

USER'S MANUAL

For the

*Data I/O Corporation
Chipwriter Device Programmers*



March 2001 - PC Based Programmer Manual Ver.3.0

Data I/O Corporation has made every attempt to ensure that the information in this document is accurate and complete. Data I/O Corporation assumes no liability for errors, or for any incidental, consequential, indirect or special damages, including, without limitation, loss of use, loss or alteration of data, delays or lost profits or savings, arising from the use of this document, or the product which it accompanies.

No device of this document may be reproduced or transmitted in any form or by means, electronic or mechanical, for any purpose without written permission from Data I/O Corporation

Data I/O Corporation
10525 Willows Road NE
P.O. Box 97046
Redmond, WA 98073-9746

Acknowledgments:

- Data I/O Corporation and the “IT” logo are registered trademarks of Data I/O Corporation
- Data I/O Corporation acknowledges the trademarks of other organizations for their respective products or services mentioned in this document.

Limitation of Liability:

Data I/O only use the specified programming algorithms provided by the specific device manufacturers. Whilst every effort has been made to ensure that programming algorithms are correct, all said algorithms are not guaranteed by the device manufacturer and therefore cannot be guaranteed by Data I/O. Users are advised to ensure that devices work correctly in circuit before programming large quantities.

In no event shall Data I/O be liable to the user or any third party for incidental, special or consequential damages, including without limitation, loss of use, loss or alteration of data, lost profits or lost savings, arising out of the use or inability to use Data I/O products or software or any combination or portion thereof.

Contents

| | |
|---|-----------|
| Contents..... | 3 |
| 1. Introduction..... | 6 |
| 1.1 About this manual..... | 6 |
| 1.2 Conventions..... | 6 |
| 1.3 Low Voltage Features..... | 7 |
| 2. Installing the Software | 8 |
| 2.1 Latest Software..... | 8 |
| 2.2 Software Installation..... | 8 |
| 2.3 Windows Software Installation | 8 |
| 2.4 DOS Software Installation..... | 9 |
| 3. Connecting the Programmer | 10 |
| 3.1 Computer Requirements..... | 10 |
| 3.2 Connection to the PC..... | 10 |
| 3.3 Running the SELFTEST Program..... | 10 |
| 3.4 Operating from Batteries | 11 |
| 3.4.1 Recharging the Batteries..... | 11 |
| 3.5 Programming surface-mount and other non-DIP devices..... | 12 |
| 3.5.1 Adapter usage..... | 12 |
| 4. Getting Started..... | 13 |
| 4.1 Before You Start..... | 13 |
| 4.2 Quickstart tutorials | 14 |
| TUTORIAL 1: ChipWin – Programming a Microcontroller from a master device | 14 |
| TUTORIAL 2: DOS - Programming an EPROM or Microcontroller from a file on Disk | 14 |
| TUTORIAL 3: Programming an EPROM or Microcontroller from a file on disk..... | 15 |
| TUTORIAL 4: Programming a PAL or GAL | 15 |
| 5. Using the DOS Programmer Software..... | 16 |
| 5.1 Basic Software Configuration | 16 |
| 5.2 Screen Layout..... | 16 |
| 5.3 General Menu Selection | 17 |
| 5.4 Dialogue Boxes | 17 |
| 5.5 Changing the Settings..... | 17 |
| 5.6 Using a mouse | 17 |
| 5.6.1 Left Hand Mouse Button (LMB)..... | 18 |
| 5.6.2 Right Hand Mouse Button (RMB) | 18 |
| 5.6.3 Middle Mouse Button (MMB) | 18 |
| 5.6.4 Disabling the Mouse..... | 18 |
| 5.7 Edit and View modes and Scroll Bars | 18 |
| 5.8 Running the Software under Windows..... | 18 |
| 5.9 Using the Programmer Software over a Network..... | 18 |
| 5.10 Saving software settings | 19 |
| 6. Memory and Microcontroller Programming..... | 20 |
| 6.1 Device Selection..... | 20 |
| 6.1.1 To select a device | 20 |
| 6.1.2 Changing the device selected | 20 |
| 6.1.3 Which part? | 21 |

| | |
|--|-----------|
| 6.1.4 Low Voltage Parts | 21 |
| 6.2 Opening data files | 21 |
| 6.2.1 Setting the data memory buffer size | 21 |
| 6.2.2 Opening A File | 21 |
| 6.2.3 File Format | 22 |
| 6.2.4 Handling large files and devices..... | 22 |
| 6.2.5 File Checksum..... | 23 |
| 6.2.6 Wild card Loading..... | 23 |
| 6.2.7 Saving a file to disk | 23 |
| 6.2.8 Saving a device configuration | 24 |
| 6.3. Using the Editing Facilities | 25 |
| 6.3.1 Entering EDIT mode | 25 |
| 6.3.2 Viewing without making changes | 25 |
| 6.3.3 Quickly Moving Around the Buffer..... | 25 |
| 6.3.4 Quick buffer editing commands | 25 |
| 6.3.5 Buffer checksum..... | 26 |
| 6.3.6 Buffer Display Modes | 26 |
| 6.3.7 Printing the Contents of the Buffer..... | 26 |
| 6.4 Reading and Checking Devices | 26 |
| 6.4.1 Reading a Device..... | 26 |
| 6.4.2 Verifying the contents of a chip against a file..... | 27 |
| 6.4.3 Verifying the device type | 27 |
| 6.4.4 Checking if a device is blank..... | 28 |
| 6.5 Programming Devices | 28 |
| 6.5.1 Programming Options | 28 |
| 6.5.2 Programming Operation | 29 |
| 6.5.3 Overprogramming devices | 30 |
| 6.6 Microcontroller Programming | 30 |
| 6.6.1 Address Locations | 30 |
| 6.6.2 Security Features | 30 |
| 6.6.3 Device- Specific Features | 31 |
| 6.7 Erasing devices | 31 |
| 7. Programmable Logic | 31 |
| 7.1 Device Selection | 32 |
| 7.1.1 To select a device | 32 |
| 7.1.2 Changing the device selected | 32 |
| 7.1.3 Which device? | 32 |
| 7.1.4 Low Voltage devices | 32 |
| 7.2 Opening data files | 33 |
| 7.2.1 Opening a file | 33 |
| 7.2.2 File format | 33 |
| 7.2.3 Wild card loading | 33 |
| 7.2.4 Saving a file to disk | 33 |
| 7.2.5 Saving configuration | 33 |
| 7.3 Reading and Checking Devices | 34 |
| 7.3.1 Reading a Device..... | 34 |
| 7.3.2 Verifying the contents of a device against a file..... | 34 |
| 7.3.3 Checking if a device is blank..... | 34 |
| 7.4 Programming Logic Devices | 35 |
| 7.4.1 Programming operation | 35 |
| 7.4.2 Overprogramming devices | 35 |
| 7.4.3 Single key programming | 35 |
| 7.5 Erasing devices | 35 |
| 7.6 Device Security Features | 36 |
| 7.7 Using the editing facilities | 36 |
| 7.7.1 Editing the AND/OR Array..... | 36 |
| 7.7.2 Moving around the buffer..... | 36 |
| 7.7.3 User's Electronic Signature (UES) | 37 |

| | |
|---|-----------|
| 7.7.4 Viewing the Fusemap | 37 |
| 7.7.5 Buffer checksum..... | 37 |
| 7.8 Converting PAL Files for GAL Devices | 37 |
| 7.9 Test Vector Editing and Operation | 37 |
| 7.10 Programming Altera MAX devices | 38 |
| 7.10.1 Using POF files | 38 |
| 7.10.2 Using JAM files..... | 38 |
| 8. Command Line Batch Software | 40 |
| 9. Using CWCTEST.EXE | 42 |
| 9.1 Testing a device | 42 |
| 9.2 Identifying a Device | 42 |
| 9.3 Adding Devices to the User Library | 42 |
| 8.4 CWCTEST Restrictions | 42 |
| 9.5 CWCTEST Example | 43 |
| Appendix A: Shorthand Keystrokes | 44 |
| Appendix B: Technical Support & Contact Information..... | 45 |
| Sending E-mail Technical Support Request | 45 |
| Using the Data I/O Web Page | 45 |
| Appendix C: Frequently Asked Questions | 46 |
| Guarantee Conditions | 47 |

1. Introduction

Congratulations on choosing a Data I/O programmer. We are sure that you will find the programmer software very easy to use and full of features to support all of your device programming requirements, whether you are using it for development, field service, or low-medium scale production. With your new programmer you will be able to

- ✓ Read data from programmed devices
- ✓ Load data in from disk
- ✓ Edit data
- ✓ Program a device
- ✓ Verify the contents of a device

There are many other features of your programmer that will allow you to achieve the results you need. This manual explains these features and how to use them.

1.1 About this manual

This manual provides the instructions for using The ChipWriter programmers available from Data I/O. These are the ChipWriter, ChipWriter Portable and ChipWriter Gang. All of the programmers use exactly the same software, so all the features are common across the range. Intelligent interrogation of programmer firmware results in only the devices supported by the particular programmer being available.

An installation guide is provided with the programmer which allows you to get the programmer installed successfully. Also included with the guide are a number of QuickStart tutorials, which give a number of common usage examples.

For Windows users, there is comprehensive on-line Help within the ChipWin software. This manual is therefore primarily concerned with the DOS software. However, there are some features relevant to both DOS and Windows users. Sections 1 – 4 cover all aspects of the programmer set-up and installation, and should be read by all users.

It is recommended that you read through the relevant sections of this manual before using the programmer, and in case of any difficulties encountered during use.

1.2 Conventions

Throughout this manual, certain conventions will be followed in typefaces in order to make instructions clear.

e.g. In the line

Type EPROM <ENTER>

EPROM: this font is used throughout the manual to indicate something that should be typed into the computer

<ENTER>: Any items encased in <> indicate the name of a key to press; in this case the ENTER, or RETURN key.

Menu selections are indicated by words separated by forward slashes, e.g. FILE/LOAD indicates select the menu option FILE, followed by the sub-menu option LOAD. Messages displayed by the programmer software are indicated in *Italics*, for example:

DEVICE PROGRAMD AND VERIFIED

1.3 Low Voltage Features

All the ChipWriter programmers support Low-Voltage devices, including 1.8V, 2.7V and 3.3V parts, as well as standard 5V devices. They are capable of programming and verifying devices down to 1.8 Volts. When using low voltage devices, the programming algorithm in the system will automatically be set to the correct voltages as specified by the manufacturer's programming algorithm.

2. Installing the Software

2.1 Latest Software

Both device programming algorithms and system operating software for this line of programmers is distributed via the BBS and Internet WEB page. Since software algorithms are added on a regular basis, it is possible that the software provided with your new system has been superseded by a newer revision.

To insure that you have all the very latest device programming algorithms and system software installed on your programmer, it is recommended that you contact the Data I/O BBS or WEB page.

Contact information is given in Appendix B of this manual.

2.2 Software Installation

Before you try connecting the programmer, please ensure that your PC has the following minimum requirements before you install the software. (Note that ChipWin has the same minimum requirements as the Windows 95/98™ operating system).

