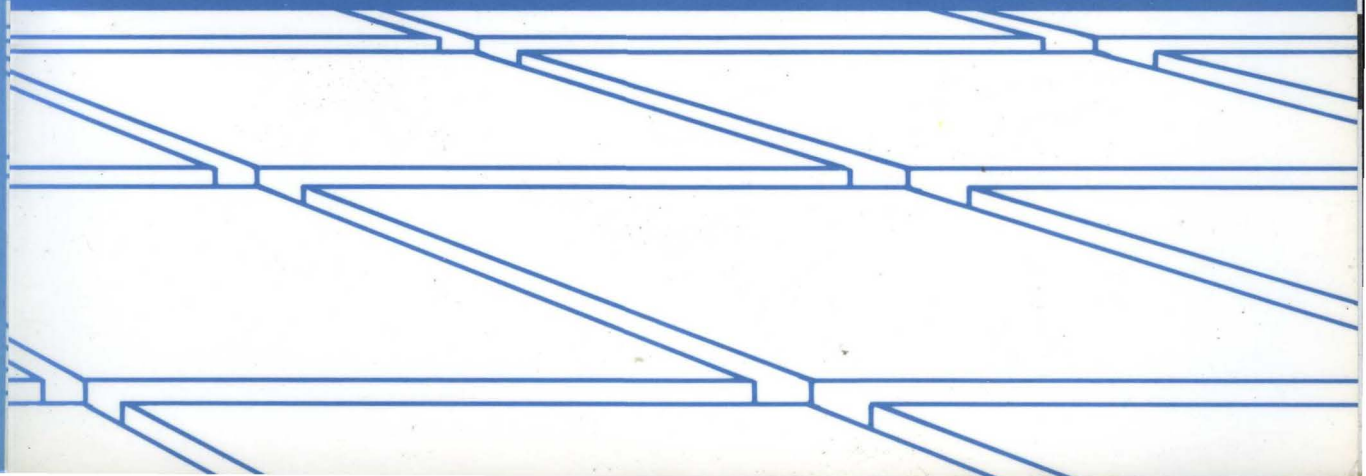
INTERNATIONAL CMOS
TECHNOLOGY, INC.

PEEL™ Software and Applications Handbook



PEEL Software and Applications Handbook

1989

Trademarks

PEEL™ and APEEL™ are trademarks of International CMOS Technology Inc.
PAL® and PALASM® are registered trademarks of Monolithic Memories Inc
GAL® is a registered trademark of Lattice Semiconductor Corp.
ABEL™ is a trademark of Data I/O Corporation
CUPL™ is a trademark of Assisted Technology
Wordstar® is a registered trademark of MicroPro International Corporation

PEEL™ Software and Applications Handbook Copyright

TABLE OF CONTENTS

1.0 Introduction

2.0 Software Overview

3.0 Getting Started with the Software

4.0 A Step-by-Step Guide

5.0 APEEL™ Language Reference

6.0 Application Primer and Examples

Appendix

ICT Sales Network

1.0 Introduction

The PEEL Software and Applications Handbook

Welcome to the PEEL Software and Applications Handbook from International CMOS Technology, Inc. (ICT). This handbook is designed to be used in conjunction with PEEL Device Software included on two floppy diskettes (System diskette and Applications diskette). PEEL Device data sheets may also be useful for reference. Please check to see if you have these items, if not, contact your local ICT sales representative listed in the last section of this manual.

Before using the software, read through chapter 2.0 of the handbook for an overview of the features and operation of the software. Once familiar with its operation, chapter 3.0 contains information for getting started with the software, including installation procedures. To assist in using the software quickly, chapter 4.0 offers a step-by-step guide for executing common functions.

For designing with the APEEL Logic Assembler, chapter 5.0 contains a complete language reference guide, and chapter 6.0 provides an APEEL applications primer as well as several application examples of APEEL source files that are included on the Applications Diskette. Additionally, the Appendix includes useful reference information for using the software and programming PEEL devices.

Please note that although this book deals exclusively with PEEL device software from ICT, PEEL designs can also be implemented using popular third party software products such as ABEL from Data I/O Corporation and CUPL from Logical Devices. Many of the examples and techniques covered in this book will serve as a useful reference when using these software packages with PEEL Devices.

We at ICT hope you find the PEEL Software and Applications Handbook a valuable tool for designing with the PEEL device family. Thank you for your interest in PEEL Devices.

2.0 Software Overview

The PEEL Device Software

The PEEL Device Software provides a PC-based tool for designing with the PEEL Device family of Programmable, Electrically-Erasable Logic devices from International CMOS Technology, Inc. (ICT). Created to serve as a "personal PLD workstation", the PEEL software allows easy access to all operations, from initial design entry to program and test using the PEEL Development System (PDS-1, see appendix B) or by using third-party programmers via the serial port download utility.

The Software consists of:

- A System diskette which contains the PEEL system and installation files.
- An Applications diskette containing several APEEL example files, one PEEL Translator example file, and the PEEL Editor system files.

The software features described in this manual refer to software marked with version number V3.30 or greater

Key Features of the PEEL Device Software

Key features include:

- APEEL™ Boolean Logic Assembler
  - Supports all advanced features of PEEL devices
  - "PALASM®-like" sum-of-products equations
  - "ABEL®-like" macro cell definitions
  - Logic Simulation
- JEDEC file translators for sixty-nine Standard PLDs
- Built-in File Editor
  - To edit source, JEDEC, or test-vector files
  - Interacts with APEEL when assembly errors are found
- Upload/download JEDEC files via the serial port 1 or 2
- Video Terminal Emulator to control third-party programmers within the PEEL software environment
- Expandable and Accessable
  - New features and devices supported with software updates
  - No copy protection

PEEL Device Development Process

The PEEL Device Software includes two unique tools for designing with PEEL devices:

- The APEEL Logic Assembler
- JEDEC File Translator

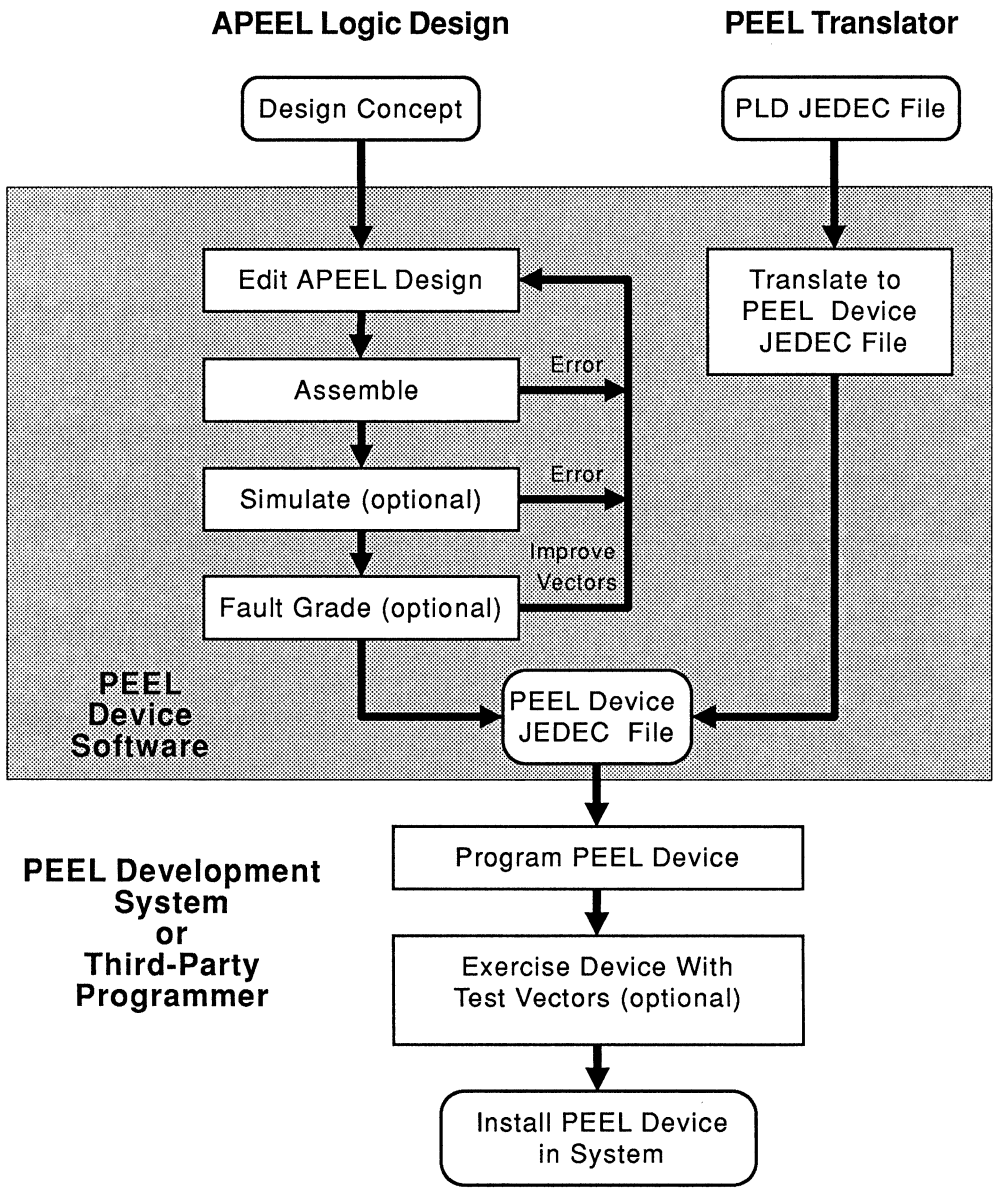
Figure 2-1 illustrates the PEEL development process using the APEEL Logic Assembler or the JEDEC file translator.

The **APEEL Logic Assembler** can be used to create new designs specifically for PEEL devices. The software includes a screen editor for entering and editing your design files. APEEL software then assembles the design file and creates a JEDEC file which can be used to program PEEL devices.

The **JEDEC file translator** allows logic functions created for other PLDs to be programmed into PEEL devices. That is, if you have used some other logic assembler or compiler to develop a design for one of the PLDs which PEEL devices emulate, you can use the translator to program the same function into a PEEL device. The translator reads a PLD JEDEC file and produces a new JEDEC file which can be used to program a PEEL device.

Once a PEEL JEDEC file has been created, a PEEL device can be programmed and secured by using ICT's PEEL Development System (see Appendix B) or by using third-party programmers via the serial port utility.

Figure 2-1. The PEEL Device Development Process



APEEL Logic Assembler

The APEEL Logic Assembler is used to create new designs specifically for PEEL devices. Using the software's built-in text editor or another word processor, APEEL logic equations and test vectors are entered and saved as an APEEL input file (refer to the APEEL Language Reference chapter for a detailed description of APEEL).

Once the design is entered and saved, APEEL software can assemble the file. During assembly APEEL software checks for syntax errors. If errors are found, the APEEL assembler will interact with the editor to open the file and move the editor's cursor to the error (highlighted) location. A brief description of the type of error will also be displayed on the upper right corner of the editor's screen.

When the input file is successfully assembled, its logic functions may be simulated if user-defined test vectors are provided. If simulation errors are found, APEEL will specify the test vector where the error occurred. If there are no simulation errors, a PEEL JEDEC (*.JED) file can be created, or optionally, fault grading may be performed.

Fault grading provides the means for measuring how well the test vectors for the APEEL file cover all possible logic conditions. The fault grader exhaustively exercises the inputs of the simulated files to check for "stuck-at-1" and "stuck-at-0" fault conditions. It then compares the vectors required to do this with the APEEL vectors provided. The final grading is given in percentage of faults covered. The better the percentage of coverage the more likely the test vectors will have tested all possible logic conditions. A report of low fault coverage is an indication to improve the APEEL file test vectors. Because fault grading can be very time consuming, it is optional and recommended only for those who wish to use it.

Once the assembly (simulation and fault grading are optional) is completed, a PEEL device JEDEC file is created and saved to disk. This JEDEC file is now ready to be used for programming a PEEL device. A step-by-step procedure for assembling and simulating an APEEL design file is described in the chapter titled "A Step-By-Step Guide".

These PEEL devices are supported by APEEL V3.30

PEEL18CV8	PEEL20CG10	PEEL22CV10	PEEL22CV10Z
PEEL153	PEEL173	PEEL253	PEEL273

JEDEC File Translator

The JEDEC-file translation utility of the PEEL Device Software translates JEDEC files created for programming other PLDs to JEDEC files used for programming PEEL devices. The translated JEDEC file will program a PEEL device to be a pin-for-pin replacement for the original PLD. The utility translates JEDEC files that have been written on a disk. If a disk file is not available, the pattern may be read from a master device with ICT's PEEL Development System or a third-party programmer. When a third-party programmer is used, the software will allow the file to be uploaded through a serial port and written to a file.

The translated PEEL JEDEC file is given the name of the original file with the ".JED" extension modified to ".JEX". The ".JEX" file can then be used to program your PEEL devices. The complete procedure for translating a PLD file to a PEEL device is described in the section titled "A Step-By-Step Guide".

These devices can be translated to the PEEL18CV8 (V3.30)

PAL16L8	PAL16R8	PAL16R6	PAL16R4
PAL16P8	PAL16RP8	PAL16RP6	PAL16RP4
PAL10L8	PAL12L6	PAL14L4	PAL16L2
PAL10H8	PAL12H6	PAL14H4	PAL16H2
PAL16H8	PAL16LD8	PAL16HD8	PAL18P8
GAL16V8	EP310	EP320	5C031
5C032			

These devices can be translated to the PEEL20CG10, PEEL22CV10, and PEEL22CV10Z (V3.30)

PAL20L8	PAL20R8	PAL20R6	PAL20R8
PAL20L10	PAL20L2	PAL18L4	PAL16L6
PAL14L8	PAL12L10	PAL22V10	PAL20G10
GAL20V8			

☞ The PAL22V10 is JEDEC file compatible with PEEL22CV10, so no translation is actually performed by the software in this case.

These devices can be translated to the PEEL273 (V3.30)

PLS173	PAL20L10	PAL20L8

These devices can be translated to the PEEL173 (V3.30)

PAL20L10	PAL20L8

☞ The PLS173 is JEDEC file compatible with PEEL173, so no translation is actually performed by the software in this case.

These devices can be translated to the PEEL253 (V3.30)

PLS153

These devices can be translated to the PEEL153 (V3.30)

PLS153

☞ The PLS153 is JEDEC file compatible with PEEL153, so no translation is actually performed by the software.

3.0 Getting Started

System Requirements

The following computer system configuration is required for operating the PEEL Device Software:

- IBM PC, XT, AT or compatible
- Minimum 384 K-bytes RAM space
- Monochrome or Color Display
- Two 360K floppy-disk drives, or one floppy disk drive and a hard disk
- DOS version 2.1 or greater

If necessary, both original diskettes can be copied to a 1.2M diskette for single floppy drive operation.

Although the minimum RAM requirement is specified at 384 K-bytes, the PEEL software can still be executed with only 256 K-bytes of RAM. When using the software on a system with only 256 K-bytes of RAM, certain features (listed below) must be executed from DOS by typing the specific file name.

- PEEL Translator (PEELXLT.EXE).
- PEEL Editor (PEELEDIT.COM).

Software Initialization

Floppy-Disk Based Systems

If you are using a computer with only two floppy disk drives, the following procedure must be used the first time you use your PEEL Device Software, or whenever you move your PEEL Device Software to a different computer:

- ➤ Turn computer on and boot-up with DOS 2.1 or greater
- ➤ Make sure you have a back-up of the original diskettes.
- ➤ Insert the system diskette into drive A: and the applications diskette into drive B:
- ➤ At the A: prompt, type **INIT** and press **[ENTER]**.
- ➤ Answer the set-up questions for system-type, color, and sound. Be sure to select **B** for dual floppy-disk operation when prompted for system-type.
- ➤ The software is now ready for operation.

Once initialized for your computer, the software is re-entered with the following procedure:

- ➤ Turn computer on and boot-up with DOS 2.1 or greater.
- ➤ Insert the system diskette into drive A: and the Applications diskette into drive B:
- ➤ At the A: prompt, type **PEEL** and press **[ENTER]**. Note, press **[ENTER]** twice to bypass the PEEL software boot-up messages.
- ➤ The software is now ready for operation.

☞ The system diskette must remain in drive A If you wish to access a file from another diskette, use drive B: (see "Select New Drive, Directory, or Mask" in Step-By-Step Chapter for instructions on changing the file selection screen to select drive B:)

Hard-Disk Based Systems

If you are using a computer with a hard disk, the following procedure must be used the first time you use your PEEL Device Software, or whenever you move your PEEL Device Software to a different computer.

- ➤ Turn computer on and boot-up with DOS 2.1 or greater
- ➤ Make sure you have a back-up of the original disk diskettes.
- ➤ Insert the system diskette into drive A:
- ➤ Type **A:INSTALHD** and press **[ENTER]**. This creates a "PEEL" directory on your hard disk and copies into it all the files from both the diskettes.
- ➤ A prompt will instruct your to insert the PEEL Applications diskette into the drive. Do so and press **[ENTER]**. After copying the files, the PEEL software will boot-up automatically.
- ➤ Answer the set-up questions for system-type, color, and sound. Be sure to select **A** for hard disk operation.
- ➤ The software is now ready for operation.
  After exiting the software, you can leave the PEEL directory by typing **CD** and pressing **[ENTER]**.

Once the PEEL Device software is installed in your hard disk, the software can be started with the following procedure:

- ➤ Turn computer on and boot-up with DOS 2.1 or greater.
- ➤ Change to the PEEL directory by typing **CD\PEEL** and pressing **[ENTER]**.
- ➤ Type **PEEL** and press **[ENTER]**. Note, press **[ENTER]** twice to bypass the PEEL software boot-up messages.
- ➤ The software is now ready for operation. After exiting the the software, you can leave the PEEL directory by typing **CD** and pressing **[ENTER]**.

The "Main Menu"

The first screen to appear after the initialization and sign-on messages will be the "Main Menu" which contains the majority of the commands used with the PEEL Device Software Kit.

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

The main menu is divided into three command sections: **DEVICE**; **JEDEC FILE**; and **UTILITIES**.

The **DEVICE** section controls all operations pertaining to the PEEL device programmer module. The commands include: **Load**; **Program**; **Verify**; **Test**; and other auxiliary functions. These commands are provided only to illustrate the capabilities of the PEEL Development System.

The **JEDEC FILE** section controls the primary operations used for working with JEDEC files. These functions include: **Read from disk**; **Write to disk**; **COM transmit/receive**; **Display/print**; **Translate**; and **Check sum**.

