ;default power-on prompter...
                                   ;this stays set until you change it
                                   ;remove power from the unit!
```

Example Compiler and PALX2 Session

The programmer is ready and waiting for a command at this point. This is the source file. It was created using EDLIN on an AT. See the chapter on GPC to interpret what each line means.

```
PAL12H6 PAL DESIGN SPECIFICATION                              ;LINE 1
GTEKP0007 WILLIAM C. EDMONDS 02/20/87                         ;LINE 2
BASIC GATES                                                   ;LINE 3
GTEK, INC., BAY ST. LOUIS, MS                                 ;LINE 4
CHARLIE DELTA FOX GOLF MIKE NOVEMBER POPPA QUEBEC INDIA GND   ;LINE 5
JULIET KILO LIMA ROMEO OSCAR HOTEL ECHO BRAVO ALPHA VCC       ;LINE 6
                                                              ;LINE 7
BRAVO = /ALPHA                     ;INVERTER CIRCUIT          ;LINE 8
ECHO  =  CHARLIE *  DELTA          ;AND GATE                  ;LINE 9
HOTEL =  FOX                       ;FIRST TERM OF OR GATE     ;LINE 10
      +  GOLF                      ;SECOND TERM OF OR GATE    ;LINE 11
Lima  = /india                     ;first term of nand gate   ;LINE 12
      + /JULIET                    ;second term of nand gate  ;LINE 13
      + /KILO                      ;third term of nand gate   ;LINE 14
OSCAR = /MIKE    * /NOVEMBER       ;Nor Gate                  ;LINE 15
ROMEO =  POPPA   * /QUEBEC         ;first term of XOR         ;LINE 16
+ /POPPA   *  QUEBEC               ;second term of XOR        ;LINE 17
                                                              ;LINE 18
FUNCTION TABLE                                                ;LINE 19
ALPHA BRAVO CHARLIE DELTA ECHO FOX GOLF HOTEL INDIA JULIET KILO ;   20
LIMA MIKE NOVEMBER OSCAR POPPA QUEBEC ROMEO                   ;LINE 21
;                                     N                       ;LINE 22
;      C                              o                       ;LINE 23
;      h                   J          v        Q              ;LINE 24
;A B   a D           H   I u          e  O   P u  R           ;LINE 25
;l r   r e E      G  o   n l K  L   M m  s   o e  o           ;LINE 26
;p a   l l c    F o  t   d i i  i   i b  c   p b  m           ;LINE 27
;h v   i t h    o l  e   i e l  m   k e  a   p e  e           ;LINE 28
;a o   e a o    x f  l   a t o  a   e r  r   a c  o           ;LINE 29
---------                                                     ;LINE 30
 L H   X X  X   X X  X   X X X  X   X X  X   X X  X           ;LINE 31
 H L   X X  X   X X  X   X X X  X   X X  X   X X  X           ;LINE 32
 X X   L L  L   X X  X   X X X  X   X X  X   X X  X           ;LINE 33
 X X   L H  L   X X  X   X X X  X   X X  X   X X  X           ;LINE 34
 X X   H L  L   X X  X   X X X  X   X X  X   X X  X           ;LINE 35
 X X   H H  H   X X  X   X X X  X   X X  X   X X  X           ;LINE 36
 X X   X X  X   L L  L   X X X  X   X X  X   X X  X           ;LINE 37
 X X   X X  X   L H  H   X X X  X   X X  X   X X  X           ;LINE 38
 X X   X X  X   H L  H   X X X  X   X X  X   X X  X           ;LINE 39
 X X   X X  X   H H  H   X X X  X   X X  X   X X  X           ;LINE 40
 X X   X X  X   X X  X   L L L  H   X X  X   X X  X           ;LINE 41
 X X   X X  X   X X  X   L L H  H   X X  X   X X  X           ;LINE 42
 X X   X X  X   X X  X   L H L  H   X X  X   X X  X           ;LINE 43
 X X   X X  X   X X  X   H L L  H   X X  X   X X  X           ;LINE 44
 X X   X X  X   X X  X   H H H  L   X X  X   X X  X           ;LINE 45
 X X   X X  X   X X  X   X X X  X   L L  H   X X  X           ;LINE 46
 X X   X X  X   X X  X   X X X  X   L H  L   X X  X           ;LINE 47
 X X   X X  X   X X  X   X X X  X   H L  L   X X  X           ;LINE 48
 X X   X X  X   X X  X   X X X  X   H H  L   X X  X          ;LINE 49
 X X   X X  X   X X  X   X X X  X   X X  X   L L  L          ;LINE 50
 X X   X X  X   X X  X   X X X  X   X X  X   L H  H          ;LINE 51
 X X   X X  X   X X  X   X X X  X   X X  X   H L  H          ;LINE 52
 X X   X X  X   X X  X   X X X  X   X X  X   H H  L          ;LINE 53
----------                                                   ;LINE 54
                                                             ;LINE 55
DESCRIPTION                                                  ;LINE 56
This example is used to illustrate the use of                 LINE 57
 a 12H6 to implement                                         ;LINE 58
basic gate functions.                                        ;LINE 59
```

Send the source file to the compiler by typing the following:

```
C>GPC P0007 P7OUTS ;the following is the output of the compiler...
GTEK, INC. Pal Compiler Version 1.0 January 1, 1987
Copyright 1986, 1987 by GTEK, INC.
 All Rights Reserved, World Wide.