- IBM Compatible PC
- 8 MB RAM
- Approx. 6 MB Free hard disk space
- DOS version 3.0 or higher (Windows 95/98™ for ChipWin software)
- Parallel port - ECP mode selected in PC BIOS
- Standard serial/PS2 mouse

Software is provided for both DOS and Windows 95/98™ based systems on the ChipWriter CD. Contained is the set-up program for Windows 95/98™, DOS installation program, Emulator options software, Chiptest software and Selftest diagnostics, plus command line batch utilities for DOS.

2.3 Windows Software Installation

Follow these steps to install ChipWin onto your PC:

Running Set-up in Windows 95/98™ installs both Windows 95/98™ and DOS software. Insert ChipWriter CD into the CD ROM drive

- Click on Start in Windows 95/98™ Desktop
- Select Run
- Type "E:\set-up" in the dialog box
- or click on Browse. in the run window to manually select the set-up file for ChipWin on your CD ROM
- Follow the on screen set-up instructions
- The software will be installed to C:\CHIPWRITER by default on your hard drive, and create the Data I/O group on the Start menu on the Windows 95/98™ Desktop
- The destination directory and Program group can be changed during the Set-up process
- Run ChipWin for all programmer operations, including programming memory, microcontroller and logic devices, initializing the emulator options and the built in ChipTester
- Select Programmer/Diagnostics/Selftest from the menu bar to check the functionality of the programmer

2.4 DOS Software Installation

Follow these steps to install DOS versions of the software onto your PC:

Running the Install program in DOS installs only the DOS software programs and utilities. Insert the ChipWriter CD ROM into the CD ROM drive

- Select drive E and type `INSTALL C:\CHIPWRITER` at the prompt in the DOS window
- This installs the software to `C:\CHIPWRITER` by default on your hard drive.
- The destination directory can be changed simply by selecting a different directory name on the `INSTALL` command line.
- Once installed, run the diagnostic utility by typing `SELFTEST <ENTER>` in the `CHIPWRITER` sub-directory. If you have the Emulator card option installed, this can be tested by typing `CWEMUL <ENTER>`.
- Use `CWEPROM` for programming Memory and Microcontroller devices, `CWPAL` for logic devices, `CWEMUL` for the built in Emulator, `CWGANG` for the Gang programmer and `CWCTEST` for the ChipTester.

3. Connecting the Programmer

3.1 Computer Requirements

The programmer will operate with any IBM compatible PC with an ECP Parallel Port Interface. This means that it is very easy to move it between PCs. Users who prefer to have an internal card or who do not wish to use their existing parallel port can fit an additional parallel card.

The software is compatible with MS-DOS v.3.0 or later, and with MS-Windows, v.3.1, Windows 9x, Windows NT, Windows 2000 and Windows ME.

3.2 Connection to the PC

STEP 1 - Connect one end of the parallel cable to the programmer. The other end of the cable should connect to the PC. The cable will only fit the parallel port. Tighten the holding screws thus ensuring that the cable will not cause programming problems.

NOTE: *The programmer does not have a power on switch. It will power up when a signal is received from the computer, and automatically power down when not in use.*

STEP 2 - Connect the power supply provided to the programmer and plug into any mains socket. Alternatively, fit 8 alkaline AA batteries.

USE ONLY THE POWER SUPPLY ADAPTER PROVIDED.

Replacements and adapters for different countries (UK/Euro/Japan) can be supplied by Data I/O or their distributors.

STEP 3 - Switch on the PC.

STEP 4 - Run the SELFTEST program

3.3 Running the SELFTEST Program

Your programmer has been supplied with self-testing software which can check that the programmer has been installed correctly and can be used for fault diagnosis. Before using the programmer for the first time, whenever moving it to a new PC or if a fault is suspected, run the **SELFTEST** program. This has two purposes:

- It confirms which parallel port is being used by the programmer.
- It checks almost ALL of the programmer hardware.

WARNING :

ALWAYS ENSURE THAT THERE IS NO CHIP IN THE PROGRAMMER BEFORE RUNNING SELFTEST, AS LEAVING A DEVICE IN THE SOCKET COULD DAMAGE THE CHIP AND GIVE FALSE RESULTS.

To run the program, connect the programmer to the PC and either mains or battery power, remove any chips and type:

SELFTEST <CR> at the DOS prompt.

The SELFTEST will report the programmer's Firmware version number and run a communications test. It will then run through a series of hardware checks and will report any faults found. The Selftest program will report on which port the programmer is connected to.

If an error is reported it is possible to obtain a printout by using:

SELFTEST >PRN

where PRN is the name of the resident printer device (i.e.. LPT1).

or
followed by SELFTEST >ERROR.LST
 PRINT ERROR.LST

See Appendix B for Technical Support information

3.4 Operating from Batteries

The ChipWriter and the ChipWriter Portable programmers can be operated from mains or batteries. To operate from batteries use 8 alkaline AA type cells. The battery cover is held in place with screws. When replacing the cover, take care to use only the screws provided.

WARNING : USE ONLY THE SCREWS PROVIDED TO FASTEN THE BATTERY COVER, & DO NOT OVER TIGHTEN THE SCREWS.

NOTE: Certain Bipolar devices require high currents in very short bursts and therefore some batteries may not always be able to handle these devices.

3.4.1 Recharging the Batteries

The programmer can be used as a battery charger. In order to start recharging the batteries the programmer needs to be connected to a PC and connected to the mains using the mains adapter provided.

Please Note: This utility is ONLY available in DOS

To charge the batteries, the command is: BCHARGER

usage: BCHARGER [port] CHARGEMODE [REPORTMODE]

where:

port parallel port to use (default = 1)
CHARGEMODE TRICKLE/SLOW/FAST (default=slow)
REPORTMODE START/REPORT (default=both)
START starts the charging and does immediate return
REPORT reports current stage of charging

When the batteries are charging the 'BUSY' LED on the programmer will flash red and yellow. Once charging is complete this LED will switch off. Once recharging has commenced the programmer can be disconnected from the PC. Charging will stop once the batteries have been fully charged. If the PC is switched off battery charging will be interrupted for a few seconds. Recharging will resume automatically.

WARNING : DO NOT ATTEMPT TO CHARGE NON-RECHARGEABLE BATTERIES

3.5 Programming surface-mount and other non-DIP devices

Devices in packaging other than DIP, and devices over 40-pins (48-pins on ChipWriter programmers) require socket adapters. A range of adapters is available from Data I/O for most common devices. If you are unsure as to which adapter to use, please contact Data I/O or your distributor for advice.

Adapters from other sources, which are simple pin conversions, can usually be used successfully. However, for devices over 40-pins (48-pins), pin mappings are not usually standard, and it is therefore necessary to use an adapter specifically designed for the programmer in use. Where devices require non-standard adapters to be supported, these are available from Data I/O.

All adapters available from Data I/O are standard across the programmer range.

3.5.1 Adapter usage

Insert the adapter into the ZIF socket of the programmer with the bottom end of the adapter in the bottom pins of the socket. The text on the adapter boards should be the right way up. Lock the ZIF socket handle in place. For PLCC adapters, the notched corner of the adapter socket will usually be in the top left hand corner. Devices are placed in the socket in the 'live-bug' position, i.e. with legs downwards for all sockets 28-pins and above. 20-pin sockets are "dead-bug" loading. Push the device down to lock in place, then push down the socket sides to release.

See figure 3.1 below:

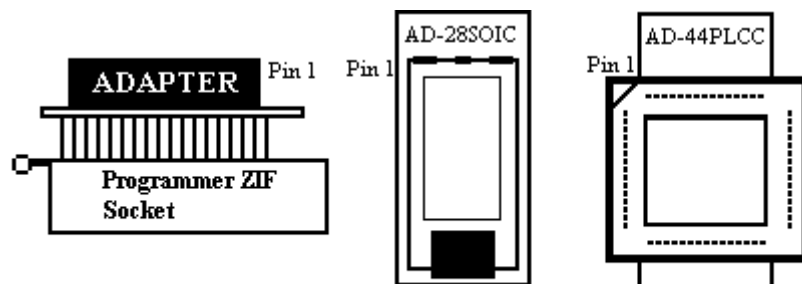


Figure 3.1 - Adapter Usage

4. Getting Started

We have provided some Quickstart tutorials to help you start using the programmer as quickly as possible. If you have not used a programmer before, these may help you to begin. However, these should not be taken as a comprehensive guide to the programmer's capabilities. The following tutorials are provided:

- TUTORIAL 1.** ChipWin - Programming a Microcontroller from a master device
- TUTORIAL 2.** DOS - Programming a Memory or Microcontroller device from a file on disk.
- TUTORIAL 3.** DOS - Programming a Microcontroller from a master device
- TUTORIAL 4.** DOS - Programming a PAL or GAL from a file on disk

These tutorials are provided to get you using the programmer quickly for simple tasks. However, we recommend that you read the reference sections of the manual if you have any difficulty and before attempting any more complex operations.

4.1 Before You Start

If you are not sure how to select menu items, please see section 5.3 for more information.

Figure 4.1 below illustrates the orientation of chips in the programmer. The chip is located at the bottom of the ZIF socket with the bottom pins next to the ZIF socket handle.

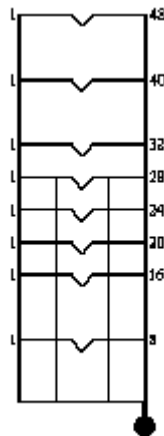


Figure 4.1 - Device Location

If you are using a socket adapter, please see the instructions with the adapter for how to place the device in the socket.

4.2 Quickstart tutorials

TUTORIAL 1: ChipWin – Programming a Microcontroller from a master device

1. Double click on the ChipWin icon to start the software
2. Select the Device you are using, either by clicking on the `SELECT DEVICE` icon from the toolbar, or by choosing `Programmer/Select Device` from the menu bar.
3. In the Database viewer, select the `Manufacturer` (for example, Atmel) in the left-hand box, expand this manufacturer and select `Microcontrollers`. All the supported Atmel Microcontrollers are then listed in the Right Hand column. Select the required part, and package. If a package adapter is required, the Data I/O adapter code is displayed at the bottom of the Viewer box. Press `Accept` to select the part.
4. All the device specific information is listed at the bottom of the screen.
5. Now select the either the `Device Operations` icon from the toolbar, or select `Programmer/Operations` from the menu bar.
6. If the `READ` icon is 'grayed' out on the Operations toolbar, Close the Operations box and select `File/Enable Edit Mode`. Now open the Operations menu again as per #5.
7. Now place the device in the programmer socket and select `READ`. Change any of the parameters as required and press `Accept`.
8. The contents of the device are now read into the buffer. Any available register information is displayed.
9. Now place the Blank device in the socket.
10. Select the `Erase, Program, Verify` icon. This option also performs a pre-program test as set in the `Programming Options`. Select the required `Programming Parameters` (by default, they will be set to the same as those read from the Master device), and press `Accept`.
11. The device will then be programmed and verified. If all is well, the `Programming Successes` counter will be incremented by one.