The **UTILITIES** section allows access to the functions: **Help**; **Setup**; **File/directory**; **Edit**; **APEEL**; **VT Emulation**; and **Quit to DOS**.

Command Selection

To select a command from the main menu, simply press the letter that corresponds to its adjacent command word. Some commands will have secondary command menus or prompts which must be followed to complete the original command.

"H" is for Help!

When in doubt about the operation of a command, press **H** for help. This will display an on-line help file that contains a complete list of the PEEL Device Software commands, each with a brief description of its operation. After selecting **H**, any command letter can be pressed to get help on that specific command, or press **H** to present the entire help file.

Exiting And Re-entering The Software

Exiting the PEEL software is done from the main menu by pressing **Q** for **Quit to DOS**. Doing this will cause a prompt to appear asking you to confirm that you wish to quit. If you respond by entering **Y**, the software will terminate and the DOS prompt will appear.

To re-enter the software from DOS, simply type **PEEL** and press **[ENTER]**. (Press **[ENTER]** twice to bypass the PEEL software boot-up messages.)

4.0 A Step-by-Step Guide

In this section, the step-by-step procedures illustrate how to use some of the major main menu commands. Please note that you must begin each of the step-by-step procedures at the main menu prompt.

Assemble/Simulate an APEEL File

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]      JEDEC FILE [R8CNTR.JEX]       UTILITIES

L = Load                R = Read from disk            H = Help
P = Program             W = Write to disk             S = Setup
V = Verify              C = COM Transmit/receive      F = File/directory
T = Test                D = Display/print             E = Edit
U = aUxiliary           X = Translate                 A = APEEL
                        K = checKsum [A8A0]           Z = VT Emulation
                                                      Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

1. Press "A" for APEEL

The "Assemble File" selection screen will appear. Several demonstration APEEL (.APL) files provided on the Applications diskette will appear. Use the arrow keys followed by **[ENTER]** to select the APEEL (.APL) file you wish to assemble. If you don't see the file you wish to assemble, refer to "Select New Drive, Directory, or Mask" found later in this section.

Use V8GATES.APL if this is the first time you are using the assembler.

2. Watch the Assembly Process

A description of the assembly process which converts the APEEL file to the PEEL JEDEC file will be displayed. Also, this description which includes the macro cell definitions (if applicable to the PEEL device), pin assignments, and product term assignments is written to a file with the same file name as the source file but with an ".OUT" extension.

☞ By pressing **[Ctrl] S**, you can start or stop the scrolling of the display. Additionally, pressing **[Ctrl] C** will terminate the assembly (or simulation) process.

If a syntax error is found, the assembly will stop and the APEEL file line number with the error will be displayed. The editor may then be entered to correct the file. Press **[space bar]** to enter the editor, or press **Q** to terminate the assembly and return to the main menu. Once the editor has opened the file, the cursor automatically goes to the error line which will also be highlighted. A brief description of the error and its location is also displayed in the upper right corner of the screen. When you return from the editor, the PEEL software will prompt you to re-assemble the file.

If no syntax errors are found, the assembly will conclude and the following prompt will appear:

```
Assembly Completed...S=Simulation...Q=Quit..any other key to continue
```

Note: Simulation is performed only if the APEEL file has test vectors.

☞ If you press **Q** (for Quit) after assembly, simulation or fault grading, the APEEL process will terminate and the PEEL JEDEC file will not be created nor saved to disk. If you want the JEDEC file to be created and saved to disk but do not want simulation, you must press any key other than "S" or "Q" (such as the space bar).

3. Press the "S" for Simulation (optional)

If there are no test vectors present in the file, APEEL will omit the simulation and fault grading. However if test vectors are

provided, they can be used to verify the function of the APEEL design through simulation. If simulation is selected, each vector will be displayed one at a time.

If a simulation error is found, an error message will appear prior to the vector that fails. An example of the error message is shown below.

ERROR! vector# 8 pin=19 sim=L vec=H

The above error message means that the vector 8 in the APEEL file has a simulation error on pin 19. The simulated level for this output is a LOW, but the expected output level (specified in the file) is a HIGH.

If no simulation errors are found, simulation will conclude and the following prompt will appear:

```
Simulation Completed..F=Fault Grade..Q=Quit..any other key to continue
```

The prompt shown above assumes that the Fault Grade mode has been turned ON (the default condition is OFF) in the Setup utility. If you wish to perform fault grading, press **S** (for Setup) at main menu prompt. Then, press **F** to set the Fault Grade mode to ON.

4. Press "F" for Fault Grade Test Vectors (optional)

Fault grading provides the means for measuring how well the test vectors for the APEEL file cover all possible logic conditions. The final grading is given as a percentage of Stuck-At-One (SA1) and Stuck-At-Zero (SA0) faults covered. The higher the percentage of coverage the more likely the vectors will have tested all possible logic conditions. A very low percentage is an indication that the test vectors need to be improved.

Fault grading can be very time consuming. You can press the space bar during the fault-grading process to get a report on the progress of the fault grading, or you can press **Q** to terminate the process.

5. Press any key to continue

Regardless of whether simulation or fault grading was used, pressing any key after proper assembly will allow the JEDEC file to be created and saved to disk. The JEDEC file will then be displayed and loaded into memory. This JEDEC file will be given the same name as the APEEL source file but with the extension changed to ".JED". The JEDEC file can then be used to program a PEEL device.

During assembly, simulation, or fault grading, a description of the execution process is displayed. For APEEL designs which many product terms, the time consumed for displaying this execution process can be significant. There is a feature in the Setup utility which will turn off the display mode and speed-up the assembly process. Press **S** (for Setup) at main menu prompt. Then, press **B** to set the Brief mode to ON (default condition is OFF).

Translate a PLD File to a PEEL Device

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

4

1. Press "X" for Translate JEDEC file

A prompt will appear on the screen:

```
D = translate Device loaded in memory
F = translate File on disk
Q = Quit

Enter Selection >
```

2. Press "F" to translate File on disk

The "Translate File" selection screen will appear showing the files found in the directory currently specified. If you don't see the file you wish to translate, refer to "Select New Drive, Directory, or Mask" found later in this section.

(Note: the selection "D = translate Device loaded in memory" is used when translating programming patterns that have been loaded directly from master devices with the PEEL Development System hardware.)

3. Use the arrow keys to select the file to be translated

Note that a translation demonstration file, **R8CNTR.JED**, is provided on the PEEL Applications Diskette. **R8CNTR.JED** is a PAL16R8 JEDEC file for an 8-bit counter.

4. Press the [ENTER] key to select file

A list of the target PEEL devices for the translation will appear:

```
A = 18CV8
B = 253                 F = 153
C = 273                 G = 173
D = 22CV10Z             H = 22CV10
E = 20CG10

Select target PEEL device >
```

Note that you can terminate the translation process and return to the main menu by pressing **[Esc]**.

5. Press the letter of target PEEL device

If you are translating the demonstration file, **R8CNTR.JED**, be sure to type **A** to select the PEEL18CV8.

6. Press the letter of source PLD

A list of PLDs that can be translated to the selected target PEEL device will appear. If translating the demonstration file, **R8CNTR.JED**, select the letter **B** for 16R8. Otherwise, select the letter corresponding to the part number of the PLD JEDEC file being translated.

Once you have made your selection, a prompt will appear to confirm your choice of the target PEEL device and source PLD.

```
Source JEDEC File: R8CNTR.JED

Source Device                    Target Device
PAL16R8      ————————————————»  PEEL18CV8

Do you want to change the device type? (Y/N) >
```

If you press **Y**, you will be asked to repeat steps 5 and 6. If you do not need to change your selection, press any other key.

7. Watch the translation process

The translated and formatted PEEL JEDEC file will then be displayed on the screen, updated in memory, and written to the currently selected drive. The file written will have the same name as the original JEDEC file with the extension modified to ".JEX" to designate a translated file. The ".JEX" file can then be used to program a PEEL device.

Turning on the Brief mode (default condition is off) in the Setup utility will turn the display function off to speed-up the translation process.

Translate a PLD File to a PEEL Device (from DOS)

When running the PDK-1 software on a system with only 256 K-bytes of RAM the translator utility cannot be used within the PEK-1 environment. However, the translator utility can be executed directly from DOS. The PEEL device translator program is contained in a file called PEELXLT.EXE on the program diskette. When using PEELXLT.EXE, you must specify a PLD source file name, a source PLD type, and a target PEEL device type.

Type "PEELXLT *source_filename source_PLD target_PEEL_device*"

You need not specify the extension for the PLD source file name because it is assumed to have a ".JED" extension. The source PLD and target PEEL device types recognized by the translator are listed below.

An example of the PEEL Translator command format is:

```
PEELXLT R8CNTR 16R8 18CV8
```

Here, the JEDEC file "R8CNTR.JED" is used in the translation from the PAL16R8 to the PEEL18CV8 device, and in the process creates the PEEL device JEDEC file "R8CNTR.JEX".

When translating to the PEEL18CV8

Target PEEL device: 18CV8
Source PLDs:

16L8	16R8	16R6	16R4	
16H8	16P8	16RP8	16RP6	
16RP4	16LD8	16HD8	18P8	
16L2	16H2	14L4	14H4	
12L6	12H6	10L8	10H8	
310	320	5C031	5C032	16V8

When translating to the PEEL20CG10, PEEL22CV10, or PEEL22CV10Z

Target PEEL device: 20CG10, 22CV10, or 22CV10Z
Source PLDs:

20R8 20R6 20R4 20L8

20L10 20L2 18L4 16L6

14L8 12L10 20G10 20V8

22V10 (20CG10 and 22CV10Z only)

☞ No translation is necessary for PAL22V10 to PEEL22CV10 because they are JEDEC-file compatible.

When translating to the PEEL273

Target PEEL device: 273
Source PLDs:

PLS173 20L10 20L8

When translating to the PEEL173

PEEL device: 173
Source PLDs:

20L10 20L8

☞ No translation is necessary for PLS173 to PEEL173 because they are JEDEC-file compatible.

When translating to the PEEL253

Target PEEL device: 253
Source PLD:

PLS153

☞ No translation is necessary for PLS153 to PEEL153 because they are JEDEC file compatible.

The translator program also provides the capability of translating multiple PLD JEDEC files to different types PEEL device JEDEC files through the application of a DOS batch file. Here is an example of the batch file (PAL2PEEL.BAT).

```
PEELXLT R8CNTR 16R8 18CV8
PEELXLT DECODE 20L10 22CG10
```

Typing **PAL2PEEL** followed by **[ENTER]** will automatically translates the R8CNTR.JED and DECODE.JED to their respective target PEEL device JEDEC files (R8CNTR.JEX and DECODE.JEX).

Read a JEDEC File from Disk

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >


------------------------------------------------------------------------

------------------------------------------------------------------------
```

4

1. Press "R" to Read JEDEC File

The "Read File" selection screen will appear showing the files found in the current directory with the specified file mask.

2. Use the arrow keys to locate the JEDEC file

If the file you wish to read is not displayed, refer to "Select New Drive, Directory, or Mask" found later in this section.

3. Press [ENTER] to read the selected file

The PEEL device JEDEC file will be read and displayed on the screen. When you respond to the prompt "any key to continue", the PEEL software will display the name of the file that has been read and the PLD or PEEL device type that has been selected for this JEDEC file. If the selection is correct, press **C** to continue and the main screen will appear. If you press any other key, you will be asked to choose a new device type. The main screen appears once a new device has been selected.

Write (Save) a JEDEC File to Disk

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

1. Press "W" to Write JEDEC File

The "Write File" selection screen will appear showing the files found in the directory with the file mask currently specified.

2. Use the arrow keys to locate the JEDEC file (for over-writing only)

If you wish to over-write an existing file and this file is not displayed, refer to "Select New Drive, Directory, or Mask" found later in this section.

If you wish to enter a new file name, press **W** and then type in the new file name followed by **[ENTER]**. After the file has been written to disk, the PEEL software will return to the main menu (bypassing step 3).

3. Press [ENTER] to over-write to the selected file

The PEEL software will prompt you to confirm over-writing to the selected file. Press **Y** to over-write the selected file (press **N** to cancel the command).

Download/Upload JEDEC File via the COM Port

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```


This feature allows you to transfer PEEL device JEDEC files between the PC and a third-party programmer.

1. Press "C" to select COM transmit/receive mode

The COM port parameter selection screen shown below will appear.

```
C = COM Port          : COM1          X = Transmit JEDEC to COM
B = Baud-rate         : 4800          R = Receive JEDEC from COM
D = Data size         : 8             Q = Quit to main menu
S = Stop bits         : 1
P = Parity            : NONE
T = Timeout           : 30

Enter Selection >
```

2. Set the port parameters by pressing the appropriate letters

The port parameter value selection is made by pressing each letter corresponding to the port parameters continuously until the desired parameter value appears.

3. To upload a PLD JEDEC file from a third-party programmer to memory:

3a. Press "R" to Receive JEDEC FILE

The PEEL software will wait for the third-party programmer to transmit the JEDEC file. If no JEDEC file is transmitted within the period specified by the timeout value (in seconds), the COM Receive mode will terminate.

☞ You can terminate the JEDEC file reception or transmission any time by pressing **[Esc]**.

3b. Set the third-party programmer to Transmit JEDEC file

When transmission is complete, the PEEL software will display the name of the file (COM.JED) that has been read and the PLD or PEEL device type that has been selected for this JEDEC file. If the selection is correct, press **C** to continue and the main menu will appear. If you press any other key, you will be asked to choose a new device type. The main menu appears once a new device has been selected.

☞ The uploaded JEDEC file resides in the memory only and has not been saved to disk yet.

4. To download a JEDEC file from memory to a third-party programmer:

4a. Set the third-party programmer to Receive mode

The PEEL software is equipped with a video terminal emulator (refer to "Using The VT Emulator" found later in this section) so that you can set your third-party programmer to "receive" mode without exiting the software. (Be sure to set your third-party programmer to the correct PEEL device type.)

4b. Press "X" to Transmit JEDEC file to COM

If the transmission is successful, press any key to return to the main menu.

☞ The PEEL software will flag any reception or transmission errors due to:

- COM port parameters incorrectly set.
- Incompatible PEEL device type between the PEEL software and the third-party programmer.

Display/Print a Formatted PEEL JEDEC File

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]      JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load                R = Read from disk           H = Help
P = Program             W = Write to disk            S = Setup
V = Verify              C = COM Transmit/receive     F = File/directory
T = Test                D = Display/print            E = Edit
U = aUxiliary           X = Translate                A = APEEL
                        K = checKsum [A8A0]          Z = VT Emulation
                                                     Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

With this feature, you can display or print the JEDEC file that currently resides in memory. If you want to display or print a file from disk, you must first read the file into memory (refer to "Read a JEDEC file from disk").

☞ The default setting for the printer port is LPT1. To select a different printer port, use the Setup utility. Press **S** at the main menu prompt, and then press **P** to change the printer port.

1. Press "D" to Display/print the JEDEC file

A prompt will appear on the screen:

```
D = list JEDEC to Display
P = list JEDEC to Printer
Q = Quit to main menu

Enter Selection >
```

3. If "D" is pressed

The currently loaded PEEL device JEDEC file will be displayed on the screen, 24 lines at a time. To display the next full screen, press the space bar or any key other than "D" or "Q". To exit the display mode, press **Q**. To display the entire file without pausing, press **D** again.

☞ **[Ctrl] S** and **[Ctrl] Q** can also be used to start and stop the scrolling of the display. [Ctrl] C will exit the display mode.

4. If "P" is pressed

The currently loaded PEEL device JEDEC file will be output to the printer port. Make sure your printer is on-line and ready to receive data. The main menu will appear when printing is complete.

Using The Built-in File Editor

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]      JEDEC FILE [R8CNTR.JEX]       UTILITIES

L = Load                R = Read from disk            H = Help
P = Program             W = Write to disk             S = Setup
V = Verify              C = COM Transmit/receive      F = File/directory
T = Test                D = Display/print             E = Edit
U = aUxiliary           X = Translate                 A = APEEL
                        K = checKsum [A8A0]           Z = VT Emulation
                                                      Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

4

1. Press "E" for Edit

The "Edit File" selection screen will appear. Use the arrow keys to select the file you wish to edit and then press **[ENTER]**. (If the file you wish to edit is not displayed, refer to "Select New Drive, Directory, or Mask" found later in this section.) The selected file is loaded into the editor which appears on the screen. Notice the command window in the upper right corner. The command window will display how and when to enter or exit the pull-down menu.

2. Press the [F10] key, then [ENTER] for pull-down menu

The first pull-down command menu will appear. Use the right and left arrow keys to select the command type. You can use the up and down arrow keys to select the specific command, then press **[ENTER]** to invoke the command.

3. Press the [Esc] key twice to exit pull-down menu

This will cause the pull-down windows to disappear and return the editor to edit mode. When in the edit mode, the standard cursor key functions and WordStar®-type **[CTRL]** commands

can be used. For further information on the editor commands, refer to Apendix A or any reference book for the WordStar-type editors.

4. To exit Editor, press [F10], then "F" , then "Q"

If any changes were made, a prompt will appear for saving changes. If you wish not to save the edits, press **N** . The main menu will then appear.

You can also use **[CTRL] K [CTRL] Q** (during edit mode only) to exit the editor.

Using The File/Directory Utilities

```
-------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
-------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >



-------------------------------------------------------------------------

-------------------------------------------------------------------------
```

4

1. Press "F" for File/directory

The "File/directory" selection screen will appear. The **create directory**, **rename directory**, **rename file**, and **erase file** commands are located in the upper left corner of the screen. (Please refer to "Select New Drive, Directory, or Mask" found later in this section for changing directory or file mask.)

If you are erasing or renaming a file, use the arrow keys to select the file, and then go to step 3.

2. Press "C" to create a directory, or press "R" to remove a directory

The PEEL software will prompt you to enter the directory name. You might need to use the **[Back Space]** key to delete the file or directory name selected by the file selection screen cursor. If the operation is successful, you will be returned to the "File/directory" selection screen. Error messages will be displayed if the operation is unsuccessful.

☞ The directory created or removed must be a sub-directory of the current directory. Please refer to "Select New Drive, Directory, or Mask" for changing the directory.

3. Press "E" to erase a file, or press "N" to rename a file

If you press **E** to erase a file, a prompt for you to enter the file name appears. Most probably, a file name (selected from the file selection screen cursor) has already been typed in at this prompt. If the file name is correct, press **[ENTER]**. If the file name is incorrect, use the **[Back Space]** key to delete the file name, and then enter the correct file name followed by **[ENTER]**. A confirmation prompt will appear.

If you press **N** to rename a file, the file name selected from the file selection screen cursor appears at the "old file" prompt. If the file name is correct, press **[ENTER]**. If the file name is incorrect, use the **[Back Space]** key to delete the file name, and then enter the correct file name followed by **[ENTER]**. When the "new file" prompt appears, type in the new file name followed by **[ENTER]**.

If the operation is successful, the PEEL software returns you to the "File/directory" file selection screen. Error messages will be displayed if the operation is unsuccessful.

Select New Drive, Directory, or Mask

The file selection screen which shows the files of the current directory with the specified file mask is used by the Read from disk, Write to disk, Translate (from file on disk), File/directory, Edit, and APEEL utilities. A menu of file selection options appears in the upper right corner of the screen.

Press "D" to change drives.

If you wish to display a different drive, press **D**. Type in the letter of the new drive (do not press **[ENTER]** after the drive letter).

Press "I" to specify a different directory

If you wish to display the files of a different directory, press **I**. Backspace over the current directory name and type the new name followed by **[ENTER]**. For example, typing **PEEL\DESIGNS** would select the sub-directory **DESIGNS** in the directory **PEEL**.

You may have noticed that all sub-directories in the current directory are displayed with a "/" at the beginning of each directory name. You can actually change directory by moving the file selection cursor to directory and then press **[ENTER]**. The "/." and "/.." represents the root and the next higher directory respectively. So, you can change your directory to the root directory if you select "/." and then press **[ENTER]**.

Press "M" for change file mask

During file selection, the PEEL software uses a mask to control which files will be displayed. If you wish to change the mask, press **M**, then backspace over current mask and type the new mask to be used. Note that a " * " is a wild card. For instance, a " ***.JE*** " will show all files with extensions that consist of the " **.JE** " (i.e., .JED and .JEX files). A " *.* " overrides the mask and shows all files. The mask defaults to " ***.A*** " in the Edit and APEEL directories in order to select the APEEL (*.APL) files.

Select File

Once the desired drive, directory, and mask is set, the file can now be selected. Use the arrow keys to highlight the desired file and press **[ENTER]**.

6. Press "Q" for Quit

If you wish not to select a file for read, write, translate, and etc., press **Q** to return to the main menu.

Using The VT Emulator

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

This feature allows you to exercise a video terminal emulation so that you can control the functions of your third-party programmer (if applicable). Programming features such as program, load, and verify can then be initiated without exiting the PEEL software.

1. Press "Z" to select VT Emulation

The COM port parameter selection screen (similar to that of the COM transmit/receive utility) will appear.