MICROSOFT QUICK-BASIC COMPILER, COPYRIGHT 1982-1986

The Input File Name is :P0007.PAL        ;Default Drive\Path
     The Output File Name is :P7OUTS.JED ;Def. Path, Spec. Filename
   The PAL Part File Name is :PAL12H6.PTM ;Default Drive\Path
Finding Output Pin In --BRAVO = /ALPHA             ;Compiling line
Finding Output Pin In --ECHO = CHARLIE * DELTA     ;compiling
Finding Output Pin In --HOTEL = FOX                ;compiling
Finding Output Pin In --LIMA = /INDIA              ;compiling
Finding Output Pin In --OSCAR = /MIKE * /NOVEMBER ;compiling
Finding Output Pin In --ROMEO = POPPA * /QUEBEC    ;compiling

PAL12H6 PAL DESIGN SPECIFICATION
GTEKP0007 WILLIAM C. EDMONDS 02/20/87

BASIC GATES
*MF*G0*F0*                          ;Set 12H6, Fill 0’S
L0000 111111101111111111111111*     ;Link Address/Data
L0096 010111111111111111111111*
L0144 111101111111111111111111*
L0168 111111110111111111111111*
L0192 111111111110101111111111*
L0240 111111111111110110111111*
L0264 111111111111111001111111*
L0288 111111111111111111111011*
L0312 111111111111111111111110*
L0336 111111111111111111101111*
V0001 XXXXXXXXXNXXXXXXXH0N*         ;Vector/number/data
V0002 XXXXXXXXXNXXXXXXXL1N*
V0003 00XXXXXXXNXXXXXXLXXN*
V0004 01XXXXXXXNXXXXXXLXXN*
V0005 10XXXXXXXNXXXXXXLXXN*
V0006 11XXXXXXXNXXXXXXHXXN*
V0007 XX00XXXXXNXXXXXLXXXN*
V0008 XX01XXXXXNXXXXXHXXXN*
```

```
V0009 XX10XXXXXNXXXXXHXXXN*
V0010 XX11XXXXXNXXXXXHXXXN*
V0011 XXXXXXXX0N00HXXXXXXN*
V0012 XXXXXXXX0N01HXXXXXXN*
V0013 XXXXXXXX0N10HXXXXXXN*
V0014 XXXXXXXX1N00HXXXXXXN*
V0015 XXXXXXXX1N11LXXXXXXN*
V0016 XXXX00XXXNXXXXHXXXXN*
V0017 XXXX01XXXNXXXXLXXXXN*
V0018 XXXX10XXXNXXXXLXXXXN*
V0019 XXXX11XXXNXXXXLXXXXN*
V0020 XXXXXX00XNXXXLXXXXXN*
V0021 XXXXXX01XNXXXHXXXXXN*
V0022 XXXXXX10XNXXXHXXXXXN*
V0023 XXXXXX11XNXXXLXXXXXN*

PAL12H6 PAL DESIGN SPECIFICATION        ;XPLOT of P0007.PAL
GTEKP0007 WILLIAM C. EDMONDS 02/20/87

BASIC GATES
                                   ;X Represents an intact fuse!
                                   ;- Represents a blown fuse
8 ---- ---X --   --   --   --   ---- ----
                                   ;Lines that are totally intact are
                                   ;not printed, they can never be
16 X-X- ---- --   --   --   --   ---- ----                    ;true


24 ---- X--- --   --   --   --   ---- ----
25 ---- ---- X-   --   --   --   ---- ----


32 ---- ---- --   -X   -X   --   ---- ----


40 ---- ---- --   --   --   X-   -X-- ----
41 ---- ---- --   --   --   -X   X--- ----


48 ---- ---- --   --   --   --   ---- -X--
49 ---- ---- --   --   --   --   ---- ---X
50 ---- ---- --   --   --   --   ---X ----