TUTORIAL 2: DOS - Programming an EPROM or Microcontroller from a file on Disk

1. Type `CWEPROM <ENTER>`
2. Select the device you are using as a master, using `PART/TYPE`. Select the device type you are wishing to use and press `<ENTER>`. You will then be given a list of manufacturers who support the selected device type. Select the appropriate manufacturer and press `<ENTER>`. You will then be prompted for the device name. Choose the correct device name from the menu you are given and press `<ENTER>` to accept. The details of the device you have chosen will now be displayed at the bottom of the screen.
3. Ensure that the data memory buffer size is large enough to hold the data to be read from the master device. You can change the buffer size using `BUFFER/BUFFERSIZE`. See section 6.2.1 for more information.
4. Insert the master device in the programmer.
5. Select `ACTION/READ` to read the contents of the chip into the data memory buffer. Press `<F10>` to accept the default addresses.
6. Select the device you are copying to if it is different (see 2.)
7. Now place the blank copy device in the programmer socket.
8. Select `ACTION/AUTOPROGRAM`. Press `<F10>` to accept the default addresses.
9. The message: *Device programmed and verified* will be displayed.

TUTORIAL 3: Programming an EPROM or Microcontroller from a file on disk

1. Type `CWEPROM` <ENTER>
2. Select the part you are using as a master, using `PART/TYPE`. Select the device type you are wishing to use and press <ENTER>. You will then be given a list of manufacturers who support the selected device type. Select the appropriate manufacturer and press <ENTER>. You will then be prompted for the part name. Choose the correct part name from the menu you are given and press <ENTER> to accept. The details of the device you have chosen will now be displayed at the bottom of the screen.
3. Ensure that the buffer size is large enough to hold the data. You can change the buffer size using `BUFFER/SIZE`. See section 6.2.1 for more information.
4. Select `FILE/LOAD`
5. Specify the path and filename where your data is located.
6. Select the file format used (HEX auto-recognition will detect any of the HEX formats available)
7. Press <F10> to accept the default settings and load the data. A checksum will be displayed on the screen.
8. Select the part you are copying to if it is different (see 2.)
9. Place the device to be programmed in the programmer.
10. Select `ACTION/AUTOPROGRAM`. Press <F10> to accept the default addresses.
11. The message: *Device programmed and verified* will be displayed.

TUTORIAL 4: Programming a PAL or GAL

1. Type `CWPAL` <ENTER>.
2. Select the part you are going to program using `PART/MANUFACTURER`. Select the manufacturer and press <ENTER>. Choose the device part name from the list on the screen by highlighting the correct part and pressing <ENTER>.
3. Select `FILE/LOAD`
4. Specify the path and filename where your JEDEC file is located.
5. Press <F10> to accept the defaults. A checksum will be displayed once the file has loaded.
6. Place the device to be programmed in the ZIF socket.
7. Select `ACTION/AUTOPROGRAM`. The device will now be erased (where relevant), programmed, verified and any test vectors present in the JEDEC file will be applied.

5. Using the DOS Programmer Software

5.1 Basic Software Configuration

The operating software contains the following programs.

CWEPROM.EXE : Used to program the various Memory and Microcontroller device types

CWPAL.EXE : Used for Programmable Logic.

CWCTEST.EXE: Used to identify and test TTL and CMOS logic devices

These are menu driven programs which contain all the features required to program and test devices. The actual programming information for individual devices is held in library files, which allows easy updating of the software to accommodate new devices and algorithm updates.

In order for these programs to run correctly, the appropriate .INI files must be configured correctly. If the software has been installed as described in section 2.3, the default .INI files will contain the correct path and directory information required by each of the programs. If however, on running EPROM for instance, the message:

Cannot find database. Modify the ini file

is displayed, then the [Directory] section of the IT_EPROM.INI file is not pointing to the correct path. If either the PAL or EPROM libraries are not correctly mapped, after selection of a device within the software, the first selection off the ACTION menu will display the message:

Bad or missing library file

This may also be a result of installing the 40-pin programmer software when trying to use a 48-pin programmer, or vice-versa. Using any ASCII editor, edit the IT_EPROM.INI file. If the software has been installed on C:\CHIPWRITER, the [Directory] section of the .INI file should read:

```
[Directory]
Memory Libraries (48-pin) = C:\CHIPWRITER\Libs\Memory48
Pal Libraries (48-pin) =C:\CHIPWRITER\Libs\Pals48
Pars=C:\CHIPWRITER\Pars
```

(for the 40-pin programmers, simply remove the '48' reference). Batch software is also included which allows the programmer to be operated from DOS batch files. Details of the commands available are shown in section 8.

For some programmable logic devices (Xilinx EPLDs and Altera MAX), separate programs are supplied to handle non-JEDEC standard files.

5.2 Screen Layout

Each program displays information as follows :

Menu headings are displayed across the top of the screen

Device programming details are displayed in the bottom window.

Data memory contents are displayed in the main window.

The software and library version numbers are displayed at the top of the screen, just below the Menu bar. These are useful to note should you need to contact Technical Support or to check for new version numbers on the BBS or Internet.

5.3 General Menu Selection

The software has been designed to make the programmer as simple as possible to use. All functions are contained within the menu system. The menus are obtained by pressing the <ESCAPE> key. The user can then move around the menus by pressing the cursor keys. To select an option press <ENTER> .

Each menu function is available by pressing the first letter of the menu function. For instance, the PART pull-down menu is available by pressing the 'P' key. Some options can be selected by pressing **ALT+letter**. These are indicated by a triangle next to a letter on the menu display. For example, Auto-program can be selected by pressing **ALT+A**. A full list of shorthand keystrokes is given in Appendix A.

5.4 Dialogue Boxes

The menu system makes extensive use of dialogue boxes. Within these boxes the following prompt types may appear :

| | | |
|---------------|---|---|
| Prompt | : | requires you to type in some information |
| | > | will display a series of options when you press <ENTER> |

Inside the dialogue boxes use the cursor keys to move between options; use <ENTER> to select the contents of the dialogue box. Use <F10> to accept all chosen options within the window. If you wish to abort from the window, press <ESCAPE>.

5.5 Changing the Settings

Before starting, you may wish to change any of the following. They are found in the OTHER menu. This will have to be done for both CWEPPROM and CWPAL:

OTHER/PARALLEL PORT NO

Selects parallel port to be used. This auto-detects each time that the software is accessed when the port setting in the INI file does not agree with the actual port setting. The setting is then stored in the appropriate INI file on exit. Once the parallel port has been chosen it will be remembered when next using the programmer.

OTHER/READ COLOR INFO

To choose whether the display is MONO or COLOR

OTHER/SNOW PRECAUTIONS

Usually set to OFF but set it to ON if you have problems with snow on your monitor. (affects older PC's with CGA)

OTHER/HI-RES MODE

Select ON for 40/50 line mode. Effective only with EGA and VGA monitors.

5.6 Using a mouse

All three application programs can be mouse driven. However your mouse driver software must previously have been installed. This is normally done automatically in your CONFIG.SYS file (by loading a device driver such as MOUSE.SYS) or in your AUTOEXEC.BAT file (by invoking a mouse driver program such as MOUSE.COM). If you are unsure how to do this please refer to your system manual or the manual supplied with your mouse. **CWEPPROM.EXE**, **CWPAL.EXE** or **CWCTEST.EXE** automatically enable the mouse if one is installed (see later if you wish to disable it).

5.6.1 Left Hand Mouse Button (LMB)

The LMB is used to select items in a list, such as a menu selection or a device name from the device list. Simply move the mouse to the required item and press the LMB. If the item is a menu heading, such as FILE, ACTION etc. then this will open up that particular menu. If the item is a sub-menu or part of a list this will highlight the chosen item. To select the highlighted item DOUBLE CLICK on the LMB. i.e. QUICKLY press and release the LMB twice. DOUBLE CLICKING performs the same action as PRESSING <CR>

5.6.2 Right Hand Mouse Button (RMB)

Pressing the RMB is exactly the same as pressing <ESC>

5.6.3 Middle Mouse Button (MMB)

Pressing the MMB is exactly the same as pressing <F10>. When this key is a possible input the window being displayed will have F10 in the bottom border. Pointing at the text F10 and pressing the LMB will also produce the same result (For two button mouse users.)

5.6.4 Disabling the Mouse

To disable the use of the mouse invoke each of the programs with the /NM option. e.g.

```
CWEPROM /NM
```

5.7 Edit and View modes and Scroll Bars

In order to observe data in the buffer without risking modification a VIEW mode is available in the BUFFER menu. This will only allow scrolling, paging etc. Mouse users can enter this mode directly by pointing at the buffer window and pressing the LMB. In order to make moving around the buffer much easier SCROLL BARS have been added to the window borders.

5.8 Running the Software under Windows

All three programs can be run in a DOS window under Windows 3.1x or Windows 95/98. To create an ICON for CWEPROM.EXE, for example on Windows 95, do the following:

- Select the CWEPROM.EXE icon in the ChipWriter sub-directory.
- Drag to the Desktop to create a Short-cut.
- Change Filename to CWEPROM.
- Double click on ICON to start

Repeat for CWPAL and CWCTEST. Use Optional Parameters if required. The mouse functions should work in a DOS window in 386 enhanced mode in Windows 3.1x or Windows 95, but if you have trouble see the Windows readme file.

5.9 Using the Programmer Software over a Network

It is possible to run the programmer software from a network hard drive with the programmer itself connected to the local computer. However, the appropriate .INI files will have to be modified in order to map the PAL and EPROM libraries, and PARS sub-directory to the correct full path name. The **CWEPROM.EXE** and **CWPAL.EXE** files use their respective .INI files to locate the Devices menu and device libraries.

If the database parameters PARS sub-directory is not correctly mapped, on running the software, the message:

Cannot find database. Modify the .ini file

will be displayed.

If either the PAL or EPROM libraries are not correctly mapped, after selection of a device, the first selection from the ACTION menu will display the message:

Bad or missing library file

5.10 Saving software settings

When exiting the programmer software, settings such as parallel port, the device chosen, etc. are automatically saved to avoid having to repeatedly set up the chosen options.

It is also possible to save the whole of a device environment, including full file load and programming options, in an **ENVIRONMENT** file. See section 6.2.8

6. Memory and Microcontroller Programming

This section covers the Memory and Microcontroller device types which are programmed using **CWEPROM.EXE**.

To use this program, at the DOS prompt, type:

```
CWEPROM <ENTER>
```

A HELP screen is available by typing:

```
CWEPROM /HELP <ENTER>
```

6.1 Device Selection

In order to tell the programmer software which programming algorithm to use, it is necessary first to select the part which is being used.

6.1.1 To select a device

To do this, enter **CWEPROM.EXE** and select:

```
PART/TYPE
```

You will then be given a choice of device types to select. Choose between:

```
EPROM's  
EEPROM/FLASH/NVRAM's  
Serial (E)EPROM's  
Microcontrollers  
BPROM's
```

Highlight the type required or press the first letter, and press <ENTER>.