```
C = COM Port          : COM1
B = Baud-rate         : 4800          T = Terminal Emulation
D = Data size         : 8             Q = Quit to main menu
S = Stop bits         : 1
P = Parity            : NONE
T = Timeout           : 30

Enter Selection >
```

2. Press the letter corresponding to the port parameters

The port parameter value selection is made by pressing each letter corresponding to the port parameters continuously until the desired parameter value appears.

Changing the port parameter values here will affect the port parameter values set in the COM transmit/receive utility, and visa versa.

3. Press "T" to start terminal emulation

You can terminate the terminal emulation any time by pressing **[Esc]**. When you press **[Esc]** to abort, the ASCII character for the [Esc] key would not be sent to the COM port. Some third-party programmers will end terminal or remote mode if they receive the [Esc] ASCII character. In this way, you can exit the terminal emulation mode without terminating your third-party programmer's terminal or remote mode. One application example is the downloading of the PEEL device JEDEC file from memory to your third-party programmer via the COM port. Here, you use the terminal emulation to set the programmer to receive mode and then return to the main menu to execute the COM Transmit utility.

Setup Utilities

```
------------------------------------------------------------------------
                      ICT PEEL DEVICE SOFTWARE          Version 3.30
------------------------------------------------------------------------
DEVICE [PEEL18CV8]     JEDEC FILE [R8CNTR.JEX]      UTILITIES

L = Load               R = Read from disk           H = Help
P = Program            W = Write to disk            S = Setup
V = Verify             C = COM Transmit/receive     F = File/directory
T = Test               D = Display/print            E = Edit
U = aUxiliary          X = Translate                A = APEEL
                       K = checKsum [A8A0]          Z = VT Emulation
                                                    Q = Quit to DOS
Enter Selection >



------------------------------------------------------------------------

------------------------------------------------------------------------
```

Additional features of the PEEL Device Software can be found under the **Setup** utility heading.

1. Press "S" for Setup Utility

Allows you to configure the operation of the system to suit your preferences or the requirements of your computer. Note that most the Setup configurations are saved to the "PEEL.SAV" file. Every time you call the PEEL software by typing **PEEL** followed by **[ENTER]**, the previous Setup configuration will be implemented. If you type **[INIT]**, the default configuration will be implemented.

```
C = Color mode is ON              S = Sound mode is ON
T = Timer is OFF                  B = Brief mode is OFF
D = Directory sort is ON          V = Vector trace mode is OFF
F = Fault Grade is OFF            A = APEEL output file mode is ON
R = Reset Program Counter         P = Printer Port is LPT1
Q = Quit to last menu

Enter Selection >
```

Color Mode

Pressing **C** will toggle the color mode from ON to OFF, and visa versa. Turn color mode on for color monitors. Turn color mode off for monochrome monitors or to display in black and white on color monitors.

Sound Mode

Pressing **S** will toggle the sound mode from ON to OFF, and visa versa. When the sound mode is on, your PC will produce various sounds as different functions are executed. When the sound mode is off, no sounds will be produced.

Timer Mode

Pressing **T** will toggle the timer mode from ON to OFF, and visa versa. During programming, when the timer mode is on, the system will measure and report on the time it took to program, verify, and test the PEEL device (applicable only for PEEL Development System). The default condition is OFF.

Brief Mode

Pressing **B** will toggle the brief mode from ON to OFF, and visa versa. When brief mode is off, the system will display the JEDEC or APEEL file being read, translated, or assembled. Turning brief mode on will disable this display feature and speed up the operation of the function. The default condition is OFF.

Directory Sort

Pressing **D** will toggle the directory sort mode from ON to OFF, and visa versa. When directory sort is on, files displayed in the file selection screen will be sorted in alphabetical order. This sorting process could take up to 2-3 seconds if there are a lot of files in the directory. The default condition is ON.

Vector Trace Mode

Pressing **V** will toggle the vector trace mode from ON to OFF, and visa versa. When vector trace mode is on, the simulator will report in detail the results of each pass for each vector through

the simulator. A vector pass is similar to a vector state. The default condition is OFF.

Fault Grade Mode

Pressing **F** will toggle the fault grade from ON to OFF, and visa versa. The default condition is OFF.

APEEL Output File Mode

Pressing **A** will toggle the APEEL output file mode from ON to OFF, and visa versa. When the APEEL output file mode is on, the description displayed on your screen during the APEEL process will be written to a file with the same name as your source file but with a ".OUT" extension. This is useful when you need a print-out copy of you simulation errors. The default condition is ON.

Reset Program Counter Mode (PEEL Development System only)

Pressing **R** will reset the program counter.

Printer Port Selection

Pressing **P** will allow you to select any one of these printer ports: LPT1, LPT2, and LPT3. The default port is LPT1.

5.0 APEEL™ Language Reference

Using the APEEL™ Logic Assembler

The APEEL logic assembler is used for creating custom logic functions for implementation in PEEL devices. Logic designs are specified as Boolean equations with additional statements specifying the desired configurations of the I/O macrocells. Input files may be created with most any text editor or word processor, including in the APEEL software package is a built-in word processor which is driven by familiar keyboard commands and convenient pull-down menus.

APEEL software operates in an interactive manner during assembly. It assembles the input file while checking for syntax errors. If errors are found, the APEEL assembler will interact with the PEEL editor to open the input file and moves the cursor to the line with the error (this line is highlighted). A brief description of the error and its location is also displayed in the upper right corner of the editor screen. When the input file is successfully assembled, user-defined test vectors may be used to simulate the logic function created by the design. Simulation errors are reported and identified to aid in debugging. APEEL software can also provide fault grading of the test vectors.

When the design is successfully assembled, APEEL software creates a JEDEC fusemap with JEDEC-formatted test vectors (if applicable). The JEDEC file can then be used by the PEEL Development System programmer or downloaded to a third-party programmer via the COM tramsmit/receive utility for programming PEEL devices.

For more information on using the APEEL logic assembler, refer to chapter 4.0, "A Step-By-Step Guide".

Elements of the APEEL™ Source File

APEEL™ Source File Template

You can create APEEL source files with any word processor or text editor that produces ASCII files. The elements of an APEEL source file are shown in the Figure 5-1. In the figure, lines which begin with double quotes are comments and are ignored by the APEEL assembler. The expressions in **bold type** are the APEEL reserved words which must be included in the file. The expressions in *italic type* are strings that you would replace with the appropriate pin name, logic expression, etc. The application examples in chapter 6 further illustrate the elements of an APEEL source file.

```
TITLE 'title of design'

" device declaration

        PEEL_device_number

ZERO_POWER    "only applicable for PEEL22CV10Z
AUTO_SECURE   "Sets security bit automatically

" pin and node assignments

        name   PIN     pin_#

        name   NODE    node_#

EQUATIONS

            name  =  Sum-of-Product  expression

   ENABLE  name  =  Product  expression

TEST_VECTORS

( input1  input2  ->  output1  output2 )
  input1  input2  ->  output1  output2
  input1  input2  ->  output1  output2
```

Figure 5-1. APEEL Source File Template

The elements of an APEEL source file are:

TITLE – The title is optional and is used to describe the source file. The title will be printed as a header when printing the JEDEC programming file and is useful for documenting the design. The title statement consists of the keyword **TITLE** followed by a string enclosed by single quotes ('), where the string is the desired title of the design. The maximum length of the title is 240 characters (3 lines with 80 characters wide).

DEVICE DECLARATION – The device declaration identifies the target PEEL device for the assembler. Use the ICT PEEL part number as the device identifier, eg. **PEEL18CV8**, or **PEEL173**, etc.

ZER0-POWER - If the Zero-Power option which is applicable only for PEEL22CV10Z is desired, you must enter the reserved word **ZERO_POWER** after the device declaration but prior to pin and node assignments.

AUTO_SECURE - This feature allows the security bit to be automatically set when the JEDEC file is created this reserved word must be specified after the device declaration but prior to pin and node assignments.

PIN AND NODE ASSIGNMENTS – This section defines the pins and nodes of the PEEL device with identifiers used in the source file. User-defined names are assigned to each pin and the configurations of the I/O macro cells are defined. Refer to the APEEL Syntax section for more details.

Note that node assignment statements are only applicable for devices which have Asynchronous Clear or Synchronous Preset (or both) product terms.

EQUATIONS – This section contains the Boolean equations that describe the logic design. Refer to the APEEL syntax section for details on the Equations section of the source file.

TEST VECTORS – Contains the optional user-defined test vectors. Test vectors can be used by the APEEL assembler's logic simulation function to verify the functionality of the logic design. The vectors are also used by the PEEL Development System or third-party programmers to exercise the PEEL device after it has been programmed.

Listing 5-1 is the APEEL source file for the Basic Registers and Latches application of the PEEL18CV8. This file is called "V8REGS.APL" and is found on the PEEL Device Software Applications Disk. Please study this listing carefully as it contains numerous information on the APEEL assembler.

```
TITLE 'APEEL FILE:18CV8 BASIC REGISTERS and LATCHES,
DESIGNER:Robin Jigour, ICT
DATE: 8/16/87'

PEEL18CV8



"Inputs"

CLK         pin 1
D           pin 2                        "register and latch inputs
T           pin 3
J           pin 4
K           pin 5
R           pin 6
S           pin 7
LAT         pin 8                        "gated latch
LEN         pin 9                        "enable for gated latch
SRES        pin 11                        "synchronous reset
SSET        pin 12                        "synchronous set
ARES        pin 13                       "asynchronous reset


"Outputs and Macro Cell definitions

Q_GL       pin 14 = pos com feed_or      "Latch outputs
  "(internal feedback)
Q_SL       pin 15 = neg com feed_pin
Q_SR       pin 16 = pos reg feed_reg     "Register outputs"
Q_JK       pin 17 = pos reg feed_reg
Q_T        pin 18 = pos reg feed_reg
Q_D        pin 19 = pos reg feed_reg


"Internal Nodes"

 AC         node 21                    "Asynchronous Clear node"
 SP         node 22                    "Synchronous Preset node"


EQUATIONS

SP = SSET                           "Synchronous Preset
AC = ARES                           "Asynchronous Clear
```

Listing 5-1 APEEL Source File For V8REGS.APL

```
Q_D   := !SRES & D                          "D register

Q_T   := !SRES &  T & !Q_T #                 "T register
         !SRES & !T &  Q_T

Q_JK  := !SRES &  J & !K #                  "JK register
         !SRES & !J & !K  &  Q_JK #
         !SRES &  J &  K  & !Q_JK

Q_SR  := !SRES &  S & !R #                  "SR register (clocked)
         !SRES & !S & !R  &  Q_SR

Q_SL  = ! ( R # !S & !Q_SL )                "SR latch"

"       ↑  = This '!' is actually ignored by the APEEL. The inversion
"            is performed by the output's NEG polarity.


Q_GL  = LEN & LAT #                          "Gated latch"
       !LEN & Q_GL #
        LAT & Q_GL                        "fix hazard when Q_GL=1"


TEST_VECTORS "D Register"
(  CLK  D  SRES  SSET  ARES   ->   Q_D  )
    0   0    0     0     1    ->     X
    0   0    0     0     1    ->     L
    C   0    0     1     0    ->     H
    C   0    1     0     0    ->     L
    C   0    1     1     0    ->     H
    C   0    0     0     0    ->     L
    C   1    0     0     0    ->     H
    C   0    0     0     0    ->     L
    C   1    0     0     0    ->     H

TEST_VECTORS "T Register"
(  CLK  T  SRES  SSET  ARES   ->   Q_T  )
    0   0    0     0     1    ->     L
    C   0    0     1     0    ->     H
    C   0    1     0     0    ->     L
    C   0    1     1     0    ->     H
    C   0    1     1     0    ->     H
    C   1    0     0     1    ->     L
    C   0    0     0     0    ->     L
    C   1    0     0     0    ->     H
```

Listing 5-1 APEEL Source File For V8REGS.APL

```
TEST_VECTORS "JK Register"
(  CLK   J  K   SRES  SSET  ARES   ->   Q_JK  )
    0    0  0    0     0     1     ->     L
    C    0  0    0     1     0     ->     H
    C    0  0    1     0     0     ->     L
    C    0  0    1     1     0     ->     H
    C    0  0    0     0     0     ->     H
    C    0  1    0     0     0     ->     L
    C    1  0    0     0     0     ->     H
    C    1  1    0     0     0     ->     L
    C    0  0    0     0     0     ->     L
    C    1  0    0     0     0     ->     H
    C    0  1    0     0     0     ->     L
    C    1  1    0     0     0     ->     H

TEST_VECTORS "SR Regsister  (clocked)"
(  CLK  SRES  S  R   ->   Q_SR  )
    C     0   0  0   ->     X
    C     0   1  0   ->     H
    C     0   0  0   ->     H
    C     0   0  1   ->     L
    C     0   0  0   ->     L
    C     0   1  1   ->     L

TEST_VECTORS "SR Latch"
(  S  R   ->   Q_SL  )
   0  0   ->     X
   1  0   ->     H
   0  0   ->     H
   0  1   ->     L
   0  0   ->     L
   1  1   ->     L