NUMBER OF BLOWN FUSES = 306

Done Compiling...
PAL12H6 PAL DESIGN SPECIFICATION
GTEKP0007 WILLIAM C. EDMONDS 02/20/87
BASIC GATES

GTEK, INC., BAY ST. LOUIS, MS
C>_                                ;returned to dos here...
```

The invocation of the above compiler results in two files on the default drive (C:P7outs.JED and C:P7outs.PLT). Remember that if you don't specify where to find the filename.PAL file or to put the filename.JED file, they will be put on the cur rent drive. The GPC program also expects that the files necessary to compile the part (.PTM files) are on the same drive and in the same directory as the GPC program.

7344LIMITEDWARRANTY

GTEK, INC., warrants to the original purchaser of this GTEK, INC., product that it is to be in good working order for a period of 90 days from the date of purchase from GTEK, INC., or an authorized GTEK, INC., dealer. Should this product, in GTEK, INC.'s opinion, malfunction during the warranty period, GTEK will, at its option, repair or replace it at no charge, provided that the product has not been subjected to misuse, abuse, or non-GTEK authorized alterations, modifications, and / or repairs.

Products requiring Limited Warranty service during the warranty period should be delivered to GTEK with proof of purchase. If the delivery is by mail, you agree to insure the product or assume the risk of loss or damage in transit. You also agree to prepay the shipping charges to GTEK. GTEK agrees to pay return freight (in the continental USA) by UPS surface on warranty work.

ALL EXPRESS AND IMPLIED WARRANTIES FOR THIS PRODUCT INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO THE ABOVE 90 DAY PERIOD. Some states do not allow limitations on how long an implied warranty lasts, so the above limitations may not apply to you.

UNDER NO CIRCUMSTANCES WILL GTEK, INC. BE LIABLE IN ANY WAY TO THE USER FOR DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF, OR INABILITY TO USE, SUCH PRODUCT. Some states do not allow the exclusion or limitation of incidental or consequential damages for consumer products, so the above limitations or exclusion may not apply to you.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH MAY VARY FROM STATE TO STATE.

The limited warranty applies to hardware products only.

PALX2LIMITEDWARRANTY

THIS PRODUCT IS NOT A CONSUMER PRODUCT WITHIN THE MEANING OF THE UNIFORM COMMERCIAL CODE AND APPLICABLE STATE LAW. THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (NOT GTEK, INC.) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

GTEK, Inc. does not warrant that the functions contained in the program will meet your requirements or that the operation of the program will be uninterrupted or error free. However, GTEK, Inc. warrants the diskette(s) on which the program is furnished, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from date of delivery to you as evidenced by a copy of your receipt.

Licensee herein acknowledges that the software licensed hereunder is of the class which inherently cannot be tested against all contingencies by Licensor. Licensee acknowledges Licensee's obligation to test all programs produced by the licensed software to determine suitability and correctness prior to use.

LIMITATIONS OF REMEDIES

GTEK, Inc.'s entire liability and your exclusive remedy shall be:

1. the replacement of any diskette(s) not meeting GTEK's "Limited Warranty" and which is returned to GTEK, Inc. with a copy of your receipt, or

2. if GTEK, Inc. or the dealer is unable to deliver a replacement diskette(s) which is free of defects in materials or workmanship, you may terminate this Agreement by returning the program and your money will be refunded.

IN NO EVENT WILL GTEK, INC. BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH PROGRAM EVEN IF GTEK, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

GENERAL

You may not substitute, assign or transfer the license or the program except as expressly provided in this Agreement. Any attempt otherwise to sublicense, as sign or transfer any of the rights, duties or obligations hereunder is void.

This Agreement will be governed by the laws of the State of Mississippi.

Should you have any questions concerning this Agreement, you may contact GTEK, Inc. by writing to GTEK, Inc. Sales and Service, P.O.Box 2310, Bay St. Louis, MS, 39521–2310.

GPC and PALX2 SOFTWARE LICENSE AGREEMENT

"This software is a proprietary product of GTEK, Inc. It is protected by copyright and trade secret laws. It is licensed (not sold) for use on a single micro-computer system, and is licensed only on the condition that you agree to this LICENSE AGREEMENT." GTEK, INC. provides this program and licenses its use worldwide. You assume responsibility for the use of this software to achieve your intended results, and for the installation, use and results obtained from the software.

LICENSE

The Licensee may:

a. use the program on a single machine;

b. copy the program into any machine readable or printed form for backup or modification purposes in support of your use of the program on the single machine;

c. modify the program and/or merge it into another program for your use on the single machine (Any portion of this program merged into another program will continue to be subject to the terms and conditions of this Agreement.): and,

d. transfer the program and license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the program, you must at the same time either transfer all copies whether in printed or machine-readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the program contained or merged into other programs.

You must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

YOU MAY NOT USE, COPY, MODIFY, OR TRANSFER THE PROGRAM, OR ANY COPY, MODIFICATION OR MERGED PORTION, IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE. IF YOU TRANSFER POSSESSION OF ANY COPY, MODIFICATION OR MERGED PORTION OF THE PROGRAM TO ANOTHER PARTY, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

TERM

The license is effective until terminated. You may terminate it at any other time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the program together with all copies, modifications and merged portions in any form.

SERVICE

For warranty service or non warranty service, contact GTEK, INC. at (601) 467–8048 to obtain an RMA (Return of Material Authorization number). We will need the serial number and date of purchase. Send the programmer, freight prepaid to:

GTEK, INC.
RMA Number #####
P. O. Box 2310
399 Highway 90
Bay St. Louis, Mississippi 39521–2310

Be sure to include the RMA <u>on second line of the address and inside with a letter of explanation</u> so we will know what to do with it. GTEK will pay return freight (UPS ground service within the Continental United States) on in-warranty service. Out of warranty service charges are determined on an hourly labor plus materials basis.

This appendix will attempt to carry you through the whole process of writing source code, compiling it, sending it to the 7344 and then programming a PAL. This example was tried a number of times with the typical set up of an AT computer and 7344 programmer. Our AT has a GTEK PCSS-8 Super Serial Card in it, but other than its mention during PINSTALL, it is transparent to you.

Basic Steps

1. Write your Boolean source code.
2. Compile it without errors with GPC.
3. From DOS, send it to the 7344 with PALX2.
4. Program blank PAL.

   Function test? (if vectors are present)
5. Secure PAL? (if necessary).

   Function test? (if vectors are present)
6. Program another PAL with the same data? Go to step 4
7. Program different PAL? Type control-C for DOS. Go to Step 1 or 3
8. Done. Type control-C to return to DOS.

Step 1. Create Boolean Equations File

Create your Boolean equations by typing in the following for practice, or use the source code provided on the disk (P0007.PAL). All source code files used by GPC must have the extension PAL (filename.PAL).