A list of manufacturers will then be displayed on the screen who support the selected device type. Use the up and down cursor keys to find the manufacturer of the device you are using. You can also press the initial letter of the manufacturer you need, e.g. After selecting EPROM's, pressing M will take the cursor down to MACRONIX. When the cursor is highlighting the required manufacturer, press <ENTER>.

The software then displays a list of all the supported devices of the manufacturer and type selected. Highlight the required device name and press <ENTER> again.

The details of the selected device will then be displayed in the bottom part of the screen. The appropriate device library file name will be displayed at the top of the screen, below the menu bar, together with the library version number.

6.1.2 Changing the device selected

Once a device has been selected, to select another part of the same manufacturer and type, press:

```
PART/PARTNAME
```

To select another manufacturer of the selected part type, select:

```
PART/MANUFACTURER
```

6.1.3 Which part?

Generally it is necessary to select the device name which is as close as possible to that stamped on the chip you are using. For example, a PIC16C54 microcontroller in windowed DIP package will be marked PIC16C54/JW, whereas an OTP version could be marked PIC16C54-XT/4. The software uses the correct nomenclature according to the manufacturer's data sheet. Speed and package indicators are only required when they affect programming algorithms.

6.1.4 Low Voltage Parts

Available on ChipWriter, low voltage devices are selected exactly as other devices. In the device information in the bottom window, you will notice that the Vcc voltages during program and verify (VccP & VccV respectively) will be a combination of 5V, 3.3V, 2.7V and/or 1.8V. As most programmers are still unable to provide 1.8, 2.7V and 3.3V circuits, manufacturers still allow the devices to be programmed at 5V. The ChipWriter programmers can not only verify a low voltage device at its nominal operating voltage, they can also program with voltages down as low as 1.8V as well, giving much greater flexibility.

6.2 Opening data files

Files can be loaded from disk into the data buffer and saved to disk after editing using the LOAD and SAVE options in the FILE menu. See section 6.2.8 for more details on saving files.

6.2.1 Setting the data memory buffer size

It is necessary to ensure that the data memory area has been set to a size large enough to accept the contents of the file to be loaded. During the device selection process, the buffer should be increased sufficiently to hold files of the size required for that part. However, there may be occasions when the buffer requirement is even larger. Do this using:

`BUFFER/BUFFER SIZE`

Enter the required size in KB. The maximum buffer size available is determined by the available memory of the PC. The programmer software uses approximately 150K plus the chosen buffer size. The EMS software will use expanded memory for the buffer before using conventional DOS memory. If the buffer cannot be increased sufficiently, and gives a maximum size of typically 384KB, then the EMM386 driver has not been correctly installed. Check the CONFIG.SYS file in the root directory on your hard drive to check that the following line exists:

`DEVICE=C:\DOS\EMM386.EXE`

Also ensure that the **NOEMS** option has **not** been included on this line. DOS may be replaced by WINDOWS if you are running under Windows 95/98. It may also be possible to run MEMMAKER and select YES when prompted for EMS memory requirements.

6.2.2 Opening A File

Select:

`FILE/LOAD`

This then displays a number of different options:

| | |
|--|--|
| <code>FILE NAME:</code> | Enter the required filename |
| <code>FILE TYPE:</code> | Select the required file format (see section 6.2.3) |
| <code>FILE WINDOW LOW ADDRESS:</code> | Enter the required file start address. Default = 0 |
| <code>FILE WINDOW HIGH ADDRESS:</code> | Enter the required file end address. Default = FFFFFFFF |
| <code>FILE INCREMENT:</code> | Select from Every Byte, Every 2nd Byte etc. up to Every 4th Word |

BUFFER INCREMENT: Select from Every Byte/Word etc. up to Every 8th Byte/Word
BUFFER START ADDRESS: Enter the required address. Default = 0
PRE-FILL BUFFER WITH FF: Select ON or OFF. ON will destroy all current buffer contents.

Pressing <F10> accepts the parameters and opens the selected file, displaying the checksum when loaded (See 6.2.5).

6.2.3 File Format

The file type can be selected by pressing <ENTER> to see the options available. Choose from:

```
RAW/BINARY
ASCII-SPACE-HEX
ASCII-OCTAL
HEX-AUTO RECOGNITION
HEX-MOTOROLA (S1 ,S2 OR S3 RECORDS)
HEX-INTEL MCS86
HEX-TECTRONIX
HEX-EXTENDED TECTRONIX
HEX-TI TAG
HEX-MOSTech
```

Choosing HEX-AUTO RECOGNITION is generally the easiest way of loading a HEX formatted file. However some linkers put additional headers at the start of the HEX file which can confuse the loader when using this option. Hence the other options.

ASCII-SPACE-HEX embodies the formats of ASCII space HEX, ASCII ' HEX etc. The software assumes that the file consists of HEX digits separated by non HEX characters. Checksum information is not looked for and no separators are needed. e.g.. If a file consisted of A BCD EF this would appear as

| | | | | |
|---------|----|----|----|----|
| Address | 0 | 1 | 2 | 3 |
| Data | 0A | BC | 0D | EF |

Please Note: When using the PICALC assembler to produce files for Microchip PICs, use the option -F INHX8M to save the file as an 8-bit Intel-Hex file, then load as HEX-Auto Recognition. **WARNING:** Any line within your code which sets the format to anything else will NOT be overridden by entering the INHX command at the command line.

6.2.4 Handling large files and devices

In order to facilitate the programming of EPROMs and the loading of files whose size is greater than the maximum buffer size the FILE/LOAD command (ALT-L) has the following parameters:

```
FILE WINDOW LOW ADDRESS (FWLA)
FILE WINDOW HIGH ADDRESS (FWHA)
FILE INCREMENT (FINC)
BUFFER START ADDRESS (BADD)
```

A byte is ONLY loaded into the buffer if its FILE ADDRESS is inside the window limits defined by the above (inclusive) range. Data at FWLA is loaded into the buffer at BUFFER START ADDRESS. The FILE INCREMENT works within this window. See Figure 6.1 below.

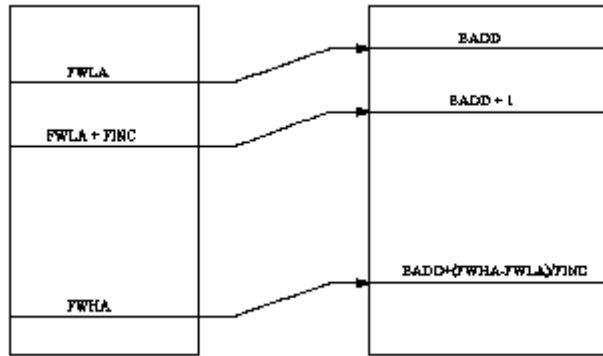


Fig. 6.1 – Handling large files

If you have file loading problems please check the parameters above. Normally FWLA=0 and FWHA=FFFFFFFF (i.e. the whole file is to be loaded). However these parameters are stored each time the program is quit. If in a previous session FWLA was changed to 10000 (HEX) then files with information in the range 0-FFFF will appear empty.

6.2.5 File Checksum

When a file has been opened and loaded into the data buffer, a checksum is calculated on the contents of the file and displayed on the screen. This is displayed as three figures:

- The sum of unsigned bytes with the carry being added to the MSB of the checksum.
- The complement of (a)
- The arithmetic negative of (a)

NOTE: The checksum displayed on FILE/LOAD is the checksum of the FILE only, and does not necessarily include the whole address range of the selected device. A checksum of the device, or of a particular address range, can be calculated using BUFFER/CHECKSUM as explained in 6.3.5.

6.2.6 Wild card Loading

Wild cards may be used in the filename to perform drive/directory searches. Wild cards consist of * and ?.

- * matches to any number of characters
- ? matches to any one character

A list of matching filenames is displayed alphabetically (Sub-Directories first) and the user may select using the cursor keys or the first letter of the name.

6.2.7 Saving a file to disk

Data edited in the buffer area or read in from a device can be saved to disk in a number of different formats. It is recommended that frequently used files are saved in binary mode to save disk space and reduce loading time.

FILE/SAVE AS

Displays two options:

SAVE ALL DEVICE AREAS

Saves the whole of the device address range and, where applicable, the configuration byte stored at the end of the device address range.

SAVE MEMORY BLOCK

Saves a user defined address range from the data memory area.

Both options allow the file to be saved in a number of formats:

```
RAW/BINARY
ASCII-SPACE-HEX
ASCII-OCTAL
HEX-MOTOROLA (S2 format)
HEX-INTEL
```

To save any modifications to the currently loaded file, simply select: FILE/SAVE

6.2.8 Saving a device configuration

Using the `SAVE ENVIRONMENT` command from the `FILE` menu saves a set of system parameters to an `TASK Environment (.TSK)` file for future use. The system parameters include the currently selected device, the open data file, the file parameters, and the device options.

If you are using microcontrollers, or memory devices with configuration options, all the configuration bits can be saved in the Environment file as an extra byte at the end of the data file, or at the location specified in the manufacturers programming algorithm. This means that the next time the device is used, all of the configuration options have been configured exactly as they were the previous time, saving time, and reducing the chance of operator error.

To implement this feature, choose your required device and load the required file, and select `ACTION/PROGRAM`.

All of the programming options are then displayed. Select all of your required configuration options and press `<F10>`. Once the device has programmed, the data buffer will be modified with the appropriate configuration byte (see Chapter 6 for full details on programming). Then save the file, either over-writing the original, using `FILE/SAVE`, or creating a new file using `FILE/SAVE AS`. For example, with a National Semiconductor COP878x device, this is stored in location #1000, with an AMD 29F040 Flash PROM, this would be stored at address #80000.

When you have the program configured the way you want it, select:

```
FILE/Save Environment
```

Type in the required Environment filename, prefixed with the appropriate drive and directory in which you want the configuration file to be saved. By default, the extension `.TSK` will be appended to the filename. Press `<F10>` to accept the filename.

To load a previously saved Environment file, select:

```
FILE/Load Environment
```

Enter the Environment filename and press `<F10>` to accept, and the whole device configuration is loaded, including the data file and the whole of the program parameters. Alternatively, use:

```
CWEPROM EnvName.TSK <ENTER>
```

from the DOS command line. This runs the `CWEPROM` software and automatically loads required Environment file and device configuration.

This procedure can be set-up in both the DOS and Windows software. However, only the DOS version can be used when setting up the Command Line Batch software - see chapter 8.

6.3. Using the Editing Facilities

As well as loading from disk or reading a pre-programmed device, data to be programmed can be entered directly into the buffer from the keyboard. In addition, files can be amended using the BUFFER menu options.

6.3.1 Entering EDIT mode

To edit the contents of the data buffer, select

`BUFFER/EDIT`

This allows you to type in HEX or OCTAL code, or ASCII characters. Once EDIT is selected, the cursor will move to the HEX display in the buffer window. Use the <TAB> key to switch between HEX and ASCII. In HEX mode, only keys 0-9 and A-F are allowed. In OCTAL mode only keys 0 to 7 are allowed. The cursor keys can be used to move around the buffer.