TEST_VECTORS "Gated Latch"
(  LAT  LEN   ->    Q_GL  )
     0   0    ->     X
     0   1    ->     L
     0   0    ->     L
     1   0    ->     L
     0   1    ->     L
     1   1    ->     H
     1   0    ->     H
     0   0    ->     H
     0   1    ->     L
     0   0    ->     L
```

Listing 5-1 APEEL Source File For V8REGS.APL

APEEL Syntax

This section describes the minimum elements and syntax required to create a file which can be assembled by APEEL software. In the sections that follow, *italics will be used to identify fields in which the user would enter identifiers, such as pin names, etc.*

Basic APEEL File Structure

The basic structure of an APEEL input file is shown below:

Title
Declarations:
    Device Type
    Zero_Power (if applicable)
    Auto_Secure (if applicable)
    Input Pin Assignments
    Output Pin Assignment and Macro Definitions
    Node Assignments (if applicable)
Boolean Equations
Test Vectors (optional)

The examples given in this section are taken from the example application file V8REGS.APL (refer to chapter 6) which is found on PEEL Device Software Applications Disk.

Identifiers

Identifiers are names that identify devices, device pins, nodes, and input or output signals.

The rules governing all identifiers are:

1. Identifiers may use only alphanumeric and underscore characters.

2. Spaces cannot be used in an identifier. To create identifiers which consist of two words, connect them with an underscore character.

Example: Q_GL

3. Identifiers are not case sensitive. Uppercase and lowercase letters are treated the same. Thus, the output pin name **Q_GL** and **Q_gl** will be considered to be the same identifier

Reserved Identifiers

Reserved identifiers are part of the APEEL language and cannot be used to name pins, nodes, or signals. Reserved identifiers are shown below:

TITLE	**PIN**	**NODE**	**ZERO_POWER**
EQUATIONS	**ENABLE**	**TEST_VECTORS**	**AUTO_SECURE**
PEEL18CV8	**P18CV8**	**PEEL20CG10**	**P20CG10**
PEEL22CV10	**P22V10**	**PEEL22CV10Z**	**P22CV10**
PEEL153	**F153**	**F82S153**	**PEEL253**
F253	**PEEL173**	**F173**	**F82S173**
PEEL273	**F273**		

Basic Syntax

In an APEEL source-code file, identifiers must be separated from each other by a space, a comma, or by an operator.

Example:
```
CLK      pin 1
Q_SL     pin 15 = neg com feed_pin
Q_D = !SRES & D
0 0 0 0 1 -> X
```

Comments

The liberal use of comments will make your source file easy to understand. Comments explain what is not readily apparent from the source code itself. Comments do not affect the meaning of the code.

Comments begin with a double quotation mark character (") and end with either another double quotation mark or the end of line character.

Example:
```
"INPUTS"
"Latch outputs internal feedback"
```

Title

The file begins with an optional declaration of the name of the design. The maximum length of the title is 240 characters (3 lines with 80 characters wide).

Format: TITLE ' *Title_of_design* '

Example: TITLE 'PEEL 18CV8 Basic Registers and Latches
DESIGNER: Robin Jigour
DATE: 8/16/87'

Device Type

The target device of the design is declared by simply entering the ICT PEEL device name:

PEEL18CV8	**P18CV8**	**PEEL20CG10**	**P20CG10**
PEEL22CV10	**P22V10**	**PEEL22CV10Z**	**P22CV10Z**
PEEL153	**F153**	**F82S153**	**PEEL253**
F253	**PEEL173**	**F173**	**F82S173**
PEEL273	**F273**		

Input Pin Assignments

Here, names are assigned to the input pins. You can include the "!" or "/" at the beginning of the pin names to implement "Active Low" inputs.

Format: *Pin_Name* PIN *Pin_Number*

Example: D pin 2

I/O Pin Assignments and Macrocell Configurations

Format: *Pin_Name* PIN *Pin_#* = *Macrocell_Configuration*
where Macrocell_Configuration is specified by the following string

Format: *Polarity* *Output Type* *Feedback Path*

You can include a "!" or "/" at the beginning of the pin name of the macrocell assignment (you do not need to specify the macrocell configuration for input definitions) to implement an "Active Low" input or feedback path. If it is an output, the output's polarity will remain in the polarity as specified by the macrocell

polarity attribute.

Example:
```
!A = neg reg feed_reg   (Macrocell Pin Definition)
 A =  I  (Eguation 1  - Output A is "Active Low")
 B =  A  (Equation 2  - Feedback A is "Active Low")
```

If you omit "!" or "/" in your pin name and specify "NEG" for your polarity attribute, you will get an "Active Low" output but an "Active High" feedback path.

Please refer to "Boolean Equations and Logical Operators" (described later in this section) for more information on the effects of feedback signals.

The options available for specifying Polarity, Output Type, and Feedback Path shown in the table below:

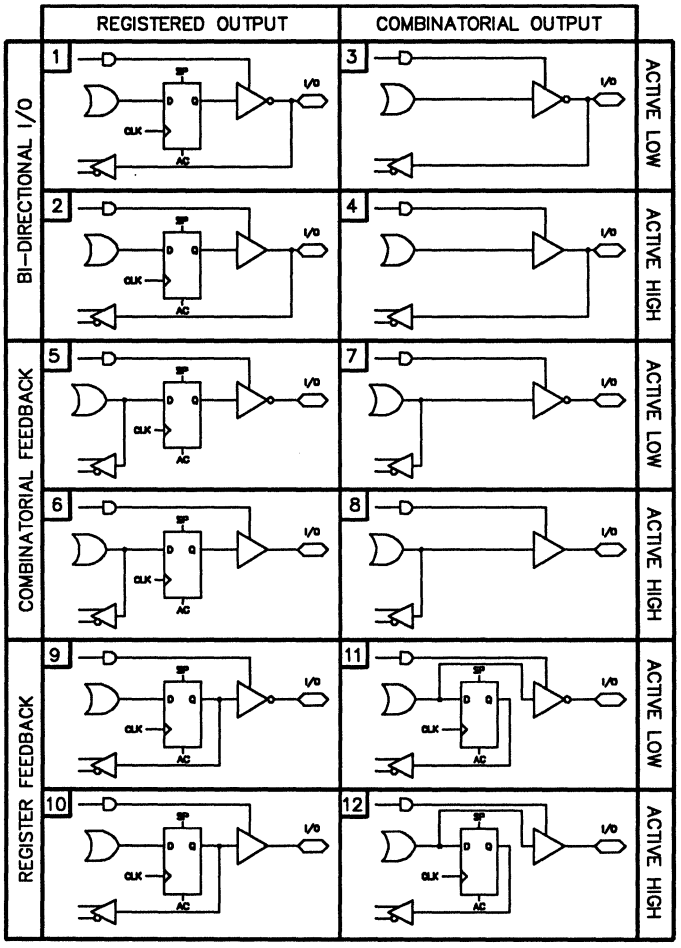
Attribute	Identifier	Definition	PEEL Devices
Polarity:	POS NEG	(positive), (negative)	All PEEL devices
Output Type:	COM REG	(combinatorial), (registered)	PEEL18CV8, PEEL20CG10 PEEL22CV10 PEEL22CV10Z
Feedback Path:	FEED_PIN FEED_OR FEED_REG	(bi-directional I/O), (feedback from the OR gate) (feedback from the flip-flop)	PEEL18CV8 PEEL20CG10 PEEL22CV10Z

Note that some macrocell attributes are not applicable to some of the PEEL devices. An assembling syntax error will occur if the PEEL devices use an invalid attribute. The PEEL devices which the attribute is applicable to are listed under the "PEEL Devices" column in the previous table.

If you do not specify the macrocell configuration, APEEL will default the configuration to:

- "POS COM FEED_PIN" for PEEL18CV8, PEEL20CG10, and PEEL22CV10Z.
- "POS COM" for PEEL22CV10
- "POS" for PEEL153, PEEL253, PEEL173, and PEEL273.

5



Configuration #	PEEL18CV8, PEEL20CG10 PEEL22CV10Z	PEEL22CV10	PEEL153, PEEL253 PEEL173, PEEL273
1	.NEG REG FEED_PIN	N/A	N/A
2	.POS REG FEED_PIN	N/A	N/A
3	.NEG COM FEED_PIN	NEG COM	NEG
4	.POS COM FEED_PIN	POS COM	POS
5	.NEG REG FEED_OR	N/A	N/A
6	.POS REG FEED_OR	N/A	N/A
7	.NEG COM FEED_OR	N/A	N/A
8	.POS COM FEED_OR	N/A	N/A
9	.NEG REG FEED_REG	NEG REG	N/A
10	.POS REG FEED_REG	POS REG	N/A
11	.NEG COM FEED_REG	N/A	N/A
12	.POS COM FEED_REG	N/A	N/A

N/A = Not Applicable

Figure 5-2. APEEL Macrocell Configuration Definitions

Node Assignments

Some PEEL devices have product terms assigned to control the **Asynchronous Clear** and **Synchronous Preset** of the registers. To allow control of these special functions, nodes are assigned to these product terms.

Format: *Node_Name* NODE *Node_Number*

The table below shows the node number assignments for the PEEL devices.

PEEL Devices	Node Number	
	Asynchronous Clear	Synchronous Preset
PEEL18CV8	21	22
PEEL20CG10	25	26
PEEL22CV10	25	26
PEEL22CV10Z	25	26

Boolean Equations and Logic Operators

Equation Format

The logic function to be implemented at each output pin is expressed as a Boolean equation in the **sum-of-products** form. The equations **must** be preceded by the statement **EQUATIONS**

Format: *Pin_Name = Sum-Of-Products_Expression*

Example: Q_SL = !(R # !S & !Q_SL)

The above equation has a "!" just prior to the parenthesis. This "!" character is actually **ignored** by APEEL (otherwise, the expression of the right side of the "=" symbol is not in a sum-of-products form). The inversion of the expression is actually done by setting the output "Q_SL" to NEG polarity (in the macrocell configuration). The reason for inserting the "!" is to make the equation appears logical without refering back to the macrocell configuration assignment statements.

If the "!" or "/" is inserted to the beginning of the output pin name (on the left side of the "=" symbol of the equation), it will also be **ignored** by APEEL. **This means that the polarity of the output is solely controlled by setting the macrocell**

polarity attribute (NEG or POS). However, you can include the "!" or "/" on your output pin names (only in the boolean equations) to match the pin names in your design.

☞ Note, the "=" operator may be used for specifying both combinatorial and registered functions, as the type of output is explicitly defined in the macrocell definition. However, the ":=" operator commonly used by other logic assemblers and compilers to specify registered functions may also be used in APEEL.

Logic Operators

The APEEL assembler allows the use of logic operators that are compatible with either ABEL or PALASM. The designer may choose whichever format is more familiar.

(ABEL)	(PALASM)	Logic Description
&	*	AND
#	+	OR
!	/	NOT

Output Feedback

Below is an example of an APEEL boolean equation with feedback.

Example: Q_SL = !(R # !S & !Q_SL)

Output Q_SL		FeedBack Q_SL	Feedback Logical Level	
Previous State	Next State	Level	Q_SL	!Q_SL
X	L	X	X	X
L	H	L	0	1
H	H	H	1	0

The macrocell definition for the output Q_SL was:

```
Q_SL  pin 15  =  neg  com  feed_pin
```

Because the "!" or "/" has been omitted in the pin name of the macrocell definition statement, the feedback signal Q_SL has the same voltage level but not the same logical level as the previous output level, regardless of the output's polarity or feedback type. In this case, the output is "Active Low" but the feedback is "Active High".

If the "!" or "/" was included in the pin name, then both output and feedback has the same logical level (i.e. "Active Low"). Please refer to the "I/O Pin Assignments and Macrocell Configurations" section described earlier.

In conclusion, the **feedback signal assumes an "Active Low" or "Active High" polarity depending on whether a "!" or "/" was included in the pin name of the macrocell definition statement.** This definition is consistent with the input pin definition.

Output Enable Operator

To control the programmable output enable function, the following format is used:

Format: Enable *I/O_Pin_Name* = *Product_Expression*

Enable *I/O_Pin_Name* = 1 (output enabled unconditionally)

Example: Enable *HZ_BUF* = *OE*

The output will be enabled when Product Expression is TRUE. This example can be found in the file **V8GATES.APL**.

If you do not specify the "Enable" statement, the output is enabled automatically. All unused outputs are disabled automatically.

Test Vectors

Test vectors are defined in the input file for logic simulation and function testing. The vectors are automatically converted to the JEDEC format and incorporated in the JEDEC programming file. Test vectors **must** be preceded by the statement **TEST_VECTORS**. The identifiers and format for defining test vectors in APEEL are described below.

Input transitions:

Symbol	Function
C	Clock (LOW-HIGH-LOW)
1	Input HIGH
0	Input LOW
X	Don't Care

Output transitions:

Symbol	Function
H	Output expect HIGH
L	Output expect LOW
Z	Output expect High-Z
X	Don't examine output

A set of test vectors may include a subset of the total number of input and output pins, and multiple vector sets may be included in one file. When multiple functions are implemented in a single device, it is often convenient to group vectors for each function separately (see the application examples for an illustration). Note that each set of vectors must be preceded by the **TEST_VECTORS** statement.

Format

The labels of the pins being described are separated by spaces or commas and are enclosed by parenthesis as shown:

(*Input_Pins* -> *Output_Pins*)

Below the pin label, list the desired input and expected output transitions. Separate the input transitions with spaces or commas and do likewise with the output transitions, but do not add parenthesis. If feedbacks are used, the associated outputs must be declared as output pins (not input pins).

Example:

```
( S R -> Q_SL )
  0 0 ->  X
  1 0 ->  H
  0 0 ->  H
```

If you have specified the output pin name with a "!" or "/" on the macrocell definition statement to implement an "Active Low" feedback path, you must include the "!" or "/" in the output pin name on test vector pin label statement. Please refer to the "I/O Pin Assignments and Macrocell Configurations" and "Boolean Equations and Logic Operators" (under Output Feedback) sections for additional information.

Example:

```
!A pin 19 = neg reg feed_reg  (Macrocell Pin Definition)
(I X Y Z -> !A B)             (Test_Vector pin name label)
0 1 0 1 -> H L                (Test_Vector data)
```

Creating New APEEL Source Files

APEEL template files for each PEEL device are included on the APEEL Applications Diskette. These files provide the structure for writing a new APEEL design. To use the template files, simply:

- ➤ Select the specific template file for the PEEL device you wish to design with.
- ➤ Build the source file around the template, following the syntax rules presented in this section, by adding the specific pin names, macro definitions, equations, test vectors, and comments needed for your design.
- ➤ Save the newly edited file with a new file name (using the **Save As** command) in order to preserve the template file.

APEEL Template Files Included on the Applications Diskette

PEEL Device	APEEL Template File
PEEL18CV8	ANEWV8.APL
PEEL173	ANEW173.APL
PEEL 273	ANEW273.APL
PEEL153	ANEW153.APL
PEEL 253	ANEW253.APL
PEEL 20CG10	ANEWG10.APL
PEEL 22CV10	ANEWV10.APL
PEEL 22CV10Z	ANEWV10Z.APL

This page is blank

6.0 Application Primer and Examples

A number of application examples for PEEL devices have been provided in the form of APEEL input files which can be found on the Applications diskette (disk# 2). These files show some of the possible applications for PEEL devices and provide an example of the format used to create APEEL input files. These examples and the APEEL applications primer in section 6.1 can be used to familiarize yourself with the capabilities of the APEEL software and designing with PEEL device architectures.

Application Directory

APEEL source files for the PEEL applications listed below may be found on the PEEL Device Applications Diskette.

6.1 APEEL Applications Primer

PEEL18CV8 Applications	File Name
6.2 Basic Logic Gates	V8GATES.APL
6.3 Basic Registers and Latches	V8REGS.APL
6.4 Clock Divider and Address Decod	V8CLKADD.APL
6.5 Bus Programmable 8 to 1 Multiple	V8BUSMUX.APL
6.6 Expandable 8-bit (or Dual 4-bit) Comparator	V8COMP.APL
6.7 8-bit Counter with Function Controls	V8FCNTR.APL
6.8 4-bit Change-of-State Port with Interrupt	V8CPORT.APL

PEEL 22CV10 and 22CV10Z Applications	File Name.
6.9 8-bit Loadable Up/Down Counter	V10CNT8.APL
6.10 8-bit Change-of-State Port with Interrupt	V10ZPORT.APL

PEEL173 Applications	File Name
6.11 16 to 4 Priority Encoder	PRI173.APL

PEEL273 Applications	File Name
6.12 8-bit Greater/Less-Than Mag. Comparator	MAG23.APL
6.13 10-Bit Expandable Equality Comparator	EQU273.APL

6.1 APEEL Application Primer

If programmable logic devices are new to you, this section will give you a brief overview of the how design concepts can be described by APEEL design files which are then assembled into JEDEC programming files.

- ❑ Your Design Concepts
- ❑ The Basics of an APEEL file
- ❑ PEEL Device Architectures
- ❑ The JEDEC File

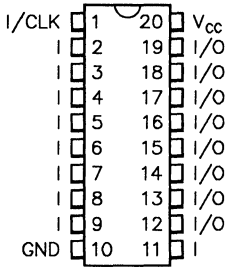
Your Design Concepts

In digital circuits, there are various ways to describe a logic design but probably the most familiar method is by schematic (or gate) form. However, like most logic assemblers APEEL requires logic to be described in equation form. Before converting your design conceptualized in schematic form into APEEL equations, you must first choose the right PEEL device for your specific application. Some of the basic issues when selecting a PEEL device for your design are:

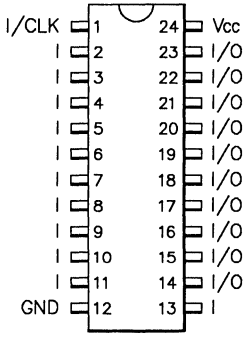
- The speed (t_{PD}, combinatorial propagation delay from input to output), power and cost considerations
- The number of input and I/O pins required
- The complexity of your design in terms of the number of product and sum-of-product functions to be implemented
- The use of output registers (flip-flops)

Most of the above issues can be better understood after reading through the following section on the PEEL device architectures, reviewing the PEEL device data sheets, and experiencing a few APEEL designs. Figure 6.1a gives you a brief summary of the features of all the devices in the PEEL family, but please note that you should refer to the product data sheets for the latest specifications.

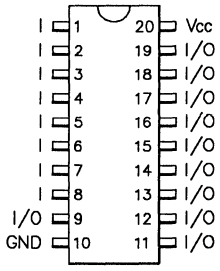
PEEL18CV8



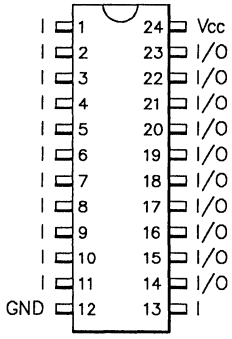
PEEL20CG10
PEEL22CV10
PEEL22CV10Z



PEEL153
PEEL253



PEEL173
PEEL273



	Architecture							Speed*		Power*
Superset Replacement	**Pins**	**Inputs**	**I/Os**	**Registers**	**Macro Configs**	**Prog. Arrays**	**Product / Sum Terms**	**Prop Delay (ns) t_{CO}**	**t_{PD}**	**Supply Current I_{CC} (mA)**
PEEL18CV8	20	10	8	8	12	AND	74	15-25	25-35	20 + 0.7/MHz
PEEL18CV8-15	20	10	8	8	12	AND	74	12	15	80 + 0.5/MHz
PEEL20CG10	24	12	10	10	12	AND	92	15-20	20-35	55 + 0.5/MHz
PEEL22CV10Z	24	12	10	10	12	AND	132	15-20	20-35	55 + 0.5/MHz†
PEEL253	20	8	10	0	2	AND/OR	42/20	N/A	30-40	35 + 1.0/MHz
PEEL273	24	12	10	0	2	AND/OR	42/20	N/A	30-40	35 + 1.0/MHz
Direct Replacement	**Pins**	**Inputs**	**I/Os**	**Registers**	**Macro Configs**	**Prog. Arrays**	**Product / Sum Terms**	**Prop Delay (nS) t_{CO}**	**t_{PD}**	**Supply Current I_{CC} (mA)**
PEEL22CV10	24	12	10	10	4	AND	132	12-20	20-35	55 + 0.5/MHz
PEEL153	20	8	10	0	2	AND/OR	42/10	N/A	30-40	35 + 1.0/MHz
PEEL173	24	12	10	0	2	AND/OR	42/10	N/A	30-40	35 + 1.0/MHz

† User-programmable "zero-power" standby mode: I_{CC} = approx.200µA

Figure 6.1a Pin Configurations and Features Summary of PEEL Devices

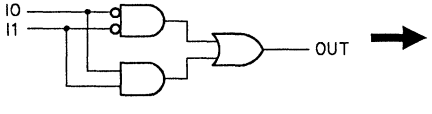
*Please refer to latest PEEL Device data sheets for more detailed and updated AC and DC specifications. Higher speed versions of all devices are scheduled for late 1989-1990.

The Basics of an APEEL file

Once a PEEL device is selected, the process of transferring your conceptualized design into APEEL design file is segmented into three main steps.

- Input and I/O pin assignments
- Macro cell configuration declarations
- Logic equations

Figure 6.1b illustrates a simple exclusive NOR design in schematic form converted to an APEEL design file for the PEEL18CV8 device. First, the input signal names "I1" and "I2" are assigned to input pins 1 and 2 respectively. Next, the output signal "OUT" is assigned to an I/O pin which macro cell is configured as: positive polarity output (POS); combinatorial output (COM); and feedback from the pin (FEED_PIN). In this case, the feedback is actually not being used but it should be specified anyway. Finally, the equivalent equation for the logic function is expressed. Note that the "!" equals the NOT function, "&" equals the AND function, and "#" equals the OR function.



Exclusive NOR

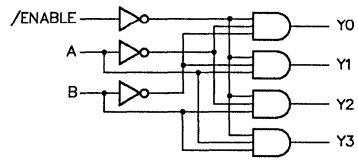
APEEL Design File