```
PAL12L10 M1 MMI PAL DESIGN SPECIFICATION                   LINE  1
GTEKP0007 WILLIAM C. EDMONDS 02/20/87                      LINE  2
BASIC GATES                                                LINE  3
GTEK, INC., BAY ST. LOUIS, MS                              LINE  4
Alp  Cha  Del  Fox  Gol  Ind  Jul  Kil  mik  Nov  Pop  Gnd  ;LINE  5
Que  Rom  Osc  Sie  Tan  Lim  Uni  Hot  Vic  Ech  Brv  Vcc  ;LINE  6
                                                          ;LINE  7
/Brv= Alp   ;INVERTER LINE 7 MUST BE BLANK                  LINE  8
    + ALP   ;SEE CHAPTER ON SEC. FUSE PGM                   LINE  9
                                                          ;LINE 10
/Ech= /Cha                        ;SIMULATE AN              LINE 11
    + /Del                        ;AND GATE                 LINE 12
                                                          ;LINE 13
/Hot= /FOX * /Gol                 ;OR GATE                  LINE 14
    + /FOX * /GOL                 ;FOR SEC. FUSE PGM        LINE 15
                                                          ;LINE 16
/Lim=  Ind * Jul * Kil            ;nand gate                LINE 17
    +  iND * jUL * kIL ;SEE SEC. FUSE PGM SIMPLE PARTS      LINE 18
                                                          ;LINE 19
/Osc= mik                         ;Nor gate first term      LINE 20
```