6.3.2 Viewing without making changes

If you wish to view the buffer contents beyond the first screen without danger of making unintentional changes to the data, use:

`BUFFER/VIEW`

This allows you to view the contents of the buffer as HEX or OCTAL code or ASCII characters. Once VIEW is selected, the cursor will move to the HEX display in the buffer window. Use the TAB key to switch between HEX and ASCII.

6.3.3 Quickly Moving Around the Buffer.

The <HOME> and <END> keys allow you to move more quickly. Press once to move to the beginning (end) of the line, twice to move to the beginning (end) of the page or three times to move to the beginning (end) of the file.

When you have finished editing or viewing the buffer, press <ESC> to return to the main menu.

6.3.4 Quick buffer editing commands

`BUFFER/FILL`

Allows you to fill a defined area of the buffer with information in byte, word or long word format. Enter the buffer start and end addresses and the information to be inserted. Use less than 3 characters for a byte, 3 or 4 for a word, more than 4 for a long word. Press <F10> to accept.

`BUFFER/COPY`

Copies a defined block of buffer memory to a new location.

`BUFFER/SEARCH`

Searches for data in HEX, OCTAL or ASCII format. Use the Up/Down arrow in the dialogue box to move between HEX, OCTAL and ASCII data entry.

`BUFFER/GOTO`

Goes to a chosen address in the buffer.

`BUFFER/SWAP BYTES`

Swaps odd and even bytes within a defined range. Useful for 16 bit wide EPROMs.

6.3.5 Buffer checksum

The command:

`BUFFER/CHECKSUM`

Gives the option of performing a checksum on either the contents of a user-defined range in the buffer, or the selected device.

Selecting `MEMORY BLOCK CHECKSUM` provides an option box:

`BUFFER START ADDRESS:` default to 0
`BUFFER END ADDRESS:` default to end of buffer
`CHECKSUM TYPE:` Select between:
 Sum of Bytes
 Sum of Words
 Sum of Long Words
 CRC 32 bit

Choosing `DEVICE CHECKSUM` just gives the option of entering a device `START` and `END` address.

Whichever option is selected, the checksum is displayed as three figures:

- a) The sum of unsigned bytes in the memory range with the carry being added to the MSB of the checksum.
- b) The complement of (a)
- c) The arithmetic negative of (a)

NOTE: The memory area can be larger than the size of the opened data file. If there is a discrepancy between the expected and calculated checksums, recalculate specifying the start and end address of the file. Checksums for Microcontrollers are displayed as defined by the manufacturer's data sheet or algorithm.

6.3.6 Buffer Display Modes

The contents of the buffer can be displayed in 8-bit HEX, 8-bit OCTAL, 12-bit (16-bit storage, LSB = low address) or 16-bit mode (LSB = low address). To change the display, select:

`OPTIONS/DISPLAY MODE`

and select the display mode required.

6.3.7 Printing the Contents of the Buffer

A print-out of all or part of the buffer contents can be obtained by selecting:

`BUFFER/PRINT`

Specify the start and end addresses of the block to be printed, and the name of the output device (e.g. LPT1)

6.4 Reading and Checking Devices

This section explains how to read the contents of a programmed chip, verify the contents against known data, check the device signature and identify a device from its signature, and perform bit tests and blank checks on devices.

6.4.1 Reading a Device

To read the contents of a device, select the correct part name from the `PART` menu, as described in section 6.1,

and place the device to be read in the programmer socket.

Please note that some devices CANNOT be read, either because the chip type itself is designed not to be read for security purposes, or because the on-chip security features have been enabled when programming.

Before reading EPROMs, ensure that the buffer size is set large enough to contain the data (see section 6.2.1 on setting the buffer size).

Please Note: A 16-bit device with a maximum device address of 128K will require 256K bytes of buffer space.

To read the contents or part of the contents of a device, select

ACTION/READ

The programmer software then quickly accesses the library file for the programming algorithm required. With Memory devices, an option box appears requesting Device Start and End address, and Buffer start address:

DEVICE START ADDRESS: Enter the first address from which you wish to start reading data from the device. Default = 0.

DEVICE END ADDRESS: Enter the last address from which you wish to read data in the device. Defaults to the highest address available for the selected device.

BUFFER START BYTE ADDRESS: Enter the buffer address where you wish to begin reading the data into. Default = 0.

Press <F10> to accept the options displayed and the device will then be read. If the selected device is a microcontroller, then the entire contents of the device are read and placed into the appropriate buffer locations. The contents of the device will then be visible in the buffer, and a checksum will be displayed.

Once the contents of a device have been read, they can be edited using the BUFFER/EDIT option, and then saved to disk, or programmed into another device. See section 6.3 for more detail on editing data.

6.4.2 Verifying the contents of a chip against a file.

To verify a chip's contents against known data, first select the part you are using. Ensure that the buffer size is large enough to handle the file. Then load in the file against which you need to verify the data. Place the device to be verified in the programmer and select:

ACTION/VERIFY

If the data matches, the message *VERIFY OK* will be displayed. If there is a mismatch, the message will indicate at what address it has failed to verify the data.

6.4.3 Verifying the device type

Many newer devices contain Electronic Signature codes which enable you to electronically identify the device. It is possible both to verify a known device type against this ID code. The ID code is also used by your programmer to check that the correct device type has been selected before proceeding with any process that may damage a chip. However, not all devices contain this ID code, and the checks for electronic signature may themselves damage a device without a signature, or one of a different number of pins from that expected. This means that these checks cannot be foolproof and need to be used with some caution.

To verify the signature of a device, select:

ACTION/VERIFY SIGNATURE

If the database contains a manufacturer's code and device code they can be verified against that of the device.

6.4.4 Checking if a device is blank

In order to program a device, it must either be fully blank, or be capable of containing the information to be programmed into it in addition to the information it already contains. In order to test whether a device can hold additional data, open the required data file and select:

ACTION/BIT TEST

This checks to see if a non-blank device can be programmed with the data currently held in memory. For example, with an EPROM whose unprogrammed state is #FF, #F0 may be programmed to #00 but not to #0F. In order to test whether a device is completely unprogrammed, select:

ACTION/BLANK CHECK

This option checks if the device is blank (all 1's or all 0's depending on the actual part).

Please Note: It is possible for a device to appear blank even though it is not fully erased. If problems are encountered in programming a device, try erasing it for a longer period and then reprogramming.

Either the Blank Check or Bit Test can be performed when you select ACTION/AUTOPROGRAM. This is selected from OTHER/PROGRAMMING OPTIONS. See section 6.5.1.

6.5 Programming Devices

IMPORTANT NOTE

Data I/O only use the specified programming algorithms as provided by the device manufacturers. Whilst we make every effort to ensure that programming algorithms are correct, the said algorithms are not guaranteed by the manufacturers and therefore cannot be guaranteed by Data I/O.

Users are advised to ensure that devices work correctly in their systems before programming large quantities.

6.5.1 Programming Options

Prior to programming, there are a number of different options available which can be selected using OTHER/PROGRAMMING OPTIONS. These are generally more applicable when using the programmer for larger quantity programming.

SERIAL NUMBERS: For production programming, it is often a requirement to have a unique serial number programmed into a designated location(s) on the selected device. The PROGRAM process implements the serialization options selected here. (See 6.5.2)

Select from:

None
Sequential Hex: LSB in low memory
Random Hex
Date, Weekday, Time in HEX:YYYYMMDDWDHMMSS
Sequential BCD: LSB in low memory
Random BCD
Date, Weekday, Time in BCD:YYYYMMDDWDHMMSS

SN BUFFER ADDRESS: Enter the required start address of the serial number.

SN DIRECTION: Select LSB or MSB in Low memory. For example: A 4-byte serial number starting at 0000, being incremented by 1, starting at buffer address 00. Selecting LSB will mean that address 0 is incremented to FF, with address 01 then being the overflow, etc. Selecting MSB will mean address 03 is incremented to FF, with address 02 then being the overflow, etc.

SN STORAGE: Select between NIBBLE, BYTE and WORD.

SN Buffer Increment: Default is 1.

SN LENGTH: Enter the number of bytes required for the serial number - up to 8 bytes.

SN (bytes 0-3): Least significant 4 bytes (8 digits)

SN (bytes 4-7): Most Significant 4 bytes (8 digits)

AUTO-DEVICE CHECKSUM: Select OFF, BEFORE SECURING or AFTER SECURING.

If turned ON, after a device has been successfully programmed and verified, the device checksum is displayed, either before or after the security fuse has been programmed.

Device Checksum Ignores SN: Select ON or OFF.

If a unique serial number is being programmed into each device, the overall checksum will change after each program cycle, invalidating the checksum as a fail-safe checking procedure. If this option is turned ON, then checksum will be calculated using only the UNCHANGED data.

AUTOPROGRAM PRE-TEST : Select None, USE BIT TEST or USE BLANK CHECK. This determines which kind of pre-testing is performed when the Auto-Program option is selected.

ASK FOR PARAMETERS ON REPEAT: Select ON or OFF

With this option set to ON, after each program cycle, the Device Programming Parameters box is displayed. If a large quantity of devices are being programmed with exactly the same information, then it is often desirable not to display the parameters box once the device has been configured correctly. Selecting OFF also ensures protection against accidental alteration to the programming parameters.

FORCE MARGINAL VERIFY: Select ON or OFF. After any program operation, an automatic verify pass is performed. This is at the nominal operating voltages as specified by the manufacturer which may be a combination of 5V, 3.3V and/or 1.8V. Selecting ON provides additional verify passes at +/- 5% of each of the designated verify voltages. See VccV for the selected verify voltages in the Device parameters window at the bottom of the screen.

6.5.2 Programming Operation

Two different options are provided for programming devices.

ACTION/AUTOPROGRAM

performs either a bit test or blank check, depending on the Programming Options (see 6.5.1), followed by a chip erase, if required (for electrically erasable devices), and program and verify, whereas:

ACTION/PROGRAM

simply programs and verifies the device.

In normal circumstances, the AUTO-PROGRAM option should be selected when you do not wish to retain any data which may already be in the device. Selecting either option displays a box which enables the required Programming Parameters to be selected. This information will differ depending on the type of device and the on-chip functions supported.

The most basic configuration will offer the following options:

DEVICE START ADDRESS: Enter the address where you wish the programming operation to start. Defaults to 0.

DEVICE END ADDRESS: Enter the last address of the device you need programming. Defaults to the device end address.

BUFFER START ADDRESS: Enter the address of the first byte in the buffer from where you want to start programming. Default is 0.

If the Sequential Serial Numbers option has been selected within the Programming Options (see 6.5.1), then the Serialization parameters will also be displayed here. These can also be modified if required.

SERIAL NUMBER (bytes 0-3): Least significant 4 bytes

SERIAL NUMBER (bytes 4-7): Most Significant 4 bytes

Microcontrollers will generally offer a number of options in addition to the basic device parameters. See 6.6.

To accept the address selections or defaults, press <F10>

The programmer will first check the device selection and position in the socket, then perform either a Bit Test or Blank Check, depending on what has been selected in the PROGRAMMING OPTIONS, and then proceed with programming. Non-blank Flash devices will always be erased during the pre-test stage.