```
"Input Pins
I1  pin  1
I2  pin  2

"I/O Pins with Macrocell Definition
Out pin 19 = POS COM FEED_PIN

EQUATIONS
Out  = !I1 & !I2 # I1 & I2
```

Figure 6.1b PEEL18CV8 exclusive NOR Design

Three other similar examples are shown in figures 6.1c (2-to-4 Bit Decoder), 6.1d (4-Bit Shifter), and 6.1e (2-Bit Counter). For a more detailed description of the APEEL language, please refer to section 5.0 and the APEEL application examples found later in this chapter

2-to-4 Decoder

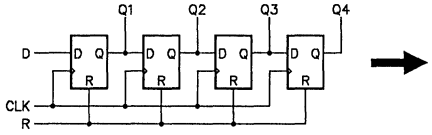


APEEL Design File

```
"Input Pins
A          pin  1
B          pin  2
!ENABLE    pin  3

"I/O Pins with Macrocell Definition
Y0 pin 19 = POS COM FEED_PIN
Y1 pin 18 = POS COM FEED_PIN
Y2 pin 17 = POS COM FEED_PIN
Y3 pin 16 = POS COM FEED_PIN

EQUATIONS
Y0 = ENABLE & !B & !A
Y1 = ENABLE & !B &  A
Y2 = ENABLE &  B & !A
Y3 = ENABLE &  B &  A
```

Figure 6.1f PEEL18CV8 2-to-4 Bit Decoder Design

4-Bit Shifter



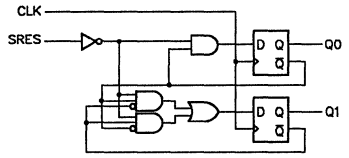
APEEL Design File

```
"Input Pins
CLK  pin  1
A    pin  2
R    pin  3

"I/O Pins with Macrocell Definition
Q1 pin 19 = POS REG FEED_REG
Q2 pin 18 = POS REG FEED_REG
Q3 pin 17 = POS REG FEED_REG
Q4 pin 16 = POS REG FEED_REG

"Node Assignment
AC  node  21 "Asynchronous CLear

EQUATIONS
AC = R

Q1 =  D
Q2 = Q1
Q3 = Q2
Q4 = Q3
```

Figure 6.1g PEEL18CV8 4-Bit Shifter Design

In Figure 6.1g, an additional declaration is needed for the internal node to control the asynchronous clear capability of the PEEL18CV8. Please refer to section 5.0 and the PEEL application examples for a more detail discussion on this feature.

2-Bit Counter



APEEL Design File

```
"Input Pins
CLK   pin  1
SRES  pin  2

"I/O Pins with Macrocell Definition
Q0 pin 19 = POS REG FEED_REG
Q1 pin 18 = POS REG FEED_REG

EQUATIONS
Q0 = !SRES & !Q0
Q1 = !SRES & !Q1 &  Q0 #
     !SRES &  Q1 & !Q0
```

Figure 6.1h PEEL18CV8 2-Bit Counter Design

PEEL Device Architectures

The following section describes PEEL Device Architectures. Although helpful, a detailed understanding of PEEL Device architecture is not always necessary to sucessfully design with PEEL Devices. Many PEEL logic designers, armed with a minimum knowledge of macrocell configurations, pin functions and number of product terms are able to create effective PEEL designs for their specific applications.

The Logic Array

Probably the most basic thing about programmable logic devices is the logic array diagram of the device. The two types of logic structures in the PEEL Device family are the programmable-AND/fixed-OR and programmable-AND/programmable-OR structures. The PEEL18CV8 logic array which utilizes the programmable-AND/fixed-OR structure is shown on Figure 6.1i. Other PEEL devices with this logic structure include PEEL22CV10, PEEL20CG10, PEEL22CV10Z (see Appendix D). As shown in the diagram, the 36 (0-35) input lines which run vertically are derived from true and complement of the 18 possible input pins. The product terms, drawn as horizontal lines, serve as AND functions (denoted by the AND gate symbol at the end of each product term) with a maximum of 36 inputs for each product term. The 74 product terms are made up of: one synchronous preset term; one asynchronous clear term; eight output enable terms; and 64 terms divided into groups of eight, with each group feeding into an OR function. The area where the input lines and the product terms intersect is known as the AND array. At each intersection of the input line and product

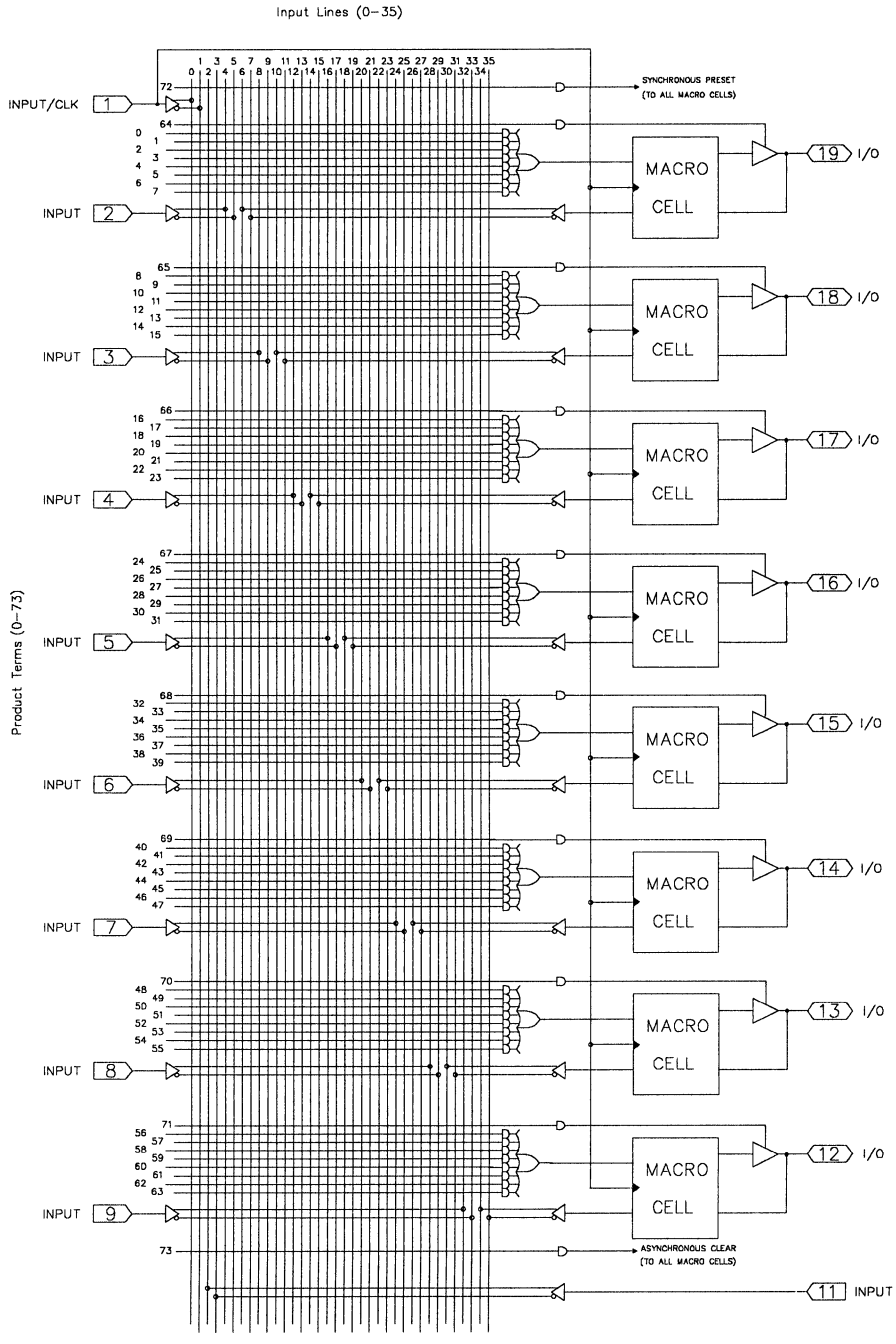


Figure 6.1i PEEL18CV8 Logic Array Diagram
(Programmable-AND/Fixed-OR)

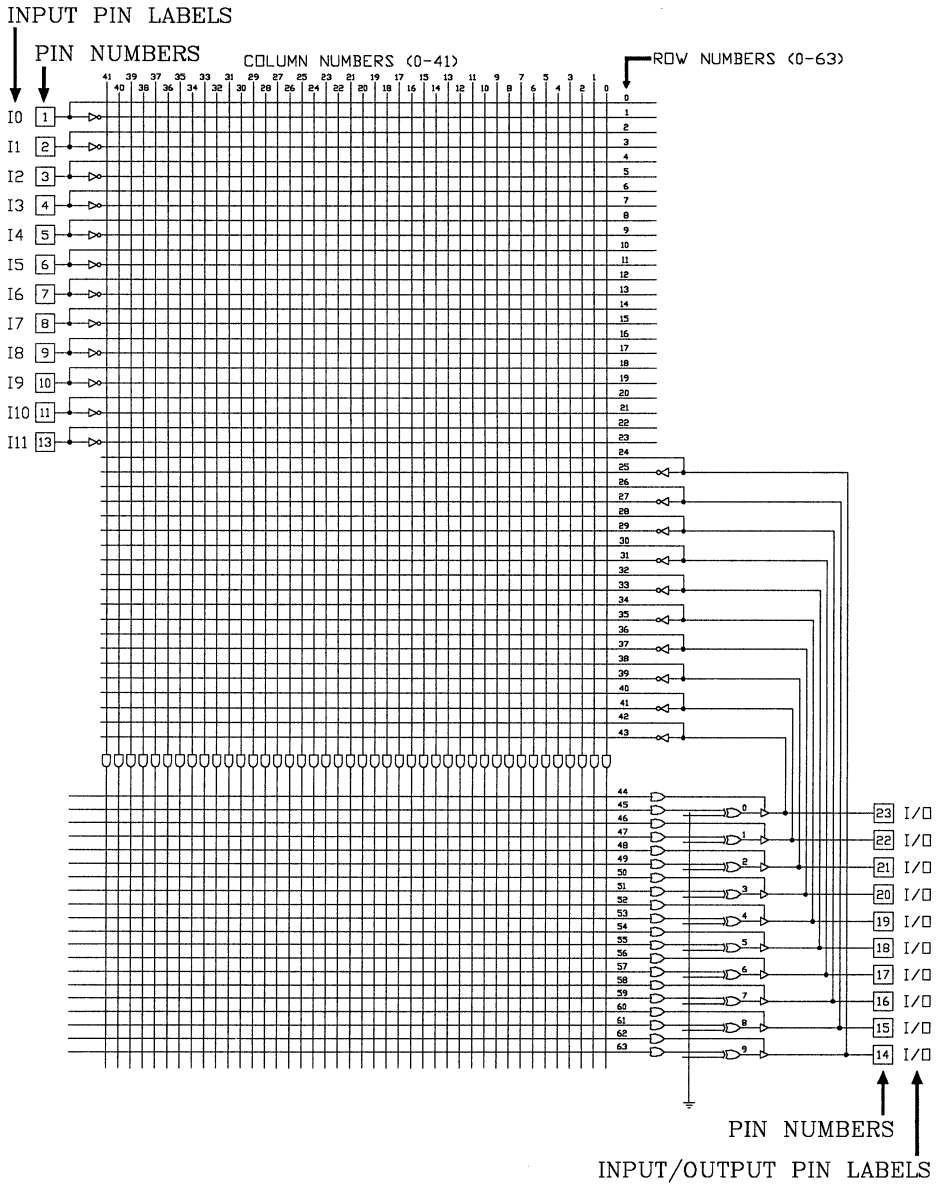


Figure 6.1j PEEL273 Logic Array Diagram
(Programmable-AND/Programmable-OR)

term is an EEPROM cell which determines whether the intersection is connected or open. A connection allows an input line to become a logical input of the intersected product term (AND gate). Hence, logic functions are performed by merely connecting the input lines to the product terms. By connecting specific inputs or I/O macrocell feedbacks to the product terms, complex sum-of-product logic functions can be created. In the PEEL18CV8 device, each sum feeds into its associated I/O macrocell where the logic function can be further controlled for output to an I/O pin and/or feedback into the array. To graphically illustrate the logic functions in the logic array diagram, X's are used to mark the intersection in which the input lines are connected to the product terms (refer to the V8GATES design in section 6.2).

The other logic array structure in the PEEL devices (PEEL153, PEEL253, PEEL173, and PEEL273) is the programmable-AND/programmable-OR structure (see Figure 6.1j and Appendix D). This logic structure is ideal for combinatorial applications such comparitors, priority encoders, etc. which utilize many product terms. Unlike the PEEL18CV8's fixed OR array, the programmable OR array allows user-specified product terms to be connected to the sum terms (drawn as horizontal lines) which are propagated to the I/O pin, output enable buffer, or back to the AND array as feedbacks. This sharing of product terms minimizes product term redundancy and avoids wasting unused terms of fixed-OR arrays.

A Closer Look at the Logic Array

Figure 6-1k illustrates how a logic function (specifically the two-input exclusive NOR design described earlier) is implemented in the PEEL18CV8 array. Note that if all true and complement inputs of a product term are left open, the output of the AND gate will be a logical true (HIGH). If both true and complement of one or more inputs are connected to a product term, the output of the AND gate is forced to a logical false (LOW). Frequently, you will find that the unused product terms are connected to both true and complement of all the possible inputs in the device, creating product terms with JEDEC 0's. Most logic assembler or compiler (including APEEL) omit these unused product terms when outputing the JEDEC file. Basically, any product terms not specified in the JEDEC file are assumed to be unused.

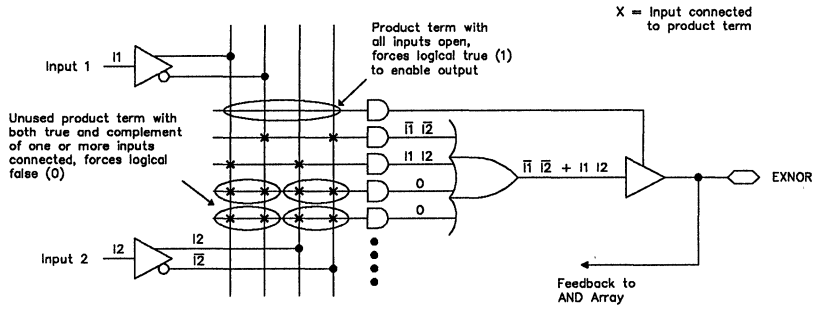


Figure 6.1k Logic function implementation in PEEL18CV8 array

The Macro Cell

In the PEEL18CV8 device, there is a macro cell associated with each I/O pin. This macro cell allows you to configure the architecture of each output independently into one of twelve possible configurations. This flexiblity makes it possible to customize the architecture to your design requirements. PEEL devices with similar macro cell capabilities include PEEL20CG10, PEEL22CV10, and PEEL22CV10Z. Macro cells in PEEL devices such as PEEL153, PEEL253, PEEL173 and PEEL273 provide for output polarity control only. Please refer to 5-12 of this handbook for more information describing the different PEEL macro cell configurations.

As shown in figure 6.1i, the PEEL18CV8 I/O macro cell consists of a D-type flip-flop and two signal-select multiplexers. The configuration of each macrocell is determined by the four EEPROM bits (A,B,C,D) controlling these multiplexers. These bits determine: output polarity; output type (registered or non-registered); and input/feedback path (bi-directional I/O, combinatorial feedback, or register feedback). Figure 6.1j shows the equivalent circuits of the PEEL18CV8 twelve macro cell configurations.

Output Type

The signal from the OR array can be fed directly to the output pin (combinatorial function) or latched in the D-type flip-flop (registered function). The D-type flip-flop latches data on the rising edge of the clock and is controlled by the global preset

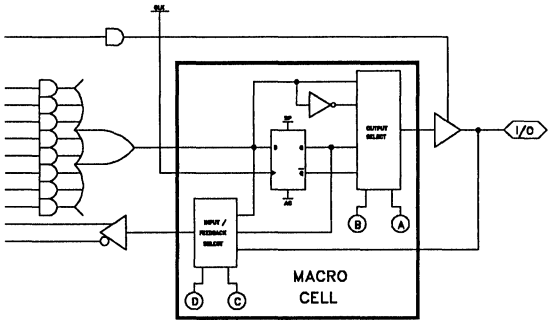


Figure 6.1i PEEL18CV8 Macro Cell

and clear terms. When the synchronous preset term is satisfied, the Q output of the register will be set HIGH at the next rising edge of the clock input. Satisfying the asynchronous clear term will set Q LOW, regardless of the clock state. If both terms are satisfied simultaneously, the clear will override the preset.

Output Polarity

Each macrocell can be configured to implement active-high or active-low logic. Programmable polarity eliminates the need for external inverters.

Output Enable

The output of each I/O macrocell can be enabled or disabled under the control of its associated programmable output enable product term. When the logical conditions programmed on the output enable term are satisfied, the output signal is propagated to the I/O pin. Otherwise, the output buffer is driven into the high-impedance state. Under the control of the output enable term, the I/O pin can function as a dedicated input, output, or a bi-directional I/O.

Input/Feedback Select

The I/O macrocell also provides control over the feedback path. The input/feedback signal associated with each I/O macrocell may be obtained from three different locations: from the I/O pin (bi-directional I/O); directly from the Q output of the flip-flop (registered feedback); or directly from the OR gate (combinatorial feedback).

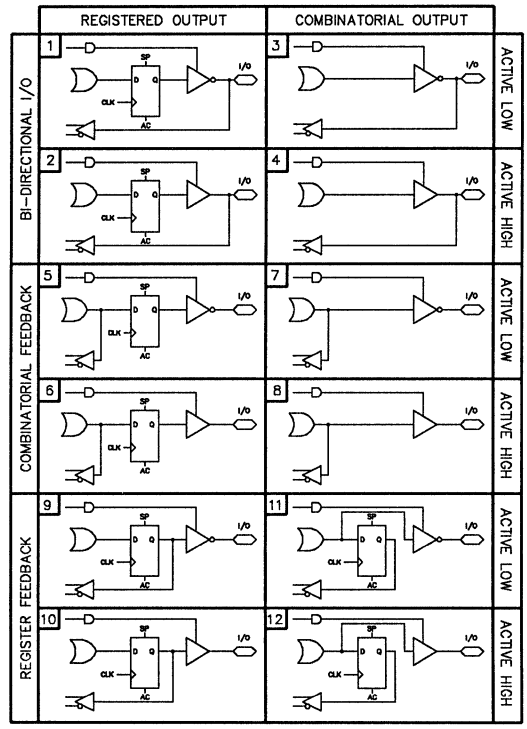


Figure 6.1j PEEL18CV8 macro cell configurations

Bi-directional I/O

The input/feedback signal is taken from the I/O pin when using the pin as a dedicated input or as a bi-directional I/O. (Note that it is possible to create a registered output function with bi-directional I/O.)

Combinatorial Feedback

The signal-select multiplexer gives the macrocell the ability to feedback the output directly from the OR gate, regardless of whether the output function is registered or combinatorial. This feature allows the creation of asynchronous latches, even when the output must be disabled.

Registered Feedback

Feedback also can be taken from the register, regardless of whether the output function is registered or combinatorial. When implementing a combinatorial output and registered feedback configuration (configurations number 11 and 12 in figure 6.1j), the register can be used for internal latching of data while leaving the external output free for combinatorial functions.

The JEDEC File

As mentioned earlier, the APEEL assembler outputs a JEDEC file which can be uploaded into a programmer to program the PEEL device. Referencing to the logic array diagram with the V8GATES application example on section 6.2, the X's which mark the connected inputs to the product term are represented by 0's in the JEDEC file.

After the PEEL device is erased (electrically), all EEPROM cells are set to a JEDEC "1" which open all logical connections. The device is then configured to perform the user-defined function by programming selected connections (JEDEC "0") in the AND array. Please refer to the JEDEC pattern of the V8GATES.APL design. Note that only used product terms are specified.

The JEDEC file is the output of the APEEL assembler (and all other PLD logic compilers). This JEDEC file which is a standard data transfer format for programmable logic devices between the development systems and programmer defines that the data "1" and "0" represent an open and short connection. For more information on the JEDEC standard, you can contact:

Electronic Industries Association
Engineering Department
2001 Eye Street N.W.
Washington, D.C. 20006.

6.2 Basic Gates

DEVICE: PEEL18CV8
FILE NAME: V8GATES.APL

This PEEL18CV8 application example implements several basic logic gates. The logic gates include an inverter, four-input AND, OR, NAND, and NOR gates, a four-input AND-OR- INVERT gate, a two- input XOR gate and a high-impedance buffer. Each gate uses one or more of the (A,B,C and D) inputs. Additionally, the high-impedance buffer uses the /HZ input for impedance control. The truth table for these gates can be examined in the test vectors. Note, the remaining unused input pins can be used as additional inputs into the gates.

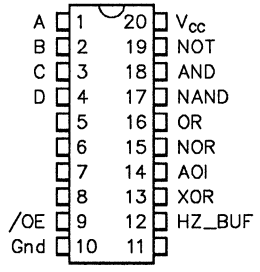


Figure 6.2a – Pinout for V8GATES.APL

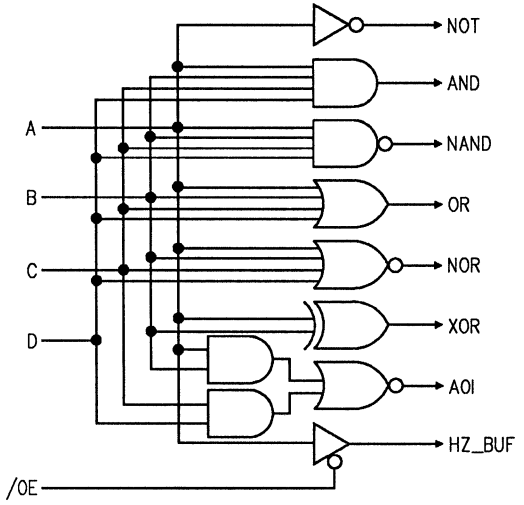
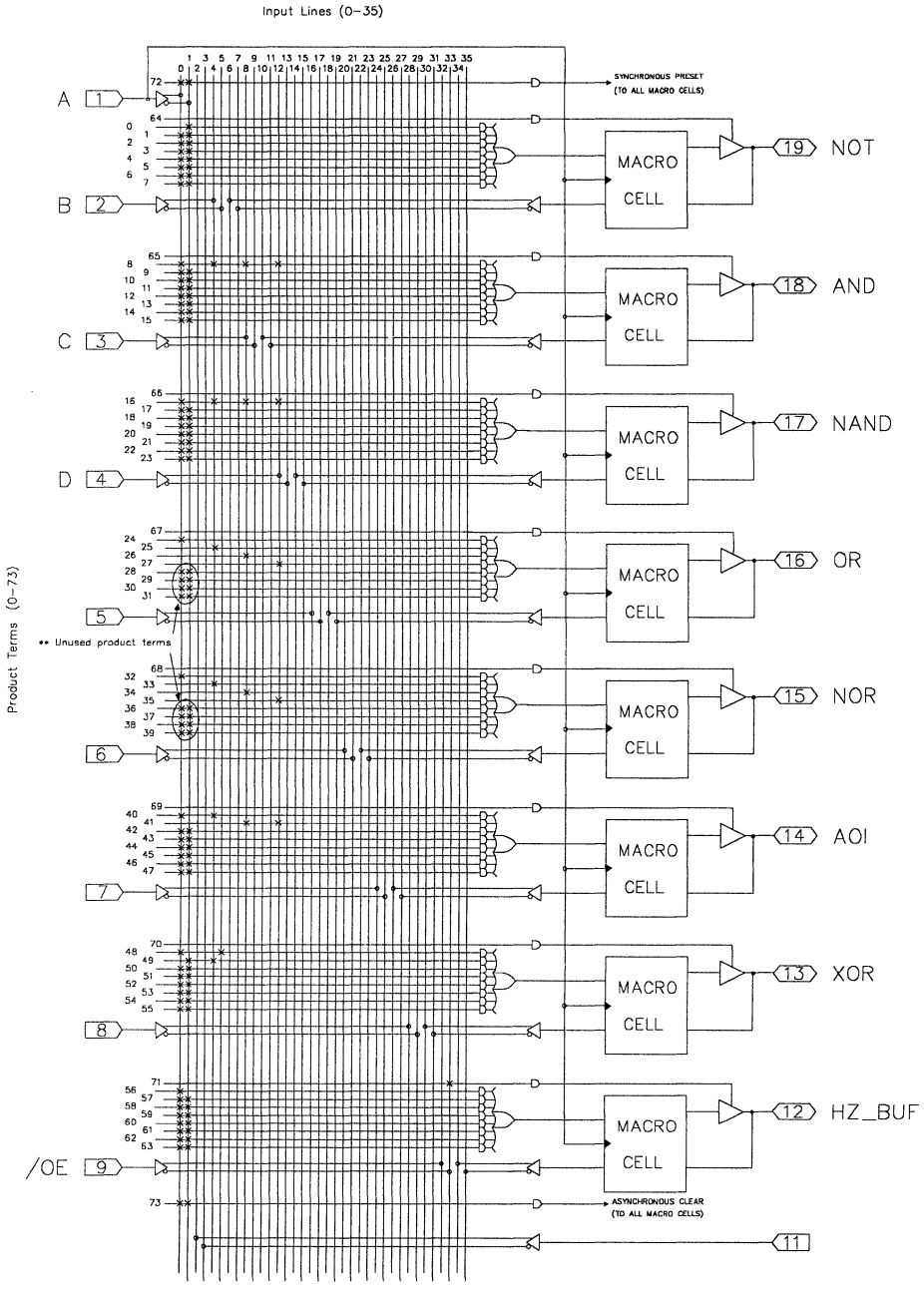


Figure 6.2b – Block diagram for V8GATES.APL