```
     + Nov                         ;Nor Gate second term       LINE 21
                                                              ;LINE 22
/Rom =  Pop *  Que                 ;first term of XOR          LINE 23
     + /Pop * /Que                 ;second term of XOR         LINE 24
                                                              ;LINE 25
/SIE = Cha * Del                   ;Nand gate                  LINE 26
     + Cha * Del                   ;See Security Fuse prog.    LINE 27
                                   ;PGM Simple parts           LINE 28
/TAN = CHA * DEL * ALP * FOX * GOL * MIK                      ;LINE 29
+  CHA * DEL * ALP * FOX * GOL * NOV                           LINE 30
                                                              ;LINE 31
/UNI =/Alp                         ;Buffer                     LINE 32
     +/ALp                         ;See CHP. for simple parts  LINE 33
                                                              ;LINE 34
/VIC= /FOX *  GOL                  ;Exclusive NOR              LINE 35
    +  FOX * /GOL                  ;See CHP. for simple parts ;LINE 36
                                                              ;LINE 37
                                   ;Some lines, like these,    LINE 38
                                   ;are optional...            LINE 39
FUNCTION TABLE                                                ;LINE 39
                                                              ;LINE 40
ALP BRV UNI CHA DEL ECH SIE fox GOL HOT VIC TAN IND JUL KIL   ;LINE 41
LIM MIK NOV OSC POP Que Rom                                   ;LINE 42
                                                              ;LINE 43
;                                         N                   ;LINE 44
;    U  C                                 o                   ;LINE 45
;    n  h      S         V      J         v       Q           ;LINE 46
;A B i  a D    i       H i T  I u         e  O  P u  R        ;LINE 47
;l r f  r e  E e    G  o c a  n l K  L  M m  s  o e  o        ;LINE 48
;p a o  l l  c r  F o  t t n  d i i  i  i b  c  p b  m        ;LINE 49
;h v r  i t  h r  o l  e o g  i e l  m  k e  a  p e  e        ;LINE 50
;a o m  e a  o a  x f  l r o  a t o  a  e r  r  a c  o        ;LINE 51
---------                                                     ;LINE 52
;A B U  C D  E S  F G  H V T  I J K  L  M N  O  P Q  R        ;LINE 53
 0 H L  X X  X X  X X  X X X  X X X  X  X X  X  X X  X        ;LINE 54
 1 L H  X X  X X  X X  X X X  X X X  X  X X  X  X X  X        ;LINE 55
 X X X  0 0  L H  X X  X X X  X X X  X  X X  X  X X  X        ;LINE 56
 X X X  0 1  L H  X X  X X X  X X X  X  X X  X  X X  X        ;LINE 57
 X X X  1 0  L H  X X  X X X  X X X  X  X X  X  X X  X        ;LINE 58
 X X X  1 1  H L  X X  X X X  X X X  X  X X  X  X X  X        ;LINE 59
 X X X  X X  X X  0 0  L H X  X X X  X  X X  X  X X  X        ;LINE 60
 X X X  X X  X X  0 1  H L X  X X X  X  X X  X  X X  X        ;LINE 61
 X X X  X X  X X  1 0  H L X  X X X  X  X X  X  X X  X        ;LINE 62
 X X X  X X  X X  1 1  H H X  X X X  X  X X  X  X X  X        ;LINE 63
 X X X  X X  X X  X X  X X X  0 0 0  H  X X  X  X X  X        ;LINE 64
 X X X  X X  X X  X X  X X X  0 0 1  H  X X  X  X X  X        ;LINE 66
 X X X  X X  X X  X X  X X X  0 1 0  H  X X  X  X X  X        ;LINE 67
 X X X  X X  X X  X X  X X X  1 0 0  H  X X  X  X X  X        ;LINE 68
 X X X  X X  X X  X X  X X X  1 1 1  L  X X  X  X X  X        ;LINE 69
 X X X  X X  X X  X X  X X X  X X X  X  0 0  H  X X  X        ;LINE 70
 X X X  X X  X X  X X  X X X  X X X  X  0 1  L  X X  X        ;LINE 71
 X X X  X X  X X  X X  X X X  X X X  X  1 0  L  X X  X        ;LINE 72
 X X X  X X  X X  X X  X X X  X X X  X  1 1  L  X X  X        ;LINE 73
 X X X  X X  X X  X X  X X X  X X X  X  X X  X  0 0  L        ;LINE 74
 X X X  X X  X X  X X  X X X  X X X  X  X X  X  0 1  H        ;LINE 75
 X X X  X X  X X  X X  X X X  X X X  X  X X  X  1 0  H        ;LINE 76
 X X X  X X  X X  X X  X X X  X X X  X  X X  X  1 1  L        ;LINE 77
;A B U  C D  E S  F G  H V T  I J K  L  M N  O  P Q  R        ;LINE 78
 0 X X  0 0  X X  0 0  X X H  X X X  X  0 X  X  X X  X        ;LINE 79
 0 X X  0 0  X X  0 0  X X H  X X X  X  1 X  X  X X  X        ;LINE 80
 0 X X  0 0  X X  0 1  X X H  X X X  X  0 X  X  X X  X        ;LINE 81
 0 X X  0 0  X X  1 0  X X H  X X X  X  0 X  X  X X  X        ;LINE 82
 0 X X  0 1  X X  0 0  X X H  X X X  X  0 X  X  X X  X        ;LINE 83
 0 X X  1 0  X X  0 0  X X H  X X X  X  0 X  X  X X  X        ;LINE 84
 1 X X  0 0  X X  0 0  X X H  X X X  X  0 X  X  X X  X        ;LINE 85
 1 X X  1 1  X X  1 1  X X L  X X X  X  1 0  X  X X  X        ;LINE 86
 1 X X  1 1  X X  1 1  X X L  X X X  X  0 1  X  X X  X        ;LINE 87
;A B U  C D  E S  F G  H V T  I J K  L  M N  O  P Q  R        ;LINE 88
----------                                                    ;LINE 89
                                                              ;LINE 90
DESCRIPTION                                                   ;LINE 91
This example is used to illustrate the use of a 12L10 to      ;LINE 92
implement basic gate functions.                               ;LINE 93
                ---++ file ends on above line ++---
```

Explanation:

Lines 1-4

Lines 1 through 4 in the above file contain information about why, where and when the file was created. The only critical part about it is the first line, which must contain the PAL part number left justified, along with the manufacturer of the part (if you don’t want it to default to MMI type parts).

Select the manufacturer menu number (eg. as displayed in the example menu under the menu command) and put it one space to the right of the PAL part number. Use the same syntax as you would from the programmer command line to indicate the manufacturer (like M4 for a TI 24 pin PAL).

Lines 5-6

Lines 5 and 6 contain the information about how the pins are represented for the Boolean equations. This is called the “Pin List”. A “Pin Name” is what you want to call the pin when you are describing how to use the pin in the equations.

You can duplicate pin names as long as they are not used anywhere in the equations, for example NC is a good one for a pin that is not used. If you are going to blow the security fuse, however, you MUST blow all unused fuses on unused terms in such a way as to not affect your equations or the operation of the part. This is discussed in the chapter on blowing security fuses. One method is demonstrated in several lines above. We’ll talk about them as we come to them. That is why several of the output pins have names, when we are not really using them. They don’t appear in the pin list for the Function Table, but you have to “use up” their OR terms to be able to secure the part properly.

If the pin list names are short enough, you can put them all on Line 5 alone. But Remember that you MUST have 1 blank (CR/LF) line between the pin list and the equations. That line should be Line 7.

Line 7

This line delineates the pin list from the Equations. It MUST always be blank (no spaces or other characters on it, only carriage return / line feed).

Line 8-38

These lines contain the boolean equations. A boolean as far as GPC is concerned is an (optionally a conditional expression) output pin name equals input pin expressions. If output pins or input pins are used on the wrong side of the equals operator, an error will be generated. Using an output pin on the input side generates an error, unless the output pin may also be an input, or if a registered part, /Q feedback. An input pin on the output side will generate an error unless it is used in a conditional expression, or it can also be used as an output.

Lines 10, 13, 16, 19, 22, 25, 28, 31, 34, 37 and 38 are used for clarity of the equations. They do not require a ";" (semi-colon) operator unless there is text on them. You can use them to separate your equations or put comments above each equation, etc. They do not affect the operation of the compiler.

Lines 9, 15, 18, 27 and 33 are used to duplicate the previous product term because we can't leave any unused terms for this output pin when we secure the part. See the chapter on blowing the security fuses in simple parts.

Line 39-89

These lines contain the information for the Function Table. The Function Table begins with a line that has the words "FUNCTION TABLE" left justified. Somewhere following that is a pin list to denote the order that the pin data will be put in the table, followed by a line of dashes (left justified, 1-80 dashes). After that there is the function table itself, composed of characters telling GPC to set the pin high or low, or the state of the pin itself. See the chapter on function tables for a better explanation of a function table.

Line 90-

These lines generally contain a description about the previous equations typically explaining the operation, or anything else the operator cares to talk about, such as special handling of the PAL and so forth.

2. Compile it without errors with GPC.

Do this with the following command:

C> **GPC [d:\path\filename[.PAL] d:\path\filename[.JED]]↵**

The square brackets [ ] means that the information inside those brackets is optional, so that you can have maximum flexibility for where your source code is located and where you want to put the JEDEC file. Notice that the extensions for the filenames are optional too. GPC always supplies the extension or tells you when you are wrong. In our example there are several ways that you can compile P0007.PAL:

A. With all files on the same drive and in the same directory:

```
C>GPC P0007↵
```

B. With source file on another disk in a directory called PAL, and you want the JEDEC file to be on same disk:

```
C>GPC A:\pal\P0007↵
```

C. With Source file on another disk in a directory call SOURCE and you want to put the JEDEC file called BURNME.PAL on another disk in a directory called JED:

```
C>GPC A:\SOURCE\P0007 B:\JED\BURNME↵
```

Example session with GPC: (box lines do not show...)

```
             GTEK, INC. Pal Compiler Version 1.0 January 1, 1987
                Copyright 1986, 1987 by GTEK, INC.
                 All Rights Reserved, World Wide.

             MICROSOFT QUICK-BASIC COMPILER, COPYRIGHT 1982-1986
   The Input File Name is :P0007.PAL
  The Output File Name is :P0007.JED
The PAL Part File Name is :PAL12L10.PTM
Finding Output Pin In --/BRV   = ALP
Finding Output Pin In --/ECH   = /CHA
Finding Output Pin In --/HOT   = /FOX * /GOL
Finding Output Pin In --/LIM   =  IND * JUL * KIL
Finding Output Pin In --/OSC   = MIK
Finding Output Pin In --/ROM   =  POP *  QUE
Finding Output Pin In --/SIE   = /FOX * /GOL
Finding Output Pin In --/UNI   = /FOX * /GOL
Finding Output Pin In --/VIC   = /FOX * /GOL

PAL12L10 PAL DESIGN SPECIFICATION
GTEKP0007 WILLIAM C. EDMONDS 02/20/87

BASIC GATES
*MP*G0*F0*
L0000 110111111111111111111111*
L0024 110111111111111111111111*
```

```
L0048 101111111111111111111111*
L0072 111110111111111111111111*
L0096 111111101011111111111111*
L0120 111111101011111111111111*
L0144 111111101011111111111111*
L0168 111111101011111111111111*
L0192 111111101011111111111111*
L0216 111111101011111111111111*
L0240 111111111101010111111111*
L0264 111111111101010111111111*
L0336 111111101011111111111111*
L0360 111111101011111111111111*
L0384 111111111111111101111111*
L0408 111111111111111111011111*
L0432 111111111111111111110101*
L0456 111111111111111111111010*
V0001 0XXXXXXXXXXNXXXXXXXXXXHN*
V0002 1XXXXXXXXXXNXXXXXXXXXXLN*
V0003 X00XXXXXXXXNXXXXXXXXXLXN*
V0004 X01XXXXXXXXNXXXXXXXXXLXN*
V0005 X10XXXXXXXXNXXXXXXXXXLXN*
V0006 X11XXXXXXXXNXXXXXXXXXHXN*
V0007 XXX00XXXXXXNXXXXXXXLXXXN*
V0008 XXX01XXXXXXNXXXXXXXHXXXN*
V0009 XXX10XXXXXXNXXXXXXXHXXXN*
V0010 XXX11XXXXXXNXXXXXXXHXXXN*
V0011 XXXXX000XXXNXXXXXHXXXXXN*
V0012 XXXXX001XXXNXXXXXHXXXXXN*
V0013 XXXXX010XXXNXXXXXHXXXXXN*
V0014 XXXXX100XXXNXXXXXHXXXXXN*
V0015 XXXXX111XXXNXXXXXLXXXXXN*
V0016 XXXXXXXX00XNXXHXXXXXXXXN*
V0017 XXXXXXXX01XNXXLXXXXXXXXN*
V0018 XXXXXXXX10XNXXLXXXXXXXXN*
V0019 XXXXXXXX11XNXXLXXXXXXXXN*
V0020 XXXXXXXXXX0N0LXXXXXXXXXN*
V0021 XXXXXXXXXX0N1HXXXXXXXXXN*
V0022 XXXXXXXXXX1N0HXXXXXXXXXN*
V0023 XXXXXXXXXX1N1LXXXXXXXXXN*