While a device is being programmed, a message will be displayed in the center of the screen showing the address which has been reached. Following programming, an automatic verify will be conducted and if all is well, the message *DEVICE PROGRAMD AND VERIFIED* will be displayed. If AUTO-DEVICE CHECKSUM has been selected, the device checksum will now be displayed.

If you need to program another device using the same configuration, simply press <F10> to repeat the process. If ASK FOR PARAMETERS ON REPEAT has been set to ON, then the programming parameters box will be displayed, which allows you to change any of the options. Again, just press <F10> to continue. If the Repeat option is set to OFF, then the parameters box will not be displayed and a single keystroke is all that is required to repeat the programming cycle.

6.5.3 Overprogramming devices

If you have a device which already contains some data, and you wish to add data to it, use

ACTION/PROGRAM

The same options are provided as for AUTOPROGRAM.

PROGRAM will program a device with the selected contents of the buffer. Program should only be used when programming part of a device while retaining data in other address locations. In normal circumstances, use AUTOPROGRAM.

6.6 Microcontroller Programming

Microcontrollers generally have a number of specific features which make their programming different to that of memory devices. The ChipWriter programmers are specifically designed to handle a wide range of microcontrollers, and can generally support all of the optional features which are available on them.

6.6.1 Address Locations

The programmer software handles address locations as they appear on the device. For example, default buffer and device start addresses are at the addresses where the data is held in the device, which is not always a single block of contiguous memory. It is not possible to edit the device start and end addresses of microcontroller devices.

6.6.2 Security Features

For microcontrollers, the security features are supported as part of another option within the Programming

Parameters dialogue box once the PROGRAM option has been selected from the ACTION menu. Security status is performed automatically before a BLANK CHECK or PROGRAM. To manually check the security status of a pre-programmed device, use:

ACTION/SECURITY STATUS

This gives the status of the device security features. The information displayed is representative of that on the device data sheet.

P represents a security bit which is programmed
U represents an unprogrammed security bit

IMPORTANT NOTE: Some devices cannot indicate security status when the security fuse(s) has been programmed and in these cases the device may look blank. Thus the device may fail programming attempts until it is erased. If any problems arise with devices that have on-chip security, please erase the device first and retry.

ACTION/VERIFY ENCRYPTION

If encryption is in effect/relevant this will verify the encrypted buffer data with the contents of the device.

6.6.3 Device- Specific Features

Where device specific features are supported, the options available will be displayed in the Programming Parameters box after selecting PROGRAM or AUTOPROGRAM (See 6.5.2). Microcontroller programming options include:

- Security
- Encryption Array
- Oscillator type
- Brown-Out protection
- Watch Dog Timer
- RAM size selection
- Programming Mode

Please refer to the manufacturers' data sheets for more information on these options. If there are any programmable options which are not included, please contact Data I/O for possible updates.

6.7 Erasing devices

For Flash memory, and some parallel EEPROM parts which support an erase feature, use:

ACTION/ERASE

This electrically erases the data held on the chip.

To erase all other EEPROM's, as well as NVRAM's, fill the buffer with (#FF) and program the whole device.

7. Programmable Logic

This section covers the Programmable Logic device types which are programmed using **CWPAL.EXE**. To use this program, at the DOS prompt, type:

CWPAL <ENTER>

7.1 Device Selection

In order to tell the programmer software which programming algorithm to use, it is necessary first to select the part which is being used.

7.1.1 To select a device

To do this, enter `CWPAL.EXE` and select:

`PART/MANUFACTURER`

A list of supported manufacturers will be displayed on the screen. Use the up and down cursor keys to find the manufacturer of the device you are using. You can also press the initial letter of the manufacturer you need, e.g. pressing `N` will take the cursor down to `NATIONAL SEMICONDUCTOR`. When the cursor is highlighting the required manufacturer, press `<ENTER>`.

The software then displays a list of all the supported devices of the manufacturer and type selected. Highlight the required device name and press `<ENTER>` again.

The details of the selected device will then be displayed in the bottom part of the screen.

Please Note: Some devices which are supported on the programmer will not appear on the menu. This is due to the fact that they do not use JEDEC standard files and therefore are programmed using a separate piece of software. At present this affects Altera and Xilinx EPLD's. See section 7.10 and 7.11 for details of how to use this software.

7.1.2 Changing the device selected

Once a device has been selected, to select another device of the same manufacturer, select:

`PART/PART NAME`

When a new device is selected which has a different fuse map from the current device, the fuse map displayed on the screen will alter.

7.1.3 Which device?

Generally, it is necessary to select the device name that is as close as possible to that stamped on the device you are using. Selecting a different device, no matter how small the difference may seem, may result in the device being damaged. Sometimes, however, a number of devices with different suffixes will come under the same device name in the software. The programmer will check the internal electronic device signature and use the correct algorithm for the device you are programming.

When using an adapter to program a non-DIP package, select the device as normal. Any pin differences are taken account of by the adapter. Use Data I/O recommended adapters wherever possible, especially for devices where the non-DIP package is NOT a straightforward pin for pin conversion of the DIP version.

7.1.4 Low Voltage devices

Low voltage devices are selected exactly as other devices. The program/verify voltages will be a combination of 5V, 3.3V, 2.7V and/or 1.8V. Many programmers are still unable to provide the Low Voltage circuits, semiconductor vendors still allow the devices to be programmed at 5V. The ChipWriter programmers not only verify a low voltage device at its nominal operating voltage, they can also program with voltages down as low as 1.8V as well.

7.2 Opening data files

7.2.1 Opening a file

To open a logic data file, first ensure that the correct device has been chosen, then select:

FILE/LOAD

You will then be prompted for the name of the file. The software will expect the file to be in the correct format for the device selected. If there is a mismatch a warning will be displayed. Once the file has been opened, the screen will show the total number of 1's in the file, followed by the checksum. The JEDEC file loaded will include any test vectors that are present; these are displayed at the bottom of the buffer screen. These will be applied automatically when the device is programmed.

7.2.2 File format

Standard JEDEC files can be loaded into the fuse map area. These can be produced by packages such as PALASM™, OPAL™, CUPL™, PLACE™ etc. Contact device manufacturers or for details of programmable logic design packages.

In addition to JEDEC files, Altera POF and JAM format files and HEX files for Xilinx devices are accepted by the separate programs provided for these devices (See 7.11 and 7.12).

7.2.3 Wild card loading

Wild cards may be used in the file name to perform drive/directory searches. Wild cards consist of * and ?.

* matches to any number of characters

? matches to any one character

A list of matching file names is displayed alphabetically (Sub-Directories first) and these may be selected using the cursor keys or the first letter of the name.

7.2.4 Saving a file to disk

Files edited in the buffer can be saved to disk using:

FILE/SAVE AS

All current test vectors will be saved along with the fuse-map.

If a file has been converted from an old PAL JEDEC format file to a newer GAL using the -conv options, the next time it is loaded it will be in the correct format for the GAL device.

To save any modifications to the currently loaded file, simply select:

FILE/SAVE

7.2.5 Saving configuration

Using the SAVE ENVIRONMENT command from the FILE menu saves a set of system parameters to an Environment file for future use. The system parameters include the currently selected device, the open data file, the file parameters, and the device options.

If you are using logic devices with configuration options, all of the configuration bits can be saved as part of the Environment file. This means that the next time the device is used, all of the configuration options have been configured exactly as they were the previous time, saving time, and reducing the chance of operator error.

To implement this feature, use:

FILE/Save Environment

Type in the required Environment filename, prefixed with the appropriate drive and directory in which you want the configuration file to be saved. By default, the extension .TSK will be appended to the filename. Press <F10> to accept the filename.

To load a previously saved Environment file, select:

FILE/Load Environment

The Environment filename is then loaded. Press <F10> to accept, and the whole device configuration is loaded, including the JEDEC file and the whole of the program parameters. Alternatively use:

PAL EnvName.TSK <ENTER> in DOS.

7.3 Reading and Checking Devices

This section explains how to read the contents of a programmed chip, verify the contents against known data, check the device signature and identify a device from its signature, and perform bit tests and blank checks on devices.

7.3.1 Reading a Device

To read the contents of a device, select the correct part name from the PART menu, as described in section 7.1, and place the device to be read in the programmer socket.

Please note that some devices CANNOT be read, either because the chip type itself is designed not to be read for security purposes, or because the on-chip security features have been enabled when programming.

To read the contents or part of the contents of a device, select

ACTION/READ

The programmer software then quickly accesses the library file for the programming algorithm required. The contents of the device will then be visible in the buffer, and a checksum will be displayed.

Once the contents of a device have been read, they can be edited using the BUFFER/EDIT option, and then saved to disk, or programmed into another device. See section 7.5 for more detail on editing data.

7.3.2 Verifying the contents of a device against a file.

To verify the contents of a device against known data, first select the part type you are using. Load in the file against which you need to verify the data. Place the device to be verified in the programmer and select

ACTION/VERIFY

If the data matches, the message *VERIFY OK* will be displayed. If there is a mismatch, an error message will be displayed.

7.3.3 Checking if a device is blank

In order to test whether a device is empty, select

ACTION/BLANK CHECK

This option checks if device is blank (all 1's or all 0's depending on the device).

Please Note: It is possible for a device to appear blank even though it is not fully erased. If problems are encountered in programming a device, try erasing it for a longer period and reprogramming.

7.4 Programming Logic Devices

IMPORTANT NOTE

Data I/O only use the specified programming algorithms as provided by the device manufacturers. Whilst we make every effort to ensure that programming algorithms are correct, the said algorithms are not guaranteed by the manufacturers and therefore cannot be guaranteed by Data I/O.

Users are advised to ensure that devices work correctly before programming large quantities.

7.4.1 Programming operation

Two different options are provided for programming devices.

ACTION/AUTOPROGRAM

performs a chip erase (for electrically erasable devices), bit test, program and verify, whereas

ACTION/PROGRAM

simply programs and verifies the device.

In normal circumstances when you do not wish to retain any data which may already be in the device, first select the device required, and ensure that the data in the loaded fuse-map is correct. Should the device support on-chip security features, you may want to set the Security feature ON. Do this by selecting:

ACTION/AUTOSECURE

Once this has been turned on, AUTO-SEC is displayed at the bottom of the data buffer. Then choose:

ACTION/AUTOPROGRAM

The programmer will first check the device selection and position, then perform a blank check, followed by an Erase (should the device support electrical erase), then proceed with programming. While a device is being programmed, an activity bar will be displayed in the center of the screen. Following programming, an automatic verify will be conducted and if all is well the message *DEVICE PROGRAMD AND VERIFIED* will be displayed.

7.4.2 Overprogramming devices

If you have a device which already contains some data, and you wish to add data to it, use

ACTION/PROGRAM

The same options are provided as for AUTOPROGRAM. The PROGRAM option programs device with contents of buffer. Program should only be used when programming part of the device while retaining data in other address locations. **In normal circumstances, use AUTOPROGRAM.**

7.4.3 Single key programming

If you need to program more than one device in exactly the same way, simply press <F10> at the end of programming to repeat the process.