```
TITLE 'APEEL FILE: PEEL18CV8 BASIC GATES
DESIGNER: Robin Jigour
DATE: 8/16/87'

PEEL18CV8



"Inputs"

A            pin 1  "A,B,C, and D are gate inputs.
B            pin 2
C            pin 3
D            pin 4
!OE          pin 9   "The '!' is only used to indicate an
                     "active low output
                     "enable for HZ_BUF. This character
                     "is NOT part of the pin name.


"Outputs and Macro Cell definitions

HZ_BUF       pin 12 = pos com feed_pin  "High Impedence Buffer.
XOR          pin 13 = pos com feed_pin  "Exclusive OR.
AOI          pin 14 = neg com feed_pin  "AND-OR-Invert.
NOR          pin 15 = neg com feed_pin
OR           pin 16 = pos com feed_pin
NAND         pin 17 = neg com feed_pin
AND          pin 18 = pos com feed_pin
NOT          pin 19 = pos com feed_pin  "Inverter.


EQUATIONS

NOT    =  !A                 "Note macro cell polarity for each output.

AND    =   A & B & C & D

NAND   = !(A & B & C & D)   "Here, the '!' (which is actually ignored
                           "by APEEL) is to make the equation
                           "appears logically correct. This output's
                           "macro cell which is set to NEG polarity is
                           "actually doing the inversion.

OR     =   A # B # C # D
```

```
NOR     = !(A # B # C # D)     "Again, the '!' is ignored by APEEL.

AOI     = !(A & B # C & D)     "Same as above.

XOR     =   A & !B # !A & B

HZ_BUF =  A
Enable HZ_BUF = OE        "OE controls output enable.
                           "Outputs without the 'ENABLE' statements are
                           "enabled unconditionally.


TEST_VECTORS
(  D   C   B   A   OE    -> NOT   AND   NAND   OR   NOR   AOI   XOR HZ_BUF )
   0   0   0   0    1    ->  H     L     H      L     H     H     L     L
   0   0   0   1    1    ->  L     L     H      H     L     H     H     H
   0   0   1   0    0    ->  H     L     H      H     L     H     H     Z
   0   0   1   1    0    ->  L     L     H      H     L     L     L     Z
   0   1   0   0    X    ->  X     L     H      H     L     H     X     X
   0   1   0   1    X    ->  X     L     H      H     L     H     X     X
   0   1   1   0    X    ->  X     L     H      H     L     H     X     X
   0   1   1   1    X    ->  X     L     H      H     L     L     X     X
   1   0   0   0    X    ->  X     L     H      H     L     H     X     X
   1   0   0   1    X    ->  X     L     H      H     L     H     X     X
   1   0   1   0    X    ->  X     L     H      H     L     H     X     X
   1   0   1   1    X    ->  X     L     H      H     L     L     X     X
   1   1   0   0    X    ->  X     L     H      H     L     L     X     X
   1   1   0   1    X    ->  X     L     H      H     L     L     X     X
   1   1   1   0    X    ->  X     L     H      H     L     L     X     X
   1   1   1   1    X    ->  X     H     L      H     L     L     X     X
```



** To simplify the diagram, unused product terms are only connected to the true and complement of input pin 1.

Figure 6.2c - PEEL18CV8 Logic Diagram with Fuse Map for V8GATES.APL

V8GATES JEDEC File

```
<STX>ICT APEEL(tm) V3.30.  Tue 3-21-1989  18:17:11
APEEL FILE: PEEL18CV8 BASIC GATES
DESIGNER: Robin Jigour
DATE: 8/16/87
*DF PEEL
*DD 18CV8
*DM ICT
*QP20
*QF2696 *F0 *
N Output Pin 19  *
L0000 1011 1111 1111 1111 1111 1111 1111 1111  *

N Output Pin 18  *
L0288 0111 0111 0111 0111 1111 1111 1111 1111  *

N Output Pin 17  *
L0576 0111 0111 0111 0111 1111 1111 1111 1111  *

N Output Pin 16  *
L0864 0111 1111 1111 1111 1111 1111 1111 1111  *
L0900 1111 0111 1111 1111 1111 1111 1111 1111  *
L0936 1111 1111 0111 1111 1111 1111 1111 1111  *
L0972 1111 1111 1111 0111 1111 1111 1111 1111  *

N Output Pin 15  *
L1152 0111 1111 1111 1111 1111 1111 1111 1111  *
L1188 1111 0111 1111 1111 1111 1111 1111 1111  *
L1224 1111 1111 0111 1111 1111 1111 1111 1111  *
L1260 1111 1111 1111 0111 1111 1111 1111 1111  *

N Output Pin 14  *
L1440 0111 0111 1111 1111 1111 1111 1111 1111  *
L1476 1111 1111 0111 0111 1111 1111 1111 1111  *

N Output Pin 13  *
L1728 0111 1011 1111 1111 1111 1111 1111 1111  *
L1764 1011 0111 1111 1111 1111 1111 1111 1111  *

N Output Pin 12  *
L2016 0111 1111 1111 1111 1111 1111 1111 1111  *
```

```
N Output Enable 19,18,...12  *
L2304 1111 1111 1111 1111 1111 1111 1111 1111 1111  *
L2340 1111 1111 1111 1111 1111 1111 1111 1111 1111  *
L2376 1111 1111 1111 1111 1111 1111 1111 1111 1111  *
L2412 1111 1111 1111 1111 1111 1111 1111 1111 1111  *
L2448 1111 1111 1111 1111 1111 1111 1111 1111 1111  *
L2484 1111 1111 1111 1111 1111 1111 1111 1111 1111  *
L2520 1111 1111 1111 1111 1111 1111 1111 1111 1111  *
L2556 1111 1111 1111 1111 1111 1111 1111 1111 1011  *

N Sync Preset, Async Clear, Macrocell 19,18,...12  *
L2592 0000 0000 0000 0000 0000 0000 0000 0000 0000  *
L2628 0000 0000 0000 0000 0000 0000 0000 0000 0000  *
L2664 0011 0011 1011 0011 1011 1011 0011 0011  *

V0001 0000XXXX0NXLLHHLHLHN *
V0002 1000XXXX0NXHHHLHHLLN *
V0003 0100XXXX1NXZHHLHHLHN *
V0004 1100XXXX1NXZLLLHHLLN *
V0005 0010XXXXXNXXXHLHHLXN *
V0006 1010XXXXXNXXXHLHHLXN *
V0007 0110XXXXXNXXXHLHHLXN *
V0008 1110XXXXXNXXXLLHHLXN *
V0009 0001XXXXXNXXXHLHHLXN *
V0010 1001XXXXXNXXXHLHHLXN *
V0011 0101XXXXXNXXXHLHHLXN *
V0012 1101XXXXXNXXXLLHHLXN *
V0013 0011XXXXXNXXXLLHHLXN *
V0014 1011XXXXXNXXXLLHHLXN *
V0015 0111XXXXXNXXXLLHHLXN *
V0016 1111XXXXXNXXXLLHLHXN *

C6C41 *
<ETX>0000
```

6.3 Basic Registers and Latches

DEVICE: PEEL18CV8
FILE NAME: V8REGS.APL

This application examples demonstrates the implementation of several basic registers and latches within a PEEL18CV8. Four register types are included, D, T, JK, and SR, all of which are clocked by the CLK input. All registers can be synchronously reset, set,and asynchronously reset using the SRES, SSET and ARES inputs respectively. Besides the registers, an SR latch and a Gated Latch circuit show how independent asynchronous storage elements can be implemented. Only the Q outputs of these registers and latches are provided at the output pins. The /Q outputs could easily be accessed by inverting the macro cell output polarity. Truth table operation can be referenced via the test vectors.

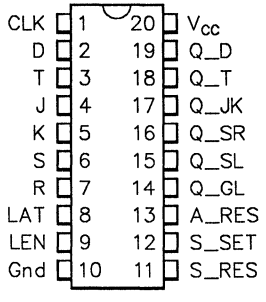


Figure 6. 3a – Pinout for V8REGS.APL

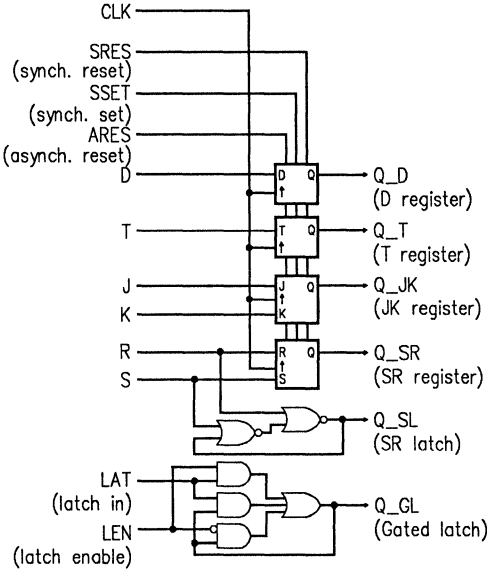


Figure 6.3b – Block diagram for V8REGS.APL

```
TITLE 'APEEL FILE:18CV8 BASIC REGISTERS and LATCHES,
DESIGNER:Robin Jigour, ICT
DATE: 8/16/87'

PEEL18CV8



"Inputs"

CLK         pin 1
D           pin 2                          "register and latch inputs
T           pin 3
J           pin 4
K           pin 5
R           pin 6
S           pin 7
LAT         pin 8                          "gated latch
LEN         pin 9                          "enable for gated latch
SRES        pin 11                          "synchronous reset
SSET        pin 12                          "synchronous set
ARES        pin 13                         "asynchronous reset


"Outputs and Macro Cell definitions

Q_GL       pin 14 = pos com feed_or      "Latch outputs
  "(internal feedback)
Q_SL       pin 15 = neg com feed_pin
Q_SR       pin 16 = pos reg feed_reg     "Register outputs"
Q_JK       pin 17 = pos reg feed_reg
Q_T        pin 18 = pos reg feed_reg
Q_D        pin 19 = pos reg feed_reg


"Internal Nodes"

 AC         node 21                     "Asynchronous Clear node"
 SP         node 22                     "Synchronous Preset node"


EQUATIONS

SP = SSET                            "Synchronous Preset
AC = ARES                            "Asynchronous Clear
```

```
Q_D  := !SRES & D                          "D register

Q_T  := !SRES &  T & !Q_T #                 "T register
        !SRES & !T &  Q_T

Q_JK := !SRES &  J & !K #                  "JK register
        !SRES & !J & !K  &  Q_JK #
        !SRES &  J &  K  & !Q_JK

Q_SR := !SRES &  S & !R #                  "SR register (clocked)
        !SRES & !S & !R  &  Q_SR

Q_SL  = ! ( R # !S & !Q_SL )               "SR latch"

"       ↑  = This '!' is actually ignored by the APEEL. The inversion
"            is performed by the output's NEG polarity.