PAL12L10 PAL DESIGN SPECIFICATION
GTEKP0007 WILLIAM C. EDMONDS 02/20/87

BASIC GATES

 0 --X- --   --   --   --   --   --   --   --   ----
 1 --X- --   --   --   --   --   --   --   --   ----


 8 -X-- --   --   --   --   --   --   --   --   ----
 9 ---- -X   --   --   --   --   --   --   --   ----
```

```
16 ---- --   -X   -X   --   --   --   --   --   ----
17 ---- --   -X   -X   --   --   --   --   --   ----


24 ---- --   -X   -X   --   --   --   --   --   ----
25 ---- --   -X   -X   --   --   --   --   --   ----


32 ---- --   -X   -X   --   --   --   --   --   ----
33 ---- --   -X   -X   --   --   --   --   --   ----


40 ---- --   --   --   X-   X-   X-   --   --   ----
41 ---- --   --   --   X-   X-   X-   --   --   ----



56 ---- --   -X   -X   --   --   --   --   --   ----
57 ---- --   -X   -X   --   --   --   --   --   ----


64 ---- --   --   --   --   --   --   X-   --   ----
65 ---- --   --   --   --   --   --   --   X-   ----


72 ---- --   --   --   --   --   --   --   --   X-X-
73 ---- --   --   --   --   --   --   --   --   -X-X

NUMBER OF BLOWN FUSES = 688

Done Compiling...
PAL12L10 PAL DESIGN SPECIFICATION
GTEKP0007 WILLIAM C. EDMONDS 02/20/87

BASIC GATES

GTEK, INC., BAY ST. LOUIS, MS

C:\>_
```

Assuming you used the defaults and a program name of P0007, AND you did not have any errors while compiling, at this point you now have a file on your disk called P0007.JED. You are now ready to transfer this JEDEC file to the 7344.

3. Send JEDEC file to 7344 with PALX2.

You can now send the JEDEC file we just created to the 7344 using PALX2. The syntax for doing this is : C>PALX2 filename.JED↵. You have to be ex plicit about the extension. PALX2 does not do any checking to see what kind of file it is about to send, so it looks at the extension to tell what kind of command to give the 7344 to transfer. TB is used for JEDEC transfers and TC is used for AHS, so use .JED for JEDEC and .AHS for Ascii-Space-Hex. Of course if all you want to do is communicate with the programmer, don't give it a filename.ext.

With PALX2 you may specify a drive number, but you cannot specify a path. We recommend that you install PALX2 in a directory that you have set a path to so you can execute PALX2 from any drive or directory on your computer. Log on to the drive and/or directory where your JEDEC file resides and then you can perform one of the following.

A. Transfer with programs on default drive\path:

```
C>PALX2 P0007.JED↵
```

B. Transfer with programs on another drive in a directory called JED:

```
C>CD B:\JED↵

C>PALX2 B:P0007.JED↵
```

Example session with 7344. Comments are made to the side with a semicolon:

```
C>PALX2 P0007.JED↵                ;invoke PALX2

Pal Programmer Com. Package Version 3.03
Copyright 1983, 1986  GTEK, INC.
I/O Hardware Driver Vers 1.04 - IBM PC/XT/AT
Serial port  - COM22:,  2400 bps
Printer port - LPT1:

Initializing....
Z
GTEK Corp
Model 7344 V1.02
Copyright 1984, 1986
MMI-XXXX>TB
PAL12L10 M1 MMI PAL DESIGN SPECIFICATION LINE  1
GTEKP0007 WILLIAM C. EDMONDS 02/20/87 LINE  2

BASIC GATES LINE  3
*MP*G0*F0*
L0000 110111111111111111111111*
L0024 110111111111111111111111*
L0048 101111111111111111111111*
L0072 111110111111111111111111*
L0096 111111100111111111111111*
L0120 111111011011111111111111*
L0144 111111101011111111111111*
L0168 111111101011111111111111*
L0192 111011111111111111111111*
L0216 111011111111111111111111*
L0240 111111111101010111111111*
```

```
L0264 111111111101010111111111*
L0288 010101010111111101111111*
L0312 010101010111111111011111*
L0336 011101111111111111111111*
L0360 011101111111111111111111*
L0384 111111111111111101111111*
L0408 111111111111111111011111*
L0432 111111111111111111110101*
L0456 111111111111111111111010*
V0001 0XXXXXXXXXXNXXXXXXLXXXHN*
V0002 1XXXXXXXXXXNXXXXXXHXXXLN*
V0003 X00XXXXXXXXNXXXHXXXXXLXN*
V0004 X01XXXXXXXXNXXXHXXXXXLXN*
V0005 X10XXXXXXXXNXXXHXXXXXLXN*
V0006 X11XXXXXXXXNXXXLXXXXXHXN*
V0007 XXX00XXXXXXNXXXXXXXLHXXN*
V0008 XXX01XXXXXXNXXXXXXXHLXXN*
V0009 XXX10XXXXXXNXXXXXXXHLXXN*
V0010 XXX11XXXXXXNXXXXXXXHHXXN*
V0011 XXXXX000XXXNXXXXXHXXXXXN*
V0012 XXXXX001XXXNXXXXXHXXXXXN*
V0013 XXXXX010XXXNXXXXXHXXXXXN*
V0014 XXXXX100XXXNXXXXXHXXXXXN*
V0015 XXXXX111XXXNXXXXXLXXXXXN*
V0016 XXXXXXXX00XNXXHXXXXXXXXN*
V0017 XXXXXXXX01XNXXLXXXXXXXXN*
V0018 XXXXXXXX10XNXXLXXXXXXXXN*
V0019 XXXXXXXX11XNXXLXXXXXXXXN*
V0020 XXXXXXXXXX0N0LXXXXXXXXXN*
V0021 XXXXXXXXXX0N1HXXXXXXXXXN*
V0022 XXXXXXXXXX1N0HXXXXXXXXXN*
V0023 XXXXXXXXXX1N1LXXXXXXXXXN*
V0024 00000XXX0XXNXXXXHXXXXXXN*
V0025 00000XXX1XXNXXXXHXXXXXXN*
V0026 00001XXX0XXNXXXXHXXXXXXN*
V0027 00010XXX0XXNXXXXHXXXXXXN*
V0028 00100XXX0XXNXXXXHXXXXXXN*
V0029 01000XXX0XXNXXXXHXXXXXXN*
V0030 10000XXX0XXNXXXXHXXXXXXN*
V0031 11111XXX10XNXXXXLXXXXXXN*
V0032 11111XXX01XNXXXXLXXXXXXN*    ;eof here. file is through
                                   ;transferring.
Check Sum = 10                     ;checksum is for RAM buffer
MMI-12L10>_
MMI-12L10>Blank? (Y/N) Y           ;beeps
Not Blank                          ;OOOPS! we forgot to put the part
                                   ; in the socket!
MMI-12L10>Blank? (Y/N) Y           ;beeps... we already inserted MMI
                                   ; PAL12L10
Check Sum = 10                     ;Checksum of RAM buf, not the part!
MMI-12L10>Program? (Y/N) Y         ;beeps...
Check Sum = 10                     ;No error message... programmed ok.
MMI-12L10>Verify? (Y/N) Y          ;Verify for the heck of it.
```