7.5 Erasing devices

For GAL, PALCE, PEEL and other electrically erasable devices, use

ACTION/ERASE

This electrically erases the data held on the chip.

7.6 Device Security Features

When a programmable logic device has been secured, it is usually NOT possible to check the security status. The device will generally read Blank. However, with devices whose Security status can be read, select:

ACTION/SECURITY STATUS

To set the security on by default during all programming, select:

ACTION/AUTO-SECURE

7.7 Using the editing facilities

Data to be programmed into programmable logic devices is usually created using a PLD development package such as OPAL™, ABEL™ etc. This takes your logic equations and translates them into a JEDEC file which contains the AND/OR array which will be programmed into the device. In general it will not be necessary to edit this AND/OR array, or fusemap. However, the programmer software provides the option to be able to do this, for example when entering simple designs without a logic compiler, or for educational purposes.

7.7.1 Editing the AND/OR Array

When a device is selected, the buffer is automatically adjusted to the correct size and layout for the chosen device. The AND array and OR array (if applicable) are highlighted separately on the display. Product lines are represented horizontally and OR lines vertically.

Relevant keys:

1 - enters a 1 in the fuse map

0 - enters a 0 in the fuse map

In AND array:

ALT 1: changes the whole horizontal product line to 1

ALT 0: changes the whole horizontal product line to 0

In OR array:

ALT 1: changes the whole vertical OR line to 1

ALT 0: changes the whole vertical OR line to 0

BUFFER/BLANK FUSE MAP

Sets all fuses to 0 and deletes the test vectors.

BUFFER/SET FUSE MAP

Sets all fuses to 1 and deletes the test vectors.

7.7.2 Moving around the buffer

Select

BUFFER/EDIT

This places the cursor in the fuse map. Use the cursor keys to move around the buffer. To go to a specific fuse, select:

BUFFER/GOTO

7.7.3 User's Electronic Signature (UES)

`BUFFER/EDIT UES`

Allows the user to enter the Users Electronic Signature (where relevant). This can be entered as either ASCII or HEX code and the number of bytes depends on the device in use. The UES can be used for stock control, labeling, revision numbers etc. The UES is completely separate from the main array, and in no way affects the operation of a device.

Please Note: Electrically and operationally equivalent devices do not necessarily have the same UES capabilities. For instance, an AMD PALCE22V10 has no UES, a Lattice GAL22V10 does. If your JEDEC file has been compiled for the AMD device, and you are using the Lattice part, a file incompatibility error will be displayed during a `FILE/LOAD` operation. In this instance, the error message can be ignored.

7.7.4 Viewing the Fusemap

To view the contents of the fuse-map without making any changes, select:

`BUFFER/VIEW`

When viewing is complete press `<ESC>` to return

7.7.5 Buffer checksum

The command

`BUFFER/CHECKSUM`

Performs a checksum on the contents of the buffer. This is displayed as two figures:

- The number of '1's in the buffer.
- A simple 2-byte checksum

7.8 Converting PAL Files for GAL Devices

It is possible to use the programmer to convert old files which have been compiled for non-erasable PAL devices to work in GAL devices. To do this, select a `-CONV` device from the menu, load the file, and select:

`BUFFER/CONVERT FUSE MAP`

This will convert the fuse map from the `-CONV` device to the generic device. This is performed automatically if the `-CONV` device is to be programmed.

For example, if an existing design in a PAL16L8 is to be converted into a National Semiconductor GAL16V8 :

1. `ALT+M` (manufacturer)
2. Select `NATIONAL SEMICONDUCTOR`
3. Select `GAL16V8-CONV` from devices list
4. Select `XPAL16L8` from conversion list
5. Open the file using `FILE/LOAD`
6. Select `BUFFER/CONVERT FUSE MAP`

7.9 Test Vector Editing and Operation

Test vectors can be edited or created using the `BUFFER/EDIT TEST VECTORS` command. This will open a window with pin numbers and test vector numbers. Relevant keys are:

0 Forces a logic zero onto pin

| | |
|---|---|
| 1 | Forces a logic one onto pin |
| L | Expect a logic zero on pin |
| H | Expect a logic one on pin |
| C | (Clock) Apply all other signals with C=0 and then apply C=1 followed by C=0 |
| X | don't care |
| Z | Expect high impedance |
| N | Power pin (Vcc or GND). (Sometimes used as don't care but the test vector loader will issue a warning and convert all non power pins to X) |

ALT+T apply the test vectors.

Standard JEDEC Test Vectors can be loaded into the test vector buffer and applied to the device through the ACTION/VECTOR TEST (ALT+T) command. Test vectors are applied to all 40 pins (48 on the ChipWriter programmers) at the same time and are applied automatically as part of the Program/Verify cycle. Any errors are highlighted, displaying exactly which pin in which vector fail. Skew will be determined by the capacitance of the actual device, with only a small skew may causing problems on fast logic devices (faster than 7ns) if asynchronous circuitry is implemented within the PLD.

7.10 Programming Altera MAX devices

All Altera MAX5000, MAX7000x and MAX9000 devices do not use JEDEC format files and are therefore not implemented in PAL.EXE. Separate software enables the programmer to handle the POF and JAM format files which are used on these devices. If you do not have this software, please request it from Data I/O or download it from the web page.

7.10.1 Using POF files

The POF software currently supports certain MAX5000 and MAX7000 devices, namely:

- EPM5016
- EPM5032
- EPM5064
- EPM5128
- EPM7032
- EPM7064-44 (48-pin programmers only)
- EPM7064-68
- EPM7064-84

The software contains drivers for each of the devices supported. To program a device, type in the part name and the file to be programmed, for example:

```
EPM7032 file.pof -pN
```

where N is parallel port number (default is LPT1)

POF file editing is not implemented as the POF format data refers to ELECTRICAL ADDRESSES and DATA. Without recourse to the chip architecture data the information in the POF file is meaningless to the end user.

7.10.2 Using JAM files

The JAM software has been specifically designed to work within ChipWin. No DOS support for JAM files is currently available. The devices supported by the JAM drivers are the complete MAX7000S and MAX9000 families. These are supported either by a range of JAM specific adapters, or using the JAM-Blaster cable for ISP programming.

7.11 Programming Xilinx EPLDs

Xilinx EPLD'S use HEX files rather than JEDEC files, and are not implemented in CWPAL.EXE. Separate software provided with the Installation disks enables the programmer to handle the HEX files that are created by the Xilinx compiler. Each EPLD has a different .EXE file which operates in the following way. The example shown is for the XI7336. All Xilinx software uses the following syntax:

```
XI7336 [FILENAME] [options]
```

FILENAME is the INTEL HEX file generated by the XEPLD development system. No other HEX format file can be used.

The following options are available (XC7336 device has been used for example purposes):

```
XI7336 readfile.hex -READ
```

This can be used to read the Xilinx device in the socket and save the file to disk as an INTEL HEX.

```
XI7336 -BLANK
```

Useful for checking to see if a device is blank. No filename is required.

```
XI7336 vfyfile.hex -VERIFY
```

A verify will be automatically executed after programming but this option may be used to verify the device against an INTEL HEX file.

```
XI7336 prgfile.hex -PROG
```

This is used to program and verify a device from the INTEL HEX file specified. If the security has been set in the file then this will automatically be programmed. If not then a separate command with -SEC may be executed as shown below.

```
XI7336 -DISPSIG
```

This allows the user signature to be displayed. This option does not depend on whether the security has been programmed and can be displayed in either case.

```
XI7336 -DISPSEC
```

Allows you to read and display the security status of the device in the socket.

```
XI7336 -SEC
```

This option allows you to program the security bits on the Xilinx device. Once this has been done only the signature can be read from the device:

If the programmer is not on Parallel Port 1 then the port number can be set using this option. The default is 1. For example, to read from port 2, you should type:

```
XI7336 testfile.hex -READ -P2
```

8. Command Line Batch Software

In addition to the EPROM menu driven software that is provided as standard with the programmer range, DOS command line batch software is also included free of charge. The batch software enables the programmer to be operated using DOS batch commands, thus eliminating the need for the operator to use the menu software, and reducing the margin for error. In addition, time-savings can be made by reducing a series of menu selections to a single command.

The batch software is initialized through the menu software. The procedure is as follows:

1. From DOS, type `CWEPROM <ENTER>`
2. Select the device to be batch programmed (see 6.1)
3. Open the required data file (see 6.2)
4. Program one device. This configures the appropriate config. options and device addresses (see 6.5).
5. Save the environment using `FILE/SAVE ENVIRONMENT` (see 6.2.8). This is stored with an TSK suffix.

At least one device must have been programmed in order for the Environment file to save correctly.

6. Exit to DOS
7. To program a device from the DOS command line, with the selected options, type: `CWEPROM EnvName.TSK -ACTION`

The ACTION options available are:

- PROG
- VERIFY
- ERASE
- BLANK
- BITTEST
- READ
- CHKSUM

Each option requires its own command line.

Please Note: The Batch software will only work using an TSK file configured in the DOS software. The TSK files created in ChipWin do not contain the appropriate parameters for the DOS software

If `-READ` is selected, the file opened within CWEPROM before saving the Environment file will be overwritten with the data read from the device. It is possible to create a blank dummy file that can store all data read from a device. This would also require a new associated Environment file name.

Error levels are reported and can be used within a Batch file to branch to a particular option. For instance, a blank check will report either a blank or non-blank device, the results of this would then determine if the device needs to be erased or not before programming.

Batch software example:

This batch file first blank checks a device, performs an erase cycle if necessary, then programs and verifies the part.

Device selected: AMD 29F040

File Opened: TESTFILE.BIN

Program options: Device address 0 - 7FFFF, Sector Protect = 0

Environment File = AMDTEST.TSK

```
REM  Check that the device is blank
CWEPPROM AMDTEST.TSK -BLANK
If errorlevel=0 go to program
REM  now erase a non-blank device
CWEPPROM AMDTEST.TSK -ERASE
If error level = 1 go to fail
:program
CWEPPROM AMDTEST.TSK -PROG
If error level = 1 go to fail
echo Device Programmed OK
go to end
:fail
echo Programming error
:end
```

9. Using CWCTEST.EXE

CWCTEST.EXE is a utility which allows a whole range of ICs to be tested. These include 74 series and 4000 series logic devices, SRAMS and DRAM's etc. To run the program type:

```
CWCTEST [ P ] [ /NM ] <ENTER>
```

Where P is the parallel port number (default = 1)
and /NM disables the mouse

Functional tests can now be made on the more common 7400 and 4000 series devices as well as on static and dynamic rams. A search facility has also been included.

9.1 Testing a device

To test a device:

1. Select the logic family or device to be tested
2. For logic devices, select the device to be tested
3. Place the device in the ZIF socket and select TEST DEVICE
4. The test vectors for the device will now be applied. If an error occurs then the failed vector will be displayed along with what was actually found when the device was tested. If no errors are found then a message will tell you that the test vectors were OK.