Q_GL  = LEN & LAT #                        "Gated latch"
       !LEN & Q_GL #
        LAT & Q_GL                        "fix hazard when Q_GL=1"


TEST_VECTORS "D Register"
(  CLK  D  SRES  SSET  ARES   ->   Q_D  )
    0   0    0     0     1    ->     X
    0   0    0     0     1    ->     L
    C   0    0     1     0    ->     H
    C   0    1     0     0    ->     L
    C   0    1     1     0    ->     H
    C   0    0     0     0    ->     L
    C   1    0     0     0    ->     H
    C   0    0     0     0    ->     L
    C   1    0     0     0    ->     H

TEST_VECTORS "T Register"
(  CLK  T  SRES  SSET  ARES   ->   Q_T  )
    0   0    0     0     1    ->     L
    C   0    0     1     0    ->     H
    C   0    1     0     0    ->     L
    C   0    1     1     0    ->     H
    C   0    1     1     0    ->     H
    C   1    0     0     1    ->     L
    C   0    0     0     0    ->     L
    C   1    0     0     0    ->     H
```

```
TEST_VECTORS "JK Register"
(  CLK   J  K   SRES  SSET  ARES   ->   Q_JK  )
    0    0  0    0     0     1     ->    L
    C    0  0    0     1     0     ->    H
    C    0  0    1     0     0     ->    L
    C    0  0    1     1     0     ->    H
    C    0  0    0     0     0     ->    H
    C    0  1    0     0     0     ->    L
    C    1  0    0     0     0     ->    H
    C    1  1    0     0     0     ->    L
    C    0  0    0     0     0     ->    L
    C    1  0    0     0     0     ->    H
    C    0  1    0     0     0     ->    L
    C    1  1    0     0     0     ->    H

TEST_VECTORS "SR Regsister  (clocked)"
(  CLK  SRES  S  R   ->   Q_SR  )
    C     0   0  0   ->     X
    C     0   1  0   ->     H
    C     0   0  0   ->     H
    C     0   0  1   ->     L
    C     0   0  0   ->     L
    C     0   1  1   ->     L

TEST_VECTORS "SR Latch"
(  S  R   ->   Q_SL  )
   0  0   ->     X
   1  0   ->     H
   0  0   ->     H
   0  1   ->     L
   0  0   ->     L
   1  1   ->     L

TEST_VECTORS "Gated Latch"
(  LAT  LEN   ->    Q_GL  )
     0   0    ->     X
     0   1    ->     L
     0   0    ->     L
     1   0    ->     L
     0   1    ->     L
     1   1    ->     H
     1   0    ->     H
     0   0    ->     H
     0   1    ->     L
     0   0    ->     L
```

6.4 Clock Divider and Address Decoder

DEVICE: PEEL18CV8
FILE NAME: V8CLKADD.APL

This application uses the PEEL18CV8 for two common microprocessor system functions: a clock divider and a memory mapped address decoder The clock divider provides divide 2, 4 and 8 clock outputs. The SET input sets all clock outputs high. The address decoder decodes the processor address lines to select one of five memory or I/O devices. The chip select for these devices are active low. The memory map over a 64K boundary is shown below.

Memory Map for Address Decoder

EPROM	(32K X 8)	8000-FFFF hex
EEPROM	(2K X 8)	5000-5FFF hex
UART		4100-41FF hex
PORT		4000-40FF hex
SRAM	(8K X 8)	0000-1FFF hex

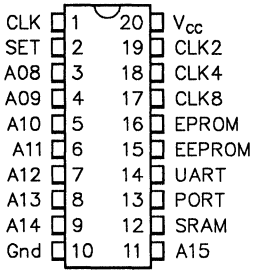


Figure 6.4b – Pinout for V8CLKADD.APL

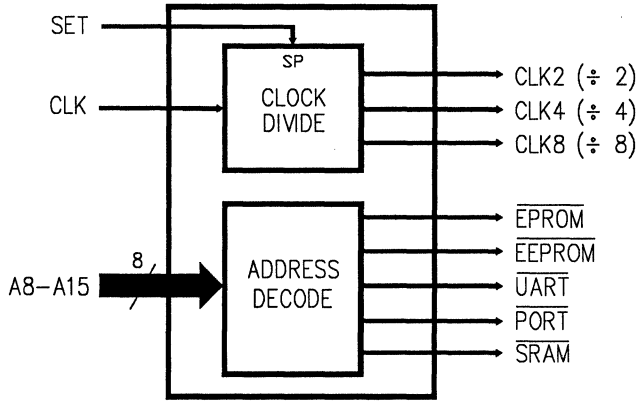


Figure 6.4b – Block diagram for V8CLKADD.APL

```
TITLE 'APEEL FILE: PEEL18CV8 CLOCK DIVIDER AND ADDRESS DECODER
DESIGNER: Robin Jigour, ICT
DATE: 9/20/87'

PEEL18CV8


"PIN ASSIGNMENTS

"Inputs

 CLK       pin 1
 SET       pin 2
 A08       pin 3
 A09       pin 4
 A10       pin 5
 A11       pin 6
 A12       pin 7
 A13       pin 8
 A14       pin 9
 A15       pin 11


"Outputs and Macro Cell definitions

SRAM    pin 12 = neg com feed_pin  "5 Combinatorial outputs.
PORT    pin 13 = neg com feed_pin
UART    pin 14 = neg com feed_pin  "Pins 12-16 have active low outputs.
EEPROM  pin 15 = neg com feed_pin
EPROM   pin 16 = neg com feed_pin
CLK8    pin 17 = pos reg feed_reg  "3 Registered outputs.
CLK4    pin 18 = pos reg feed_reg
CLK2    pin 19 = pos reg feed_reg


"Internal Nodes

 AC       node 21                    "Asynchronous Clear - not used.
 SP       node 22                    "Synchronous Preset.
```

```
EQUATIONS

"Clock Divider

SP = SET                              "If SET=1 set all CLK outputs high.

CLK2 := !CLK2                         "CLK divided by 2.

CLK4 := !CLK4 &  CLK2 #               "CLK divided by 4.
         CLK4 & !CLK2

CLK8 := !CLK8 &  CLK4 &  CLK2 #       "CLK divided by 8.
         CLK8 & !CLK4 #
         CLK8 & !CLK2


"Address Decoder (active low outputs)

/SRAM   =  !A15 & !A14 & !A13 "The '/' is actually ignored by APEEL.
                               "Here, the '/' is inserted to:
                              "  (1) indicate that the output is active
                              "      LOW, and
                              "  (2) allow the user to match the output
                              "      signal name with what's on his/her
                              "      design.
                              "Note: '/' can be used instead of '!'.

/PORT   =  !A15 &  A14 & !A13 & !A12 & !A11 & !A10 & !A09 & !A08

/UART   =  !A15 &  A14 & !A13 & !A12 & !A11 & !A10 & !A09 &  A08

/EEPROM =  !A15 &  A14 & !A13 &  A12 & !A11

/EPROM  =  A15
```

```
TEST_VECTORS "for Clock Divider"
(  CLK   SET    ->CLK8   CLK4   CLK2   )
    C     1     ->  H      H      H
    C     0     ->  L      L      L
    C     0     ->  L      L      H
    C     0     ->  L      H      L
    C     0     ->  L      H      H
    C     0     ->  H      L      L
    C     0     ->  H      L      H
    C     0     ->  H      H      L
    C     0     ->  H      H      H
    C     0     ->  L      L      L

TEST_VECTORS "for Address Decoder"
( A15 A14 A13 A12 A11 A10 A09 A08   -> EPROM EEPROM UART PORT SRAM  )
   0   0   0   X   X   X   X   X    ->  H      H     H    H    L
   0   0   1   X   X   X   X   X    ->  H      H     H    H    H
   0   1   0   0   0   0   0   0    ->  H      H     H    L    H
   0   1   0   0   0   0   0   1    ->  H      H     L    H    H
   0   1   0   1   0   X   X   X    ->  H      L     H    H    H
   0   1   1   0   X   X   X   X    ->  H      H     H    H    H
   1   X   X   X   X   X   X   X    ->  L      H     H    H    H
```

6.5 Bus Programmable 8 to 1 Multiplexer

DEVICE: PEEL18CV8
FILE NAME: V8BUSMUX.APL

This application implements an 8 to 1 multiplexor that can be interfaced to a µP bus. Any one of the 8 inputs (I0-7) can be selectively routed to the output (OUT) by writing (/WR and /CS =0) a 3-bit binary value to the data inputs (DI0-2). The value is stored into a 3-bit latch that controls the multiplexer selection. Because the latch utilizes internal asynchronous feedback (macro configuration #8), the value can also be enabled onto the data outputs (DO0-2). The DI and DO (0-2) pins should be tied together for write/read bus operation. The truth table for the mux is depicted via the test vectors.

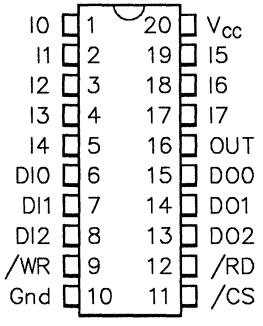


Figure 6.5a – Pinout for V8BUSMUX.APL

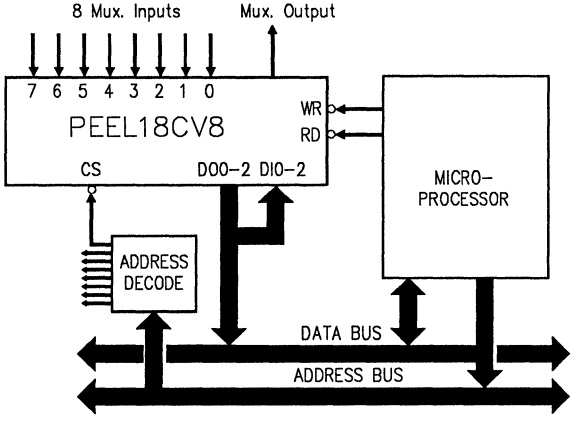


Figure 6.5b - The multiplexer in a typical microprocessor system

```
TITLE 'APEEL FILE: PEEL18CV8 BUS PROGRAMMBLE 8 TO 1 MUX
DESIGNER: Robin Jigour
DATE: 9/13/87'

PEEL18CV8


"PIN DEFINITIONS"


"Inputs"

I0          pin 1
I1          pin 2
I2          pin 3
I3          pin 4
I4          pin 5
DI0         pin 6
DI1         pin 7
DI2         pin 8
!WR         pin 9
!CS         pin 11
!RD         pin 12              "Use as inputs only.
I7          pin 17              "Pins 12, 17-19 macro configurations are
I6          pin 18              "defaulted to 'pos com feed_pin'.
I5          pin 19


"Outputs and Macro Cell definitions

DO2         pin 13 = pos com feed_or    "Internal feedback from the
DO1         pin 14 = pos com feed_or    "input of the D flip-flop.
DO0         pin 15 = pos com feed_or
MOUT        pin 16 = pos com feed_pin


EQUATIONS

DO0 = DI0 & WR & CS #           "Set D0 latch from bus write.
      DO0 & !WR #               "Hold when not selected.
      DO0 & !CS #
      DI0 & DO0                 "Prevent hazard.
Enable DO0 = RD & CS            "Enable D0 output with bus read.
```

```
DO1 = DI1 & WR & CS #          "Set D1 latch from bus write.
      DO1 & !WR #              "Hold when not selected.
      DO1 & !CS #
      DI1 & DO1                "Prevent hazard.
Enable DO1 = RD & CS           "Enable D1 output with bus read.

DO2 = DI2 & WR & CS #          "Set D2 latch from bus write.
      DO2 & !WR #              "Hold when not selected.
      DO2 & !CS #
      DI2 & DO2                "Prevent hazard.
Enable DO2 = RD & CS           "Enable D2 output with bus read.

MOUT = I0 & !DO2 & !DO1 & !DO0 #    "Select I0 when D0-2 = 0
       I1 & !DO2 & !DO1 &  DO0 #    "Select I1 when D0-2 = 1
       I2 & !DO2 &  DO1 & !DO0 #    "Select I2 when D0-2 = 2
       I3 & !DO2 &  DO1 &  DO0 #    "Select I3 when D0-2 = 3
       I4 &  DO2 & !DO1 & !DO0 #    "Select I4 when D0-2 = 4
       I5 &  DO2 & !DO1 &  DO0 #    "Select I5 when D0-2 = 5
       I6 &  DO2 &  DO1 & !DO0 #    "Select I6 when D0-2 = 6
       I7 &  DO2 &  DO1 &  DO0      "Select I7 when D0-2 = 7


TEST_VECTORS "Test programmable 8 to 1 mux"
( I0 I1 I2 I3 I4 I5 I6 I7  DI2 DI1 DI0  CS WR RD  -> DO2 DO1 DO0 MOUT )
   1  0  0  0  0  0  0  0    0   0   0   1  1  0  -> Z   Z   Z   H
   1  0  0  0  0  0  0  0    0   0   0   0  0  0  -> Z   Z   Z   H
   0  0  0  0  0  0  0  0    X   X   X   0  0  0  -> Z   Z   Z   L
   1  0  0  0  0  0  0  0    X   X   X   1  0  1  -> L   L   L   H

   0  0  1  0  0  0  0  0    0   1   0   1  1  0  -> Z   Z   Z   H
   0  0  1  0  0  0  0  0    0   1   0   0  0  0  -> Z   Z   Z   H
   0  0  0  0  0  0  0  0    X   X   X   0  0  0  -> Z   Z   Z   L
   0  0  1  0  0  0  0  0    X   X   X   1  0  1  -> L   H   L   H

   0  0  0  0  0  1  0  0    1   0   1   1  1  0  -> Z   Z   Z   H
   0  0  0  0  0  1  0  0    1   0   1   0  0  0  -> Z   Z   Z   H
   0  0  0  0  0  0  0  0    X   X   X   0  0  0  -> Z   Z   Z   L
   0  0  0  0  0  1  0  0    X   X   X   1  0  1  -> H   L   H   H

   0  0  0  0  0  0  0  1    1   1   1   1  1  0  -> Z   Z   Z   H
   0  0  0  0  0  0  0  1    1   1   1   0  0  0  -> Z   Z   Z   H
   0  0  0  0  0  0  0  0    X   X   X   0  0  0  -> Z   Z   Z   L
   0  0  0  0  0  0  0  1    X   X   X   1  0  1  -> H   H   H   H
```

6.6 8-Bit Expandable Comparator

DEVICE: PEEL18CV8
FILE NAME: V8COMP.APL

This application uses the PEEL18CV8 as an expandable 8-bit (or dual 4-bit) comparator. The inputs are organized into four 4-bit groups and are labeled A0-A3, B0-B3, C0-C3, and D0-D3. The comparison is done between inputs A and C, and inputs B and D. As an 8-bit comparator, both outputs must be tied together with a resistor pull-up (2-20K depending on system speed requirements). If the comparison is equal, the outputs will disable, allowing the resistor to pull the output high. If the comparison of is not equal an output will be asserted low pulling both outputs low. The application utilizes the individual output enables and internal feedback capability of the 18CV8 macro cell to allow expandability. (configuration 7 in figure 6 of the PEEL18CV8 data sheet). This configuration is used to implement a wired-AND function allowing additional comparators outputs to be tied together. Thus, two 18CV8s could create a 16 bit comparator, three a 24 bit comparator, and so on. A dual 4-bit comparator can be achieved by using two pull-up resistors (one resistor for each output). Note, if expandability is not needed, the pull-up resistors can be removed for the dual 4-bit comparator by permanently enabling the outputs.

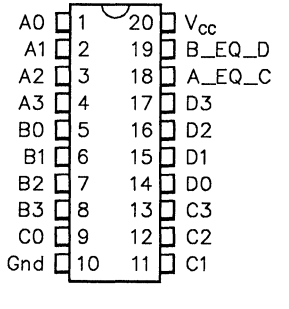


Figure 6.6a – Pinout for V8COMP.APL

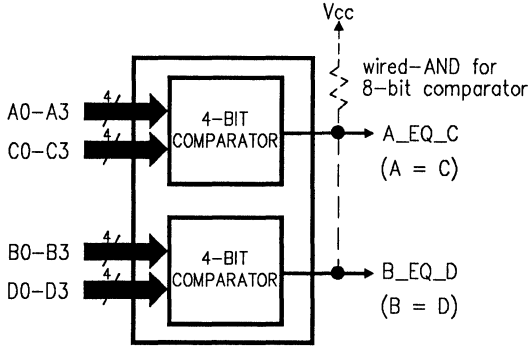


Figure 6.6b – Block diagram for V8COMP.APL

```
TITLE 'APEEL FILE: PEEL18CV8 EXPANADABLE 8-BIT (DUAL 4-BIT) COMPARATOR
DESIGNER: Robin Jigour and James Khong
DATE: 9/13/87'

PEEL18CV8


"PIN DEFINITIONS"

"Inputs"

A0        pin 1
A1        pin 2
A2        pin 3
A3        pin 4
B0        pin 5
B1        pin 6
B2        pin 7
B3        pin 8
C0        pin 9
C1        pin 11
C2        pin 12                "Use as inputs only.
C3        pin 13                "Pins 12-17 macro configurations
D0        pin 14                "are defaulted to 'pos com feed_pin'.
D1        pin 15
D2        pin 16
D3        pin 17


"Outputs Macro Cell definitions

A_EQ_C    pin 18 = neg com feed_or      "Internal feedback.
B_EQ_D    pin 19 = neg com feed_or


EQUATIONS

A_EQ_C = !( !A0 & C0   #  A0 & !C0 #     "Compare A0-3 and C0-3,
            !A1 & C1   #  A1 & !C1 #     "when A does not = C then
            !A2 & C2   #  A2 & !C2 #     "A_EQ_C is 0.
            !A3 & C3   #  A3 & !C3 )
Enable A_EQ_C = !A_EQ_C                  "Enable when A does not = C.
                                         "Disable when A = C.

B_EQ_D = !( !B0 & D0   #  B0 & !D0 #     "Compare B0-3 and D0-3.
```

```
            !B1 & D1   #   B1 & !D1 #      "When B does not = D then
            !B2 & D2   #   B2 & !D2 #      "B_EQ_D is 0
            !B3 & D3   #   B3 & !D3 )
Enable B_EQ_D = !B_EQ_D                    "Enable when B does not = D.
                                           "Disable when B = D.