```
Check Sum = 10                     ;checksum says that it’s ok.
MMI-12L10>Function testing         ;test the part
MMI-12L10>_                        ;no complaints, test is good!
MMI-12L10>Secure? (Y/N) Y          ;Secure the part
MMI-12L10>_                        ;part is secured, no complaints
```

At this point, you have gone through a complete cycle. If you want to program more parts, insert a blank part and start again at blank part.

Function Test Description

The Function Test on the 7344 uses the test vectors you have uploaded to the programmer to functionally test the part. It uses the logic states stated in the function table (X, 0, 1, C, K or P for inputs) to set the input pins, and the logic state for the output pins (Z, L, H, or X) to test the outputs.

Function Table Description

The Function Table is what you state in the source code to enable GPC to generate a test vector for the JEDEC file. It is composed of the same letters used for input and output pins (X, 0, 1, C, K, P for inputs and Z, L, H or X for outputs).

Functional Testing

When you use the "F" command on the 7344, it uses the test vectors that you have uploaded to it to functionally test the part in the selected socket. This is done by setting the input pins as specified (and clocked if necessary) and then reading the levels set on the output pins. If there is a level error on the output, it is reported (or complained about) and then the next test is performed. An error will be generated for each output pin in error so you may have as many as 10 output level errors complained about per vector. Every time a complaint is made, the test vector is also displayed.

If an input pin has an "X" in place, then it is assumed that input line is meant to be low. The 7344 does not test with a high in place of an "X" for an input pin. Output pins that have an "X" in place are not tested. If you wish, you may more thoroughly test the part by specifying more "1"'s and "0"'s instead of "X"'s for inputs and "H"'s and "L"'s instead of "X"'s for out puts.

See the previous appendix for an example of function table syntax.

FUNCTION TABLE example. Part is registered, but several different lines in this example have no real meaning as far as THIS part is concerned, because for in stance, you won't have a clock line that can be both high and low true... It will not clock at all if that is the case!

Registered logic feedback input lines depend on the previous state of the flip-flop for the equations that follow, so this example is not good for that demonstration, just keep that in mind when you are writing your own. NOTE 7344 does not support clocking with registered parts!

```
Function  Table
CK pin1 pin2 pin3 OC                ;WHERE PIN1 AND PIN2 ARE INPUTS,
                                    ;PIN3 IS OUTPUT OC IS
                                    ; TRISTATE AND CK IS CLOCK
; P  P  P  P
; I  I  I  I
; N  N  N  N  O  C
; 1  2  3  4  C  K
-----------------------             ;small example of function table
  X  X  X  L  0  P                  ;PRESET THE OUTPUT PIN TO BE LOW
                                    ;(MEANS OUTPUT REGISTER Q IS HIGH,
                                    ; /Q (FEEDBACK) IS LOW)
  X  X  X  Z  1  C                  ;DON'T CARE ON INPUTS, OUTPUT HI-Z
```

```
                                 ;BECAUSE OC IS HIGH
                                 ; C MEANS CK LINE WENT LOW FOR A
                                 ;LOW TRUE CLOCK.
X  X  X  Z  1  K                 ;DON’T CARE ON INPUTS, OUTPUT HI-Z
                                 ;BECAUSE OC IS HIGH
                                 ;K MEANS CK LINE WENT HIGH FOR A
                                 ;HIGH TRUE CLOCK.
0  0  0  L  0  C                 ;ALL INPUTS LOW, OUTPUT LOW AFTER A
                                 ; CLOCK OCCURS
0  0  0  H  0  C                 ;AFTER CLOCK STATE CHANGES BECAUSE
                                 ; OF EQUATIONS
0  0  0  Z  1  C                 ;OUTPUT HI-Z BECAUSE OC PIN WENT
                                 ;FALSE (STATE OF FF STILL CHANGES,
                                 ; HOWEVER
X  X  X  L  0  1                 ;NO CLOCK THIS TIME, OUTPUT IS IN
                                 ; CORRECT STATE
X  X  X  H  0  0                 ;CLOCK CHANGED TRUE (C) FROM
                                 ;PREVIOUS EQUATION
L  L  L  L  0  C                 ;YOU MAY USE L AND H FOR INPUTS,
                                 ;BUT NO 0’S AND 1’S
                                 ; FOR OUTPUTS!
```