9.2 Identifying a Device

If you are unsure of a device then insert the device into ZIF socket. Select one of the logic families from the main menu (It doesn't matter which family is selected, both the 74 and 4000 vectors will be applied). Select FIND A MATCH. The search will last for about 5 seconds. If all the vectors for a particular device in the library match those of the device being tested then the device number is shown at the bottom of the screen. In some cases there may be more than one match. The software displays all of the devices whose tested vectors match that of the device under test.

9.3 Adding Devices to the User Library

Users can create test vectors to test less common devices. This is done in the same way as defined for CWPAL.EXE (see section 7.9). Ground and Vcc pins can be defined with the G and V descriptors within one test vector. Ground Pins are currently limited (in software) to PIN 24 (Pin 20 on the 40 pin programmers) of the ZIF socket.

To add a new device to chiptest, select:

```
EDIT TEST VECTORS
```

and enter the test vectors required. Then select:

```
ADD TO USER LIBRARY
```

8.4 CWCTEST Restrictions

The CWCTEST software works by applying test vectors to the device to be tested. If the chip passes the tests, the device is assumed good, otherwise it is assumed bad. The search mode works by applying test vectors for all devices in the database and recording all devices that are good.

Herein lies the problem. The chip is only tested at a frequency determined by the rate at which test vectors can

be applied to the device. With the programmer hardware this is typically 500 - 3000 test vectors per second (depending on PC). Although this is much faster than most portable chip testers, it is generally much slower than the target hardware. The chip may pass the test vectors but may fail in the target system. Similarly, output loading may be completely different between system and tester.

Please Note: If you suspect a chip is causing problems in your target system replace it. The only test which is guaranteed is one with a negative result.

Testing of RAM's and LSI devices is slightly different. This is done by the microprocessor inside the tester interfacing to the chip directly. RAM tests are effectively performed on a 1MHz bus. These results should therefore be more reliable. However access times are not measured, hence please note the above warning.

9.5 CWCTEST Example

The CWCTEST software allows simple logic gates to be tested quickly and easily. As an example suppose a batch of 74LS245 devices are suspect. To test the devices run the CWCTEST software by typing:

```
CWCTEST N
```

where N is the parallel port where the programmer is connected (default is 1 if this parameter is missed out). The main screen now appears listing the types of devices that can be tested. In this case, select the 74 series family which is the first selection in the menu. This can be selected by moving the cursor to the correct line and pressing <ENTER> or by double clicking on this selection with the mouse.

A second menu should now appear on the right hand side of the screen. Choose the SELECT DEVICE option. You will now be given the list of devices supported in the 74 series. Move to 74245 by pressing the cursor keys or by using the mouse to move up and down the scroll bar by the side of the window. Once this device has been selected then information you are ready to test the suspect devices.

Select TEST DEVICE from the left hand menu. The test vectors will now be applied. If the device passes all of the test vectors then a message will be displayed on the screen informing you that the test vectors were OK. If the device fails a particular vector test then the failed vector is displayed on the screen. Press a key to return to the left menu. You could also select FIND A MATCH to apply all of the test vectors to the device to see if the device matches with any in the library. The matches are listed at the bottom of the screen. To quit from the left hand menu press <ESC> or select the MAIN MENU option. This takes you back to the right hand menu where you can test a device from another family or choose QUIT to return to the operating system.

Appendix A: Shorthand Keystrokes

For quick operation of Data I/O ChipWriter Programmers the following shorthand keystrokes are available :-

| | EPROM | PAL |
|--------------|----------------|----------------|
| Load | Alt+L | Alt+L |
| Save | Alt+S | Alt+S |
| View | Alt+I or Alt+X | Alt+I or Alt+X |
| Edit Buffer | Alt+E | Alt+E |
| Checksum | Alt+C | Alt+C |
| Fill Buffer | Alt+F | |
| Goto Fuse | | Alt+G |
| Goto Address | Alt+G | |
| Edit UES | | Alt+U |
| Buffer Size | Alt+Z | |
| Read | Alt+R | Alt+R |
| Verify | Alt+V | Alt+V |
| Blank Check | Alt+B | Alt+B |
| OverProgram | Alt+P | Alt+P |
| Autoprogram | Alt+A | Alt+A |
| Query EPROM | Alt+Q | |
| Vector Test | | Alt+T |
| Manufacturer | Alt+M | Alt+M |
| Type | Alt+T | |
| Partname | Alt+N | Alt+N |
| DOS Command | Alt+D | Alt+D |
| Hi-Res Mode | Alt+H | Alt+H |
| Display Mode | Alt+Y | |

Appendix B: Technical Support & Contact Information

Data I/O Customer Support:

For technical assistance, repair, warranty service, contact:

Technical Support—

Hours of operation are from 7:00a to 4:00p Pacific Standard Time (PST). Peak hours are from 11:00a to 2:00p (PST). US Customers in the Eastern Time zones are encouraged to call in the morning when the number of calls is usually the lowest.

Programmers:

Telephone: (800) 3-DATAIO, Press 3

Fax: (425) 867-6898

Sending a fax

Fax the information listed above with your name, phone number, and address, to the Data I/O Customer Support fax number listed previously. If the programmer was purchased through a distributor, please include details of where and when the unit was purchased.

Sending E-mail Technical Support Request

To reach Data I/O Technical Support via e-mail, go to the following URL and fill out the form completely, including Model, Serial Number, Device Type being programmed, and any and all error messages that appear:

<http://www.dataio.com/contact/ems.asp>

This appendix includes details about contacting Data I/O for technical assistance, repair and warranty services, plus how to access and use the Bulletin Board Service and Web page.

Using the Data I/O Web Page

The Data I/O Home Page on the World Wide Web (WWW) includes links to on-line information about technical products, information about Data I/O, a list of International sales channels and distributors, and technical user information such as application notes and device lists.

To access the WWW, you will need an Internet account with Web access, and a Web browser such as Netscape or Internet Explorer. The address of the Data I/O Home Page is:

<http://www.dataio.com>

Once you have reached the web site, select the **CHIPWRITER menu** link or go directly to the ChipWriter software update page at:

<http://www.dataio.com/algorithms/ChipWriter.asp>

NOTE: If you are installing a new version of software, you **MUST** completely uninstall the existing ChipWriter software *before* installing the new version. You may use the standard ChipWriter Uninstall program to do this or use the Add/Remove Programs utility in the Windows Control Panel.

- Select the correct file and download the latest software release
- The software comes as a single self-extracting file. Just run this EXE file to extract all the required installation files.
- For DOS installation, run the INSTALL file, for Windows 9x/NT software, run the SETUP file.

Appendix C: Frequently Asked Questions

- Q1.** Programmer power is connected, but the power light is not lit.
A1. *The power to the programmer is controlled by the software. This means that power is not wasted if the programmer is being run by batteries. The programmer will also power down automatically during periods of inactivity. Try running the Selftest, or running ChipWin, CWEPROM or CWPAL. The light should come on as the programmer is initialized.*
- Q2.** When installing the software in Windows 95/98, the message “system file already in use” is reported, and the installation quits.
A2. *This is caused by one of two system files (MFC42.DLL or MSVCRT.DLL) appearing to the setup file that they are in use, or even newer than that being installed. We provide the latest versions of these files as released on the Microsoft Developer Network collection. To correct this, restart your PC in MS-DOS mode. Go into the windows system subdirectory (CD C:\WINDOWS\SYSTEM). Rename the two files as MFC42.OLD and MSVCRT.OLD. Restart windows, then install the software. There will then be no problems with installation.*
- Q3.** When I am in the ChipWin software, the READ light is grayed out in the operations menu, and I cannot edit the buffer contents.
A3. *This feature is to protect the contents of the buffer from being modified by accident. To enable READ operations, select FILE/ENABLE EDIT MODE.*
- Q4.** When running the software (Selftest, ChipWin, CWEPROM or CWPAL), random communications problems are reported. Printers, and other devices connected to the port have no problems.
A4. *This can be caused by a number of sources. First, check that the programmer is connected properly to the parallel port. Remove any dongles to see if this may be the cause of the problem. Also, check our web page,*

<http://www.dataio.com/algorithms/ChipWriter.asp>

to see if you have the latest software version. The software release notes will indicate if there have been comms problems with any specific device. The programmer works best when connected to an ECP port. On newer PC's, this is selected in the BIOS. Note that on Laptops with an Ir port, selecting IrDA 1.1 could cause ECP mode to become unavailable. Select ECP, NOT ECP/EPP. It is also worth trying the programmer on another PC, or possibly even getting a separate I/O card if the problem persists.

- Q5.** When running ChipWin, the software hangs at the splash screen, or when running CWPAL or CWEPROM in DOS, the PC crashes. Printers, and other devices connected to the port have no problem.
A5. *Again, this may be caused by a number of sources. Check that all the items in A4 have been adhered to. Also, we have seen that adding C:\CHIPWRITER to the PATH statement in the AUTOEXEC.BAT file can resolve this problem, and also ensure that C:\WINDOWS; C:\WINDOWS\SYSTEM and C:\WINDOWS\COMMAND are also in the PATH statement.*

NOTE: For Q4 and Q5, the following may apply.

For certain branded PC's and Notebooks, separate software utilities are available which may solve the port problems. These are required as Compaq, Dell and Gateway, plus others, do not use standard chipsets for their parallel ports, and also include power saving and low voltage features on the I/O ports. Check their web pages for the utilities:

Compaq: <http://www.compaq.com/support/files/index.html>
Dell: <http://www.dell.com/support/index.htm>
Gateway: <http://www.gateway.com/support/product/drivers/index.html>

Guarantee Conditions

1. All programmers are guaranteed for a period of twelve months from the date of purchase, package adapters for 30 days.
2. This guarantee covers both parts and labor.
3. In the event of an item being found to be faulty, Data I/O guarantees to repair or replace the item at its own discretion.
4. Repairs will be carried out on the premises of Data I/O or its agents or distributors at the discretion of Data I/O.
5. No item will be accepted by Data I/O for repair or refund without an RMA number having been issued by Data I/O.
6. All costs of return delivery to Data I/O are the responsibility of the purchaser.
7. Return delivery costs of guaranteed items back to the customer will be met by Data I/O where these are within the US. The method of delivery will be wholly at the discretion of Data I/O.
8. Return delivery outside the US will be the responsibility of the purchaser and must be paid for before delivery will be made.
9. While every endeavor will be made to repair or replace any item as quickly as possible, no guarantees can be made on the time taken and Data I/O cannot be held responsible for any loss or inconvenience caused.
10. Data I/O cannot be held responsible for any loss or damage, consequential or otherwise, incurred while using its equipment.
11. While Data I/O makes every endeavor to ensure that programming algorithms are correct, the said algorithms are not guaranteed by the manufacturers and therefore Data I/O cannot make any guarantees regarding the algorithms implemented on its programmers. Users are advised to ensure that devices work correctly in their systems before programming large quantities.
12. All items except for custom-made items or software are covered by a 30-day money back guarantee if not completely satisfied. Before returning any goods for refund, an RMA number must be obtained. Data I/O reserves the right to make a handling charge .