TEST_VECTORS
(  A3 A2 A1 A0   C3 C2 C1 C0    -> A_EQ_C  )
   0  0  0  0    0  0  0  0     ->     Z
   0  0  0  1    0  0  0  0     ->     L
   0  0  1  0    0  0  1  0     ->     Z
   0  0  1  1    0  0  1  0     ->     L
   0  1  0  0    0  1  0  0     ->     Z
   0  1  0  1    0  1  0  0     ->     L
   0  1  1  0    0  1  1  0     ->     Z
   0  1  1  1    0  1  1  0     ->     L
   1  0  0  0    1  0  0  0     ->     Z
   1  0  0  1    1  0  0  0     ->     L
   1  0  1  0    1  0  1  0     ->     Z
   1  0  1  1    1  0  1  0     ->     L
   1  1  0  0    1  1  0  0     ->     Z
   1  1  0  1    1  1  0  0     ->     L
   1  1  1  0    1  1  1  0     ->     Z
   1  1  1  1    1  1  1  0     ->     L
   1  1  1  1    1  1  1  1     ->     Z

TEST_VECTORS
(  B0 B1 B2 B3   D0 D1 D2 D3    -> B_EQ_D  )
   0  0  0  0    0  0  0  0     ->     Z
   0  0  0  1    0  0  0  0     ->     L
   0  0  1  0    0  0  1  0     ->     Z
   0  0  1  1    0  0  1  0     ->     L
   0  1  0  0    0  1  0  0     ->     Z
   0  1  0  1    0  1  0  0     ->     L
   0  1  1  0    0  1  1  0     ->     Z
   0  1  1  1    0  1  1  0     ->     L
   1  0  0  0    1  0  0  0     ->     Z
   1  0  0  1    1  0  0  0     ->     L
   1  0  1  0    1  0  1  0     ->     Z
   1  0  1  1    1  0  1  0     ->     L
   1  1  0  0    1  1  0  0     ->     Z
   1  1  0  1    1  1  0  0     ->     L
   1  1  1  0    1  1  1  0     ->     Z
   1  1  1  1    1  1  1  0     ->     L
   1  1  1  1    1  1  1  1     ->     Z
```

6.7 8-Bit Counter with Function Controls

DEVICE: PEEL18CV8
FILE NAME: V8FCNTR.APL

This application uses the PEEL18CV8 as an 8 bit counter with four control functions: hold, reset, repeat and output enable. The operation of each control listed below. The Synchronous Preset term was utilized to free-up a product term from the eighth bit of the counter. This allowed the hold function to be implemented. SRES (Synchronous Reset) - When SRES is set high the outputs (Q0-7) will go low after the next clock. When SRES is set high the counter will start counting up with each clock. HOLD (Hold Count) - When HOLD is set high the count will hold the present state. When HOLD is low the counter will resume. REP (Repeat Count) - When REP is set high, the counter repeat the count after reaching FF hex. When REP is set low, the counter will stop after one complete count. OE (Output Enable) - When OE is high the outputs will disable to high impedance. When low, the outputs are enabled. (TEST) - This input is used to preload the registers to simplify test vector operation.

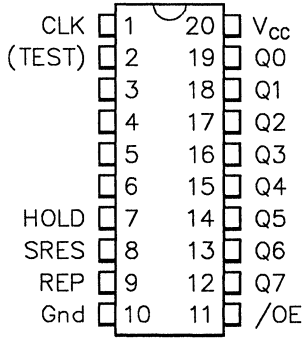


Figure 6.7a – Pinout for V8FCNTR.DOC

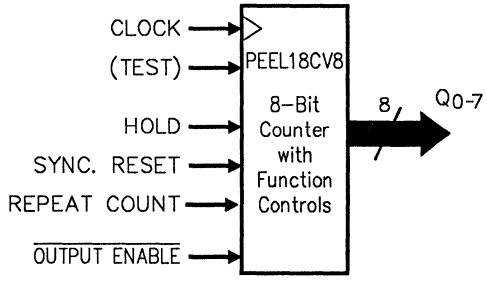


Figure 6.7b – Block diagram for V8FCNTR.APL

```
TITLE 'APEEL FILE: PEEL18CV8 8-Bit Counter with Function Controls
DESIGNER:   Robin Jigour and John Birkner
DATE:       9/14/87'

P18CV8



"PIN ASSIGNMENTS

 CLK      pin 1
 TEST     pin 2                    "For test only, set Q0-Q5 to 1's.
 HOLD     pin 7
 SRES     pin 8
 REP      pin 9
 !OE      pin 11


"Outputs"

 Q7       pin 12 = pos reg feed_reg  "All positive registered outputs.
 Q6       pin 13 = pos reg feed_reg
 Q5       pin 14 = pos reg feed_reg
 Q4       pin 15 = pos reg feed_reg
 Q3       pin 16 = pos reg feed_reg
 Q2       pin 17 = pos reg feed_reg
 Q1       pin 18 = pos reg feed_reg
 Q0       pin 19 = pos reg feed_reg


"Internal Nodes"

 AC        node 21                  "Asynchronous Clear node.
 SP        node 22                  "Synchronous Preset node.


EQUATIONS

AC  = !SRES & !REP & !Q7 & !Q6 & !Q5 & !Q4 & !Q3 & !Q2 & !Q1 & !Q0
        "If REP=1 repeat count else stop at count 00 hex.

SP  = !SRES & Q7 & Q6 & Q5 & Q4 & Q3 & Q2 & Q1 & !Q0
        "Free up product term on Q7.

Q0  =  !SRES & !HOLD & !Q0 #          "Count.
       !SRES &  HOLD &  Q0 #          "Hold state.
       !SRES &  TEST
Enable Q0 = OE                        "Enable output.
```

```
Q1   =   !SRES & !HOLD  & !Q1 &  Q0 #
         !SRES & !HOLD  &  Q1 & !Q0 #
         !SRES &  HOLD  &  Q1 #
         !SRES &  TEST
Enable Q1 = OE

Q2   =   !SRES & !HOLD  & !Q2 &  Q1 &  Q0 #
         !SRES & !HOLD  &  Q2 & !Q1 #
         !SRES & !HOLD  &  Q2 & !Q0 #
         !SRES &  HOLD  &  Q2 #
         !SRES &  TEST
Enable Q2 = OE

Q3   =   !SRES & !HOLD  & !Q3 &  Q2 &  Q1 &  Q0 #
         !SRES & !HOLD  &  Q3 & !Q2 #
         !SRES & !HOLD  &  Q3 & !Q1 #
         !SRES & !HOLD  &  Q3 & !Q0 #
         !SRES &  HOLD  &  Q3 #
         !SRES &  TEST
Enable Q3 = OE

Q4   =   !SRES & !HOLD  & !Q4 &  Q3 &  Q2 &  Q1 &  Q0 #
         !SRES & !HOLD  &  Q4 & !Q3 #
         !SRES & !HOLD  &  Q4 & !Q2 #
         !SRES & !HOLD  &  Q4 & !Q1 #
         !SRES & !HOLD  &  Q4 & !Q0 #
         !SRES &  HOLD  &  Q4 #
         !SRES &  TEST
Enable Q4 = OE

Q5   =   !SRES & !HOLD  & !Q5 &  Q4 &  Q3 &  Q2 &  Q1 &  Q0 #
         !SRES & !HOLD  &  Q5 & !Q4 #
         !SRES & !HOLD  &  Q5 & !Q3 #
         !SRES & !HOLD  &  Q5 & !Q2 #
         !SRES & !HOLD  &  Q5 & !Q1 #
         !SRES & !HOLD  &  Q5 & !Q0 #
         !SRES &  HOLD  &  Q5 #
         !SRES &  TEST
Enable Q5 = OE

Q6   =   !SRES & !HOLD  & !Q6 &  Q5 &  Q4 &  Q3 &  Q2 &  Q1 &  Q0 #
         !SRES & !HOLD  &  Q6 & !Q5 #
         !SRES & !HOLD  &  Q6 & !Q4 #
         !SRES & !HOLD  &  Q6 & !Q3 #
         !SRES & !HOLD  &  Q6 & !Q2 #
         !SRES & !HOLD  &  Q6 & !Q1 #
         !SRES & !HOLD  &  Q6 & !Q0 #
         !SRES &  HOLD  &  Q6
Enable Q6 = OE
```

```
Q7  =  !SRES & !HOLD  & !Q7 &  Q6 &  Q5 &  Q4 &  Q3 &  Q2 &  Q1 &  Q0 #
       !SRES & !HOLD  &  Q7 & !Q6 #
       !SRES & !HOLD  &  Q7 & !Q5 #
       !SRES & !HOLD  &  Q7 & !Q4 #
       !SRES & !HOLD  &  Q7 & !Q3 #
       !SRES & !HOLD  &  Q7 & !Q2 #
       !SRES & !HOLD  &  Q7 & !Q1 #
       !SRES &  HOLD  &  Q7
Enable Q7 = OE


TEST_VECTORS
( CLK  TEST  HOLD  SRES  REP  OE -> Q7  Q6  Q5  Q4  Q3  Q2  Q1  Q0 )
   C    0     0     1     0    1 -> X   X   X   X   X   X   X   X
   C    0     0     1     0    1 -> L   L   L   L   L   L   L   L
   C    0     0     0     1    1 -> L   L   L   L   L   L   L   H
   C    0     0     0     0    0 -> Z   Z   Z   Z   Z   Z   Z   Z
   C    0     0     0     0    1 -> L   L   L   L   L   L   H   H
   C    0     0     1     0    1 -> L   L   L   L   L   L   L   L
   C    0     0     0     0    1 -> L   L   L   L   L   L   L   L
   C    0     0     0     1    1 -> L   L   L   L   L   L   L   H
   C    0     0     0     0    1 -> L   L   L   L   L   L   H   L
   C    1     0     0     0    1 -> L   L   H   H   H   H   H   H
   C    0     0     0     0    1 -> L   H   L   L   L   L   L   L
   C    1     0     0     0    1 -> L   H   H   H   H   H   H   H
   C    0     0     0     0    1 -> H   L   L   L   L   L   L   L
   C    1     0     0     0    1 -> H   L   H   H   H   H   H   H
   C    0     0     0     0    1 -> H   H   L   L   L   L   L   L
   C    0     0     0     0    1 -> H   H   L   L   L   L   L   H
   C    0     0     0     0    1 -> H   H   L   L   L   L   H   L
   C    0     1     0     0    1 -> H   H   L   L   L   L   H   L
   C    0     1     0     0    1 -> H   H   L   L   L   L   H   L
   C    1     0     0     0    1 -> H   H   H   H   H   H   H   H
   C    0     1     0     0    1 -> H   H   H   H   H   H   H   H
   C    0     1     0     0    1 -> H   H   H   H   H   H   H   H
   C    0     0     0     0    1 -> L   L   L   L   L   L   L   L
   C    0     0     0     0    1 -> L   L   L   L   L   L   L   L
   C    0     0     0     1    1 -> L   L   L   L   L   L   L   H
```

6.8 Change-of-State Port with Interrupt

DEVICE PEEL18CV8
FILE NAME: V8CPORT.APL

This application uses the PEEL18CV8 as an 8-bit input port of which 4 of its inputs can detect a change-of-state. When detected, the INTR output is set for interrupting a CPU. The state change is latched by four pseudo-buried registers which can be read by the CPU on D0-D3 as listed in the address table below. Once read, unless another change has occurred, the INTR will be reset. The D4 output can be used for status polling of any remaining state change. The I4-7 inputs do not detect state changes but can be read as a standard input port.

Address			Data Outputs	
A0	CS	RD	D0-3	D4
X	1	X	Hi-Z	Hi-Z
X	X	1	Hi-Z	Hi-Z
0	0	0	Read I0-I3 Change	I0-I3 Pending Change Status
1	0	0	Read I4-I7 Inputs	Don't care

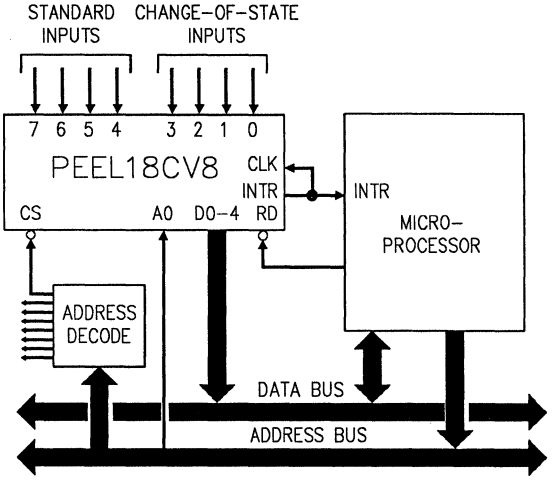


Figure 6.8a – System interface for change-of-state port

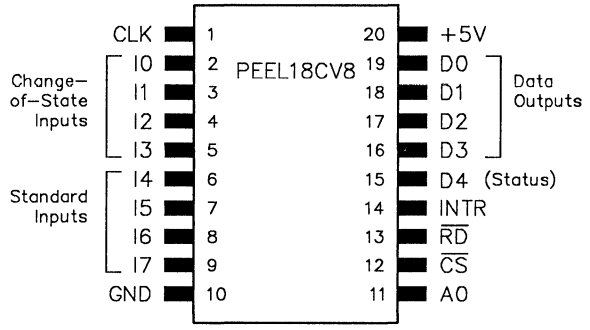


Figure 6.8b – Pinout for V8CPORT.APL

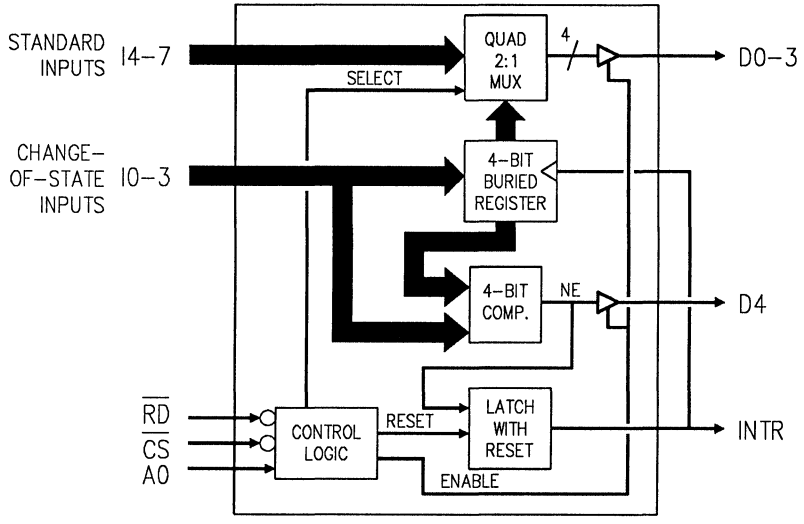


Figure 6.8c – Block diagram showing internal functions of V8CPORT.APL

```
TITLE 'APEEL FILE: PEEL18CV8 CHANGE-OF-STATE INPUT PORT WITH INTERRUPT
DESIGNER: Robin Jigour, ICT
DATE: 9/4/87'

PEEL18CV8


"PIN DEFINITIONS"

"Inputs"

CLK       pin 1    "Must be connected to pin 14, INTR.
I0        pin 2    "I0-I3 inputs can detect change-of-state.
I1        pin 3
I2        pin 4
I3        pin 5
I4        pin 6    "I4-I7 are standard inputs.
I5        pin 7
I6        pin 8
I7        pin 9
A0        pin 11
!CS       pin 12   "Use as inputs only.
!RD       pin 13   "Default macro configuration = pos com feed_pin.


"Outputs"

INTR      pin 14 = pos com feed_or
D4        pin 15 = pos com feed_or    "Internal feedback.
D3        pin 16 = pos com feed_reg
D2        pin 17 = pos com feed_reg
D1        pin 18 = pos com feed_reg
D0        pin 19 = pos com feed_reg   "Pseudo buried registers.


EQUATIONS

D0 = I0 & !CS #                 "I0 to D0 register.
     I0 &  CS & !RD #           "I0 to D0 register.
     D0 &  CS & RD & !A0 #      "Read D0 register.
     I4 &  CS & RD & A0         "Read I4.
     Enable D0 = CS & RD        "Enable onto data bus.
```

```
D1 = I1 & !CS #                "I1 to D1 register.
     I1 &  CS & !RD #          "I1 to D1 register.
     D1 &  CS & RD & !A0 #     "Read D1 register.
     I5 &  CS & RD &  A0       "Read I5.
     Enable D1 = CS & RD       "Enable onto data-bus.

D2 = I2 & !CS #                "I2 to D2 register.
     I2 &  CS & !RD #          "I2 to D2 register.
     D2 &  CS & RD & !A0 #     "Read D2 register.
     I6 &  CS & RD &  A0       "Read I6.
     Enable D2 = CS & RD       "Enable onto data-bus.

D3 = I3 & !CS #                "I3 to D3 register.
     I3 &  CS & !RD #          "I3 to D3 register.
     D3 &  CS &  RD & !A0 #    "Read D3 register.
     I7 &  CS &  RD &  A0      "Read I7.
     Enable D3 = CS & RD       "Enable onto data-bus.

D4 = I0 & !D0 # !I0 & D0 #     "Compare I0-3 with D0-3 registers.
     I1 & !D1 # !I1 & D1 #     "D4=1 if I0-3 and D0-3 are not equal.
     I2 & !D2 # !I2 & D2 #
     I3 & !D3 # !I3 & D3
     Enable D4 = CS & RD       "Enable onto data-bus for status.

INTR = D4   & !CS #  "Latch not-equal status uP interrupt and
       D4   & !RD #  "PEEL clock. Clear interrupt when registers
       D4   &  A0 #  "are read and there are no more input state
       INTR & !CS #  "changes, that is, when D4, CS, RD and A0
       INTR & !RD #  "are all 0.
       INTR &  A0
```

```
TEST_VECTORS "Test change-of-state input port I0-4 operation"
(  CLK   I0 I1 I2 I3   A0   CS   RD    -> D0 D1 D2 D3   D4   INTR  )
     0    0  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     0
     0    0  0  0  0    0    1    1    ->  X  X  X  X    0     0
     0    1  0  0  0    X    0    0    ->  Z  Z  Z  Z    Z     H
     1    1  0  0  0    X    0    0    ->  Z  Z  Z  Z    Z     H
     1    1  0  0  0    0    1    0    ->  Z  Z  Z  Z    Z     H
     1    1  0  0  0    0    1    1    ->  H  L  L  L    L     L
     0    1  0  0  0    0    1    1    ->  H  L  L  L    L     L
     0    1  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     L
     0    0  0  0  0    X    0    0    ->  Z  Z  Z  Z    Z     H
     1    0  0  0  0    X    0    0    ->  Z  Z  Z  Z    Z     H
     1    0  0  0  0    0    1    0    ->  Z  Z  Z  Z    Z     H
     1    0  0  0  0    0    1    1    ->  L  L  L  L    L     L
     0    0  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     L
     0    1  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     H
     1    1  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     H
     1    1  1  1  1    X    0    0    ->  Z  Z  Z  Z    Z     H
     1    1  1  1  1    0    1    0    ->  Z  Z  Z  Z    Z     H
     1    1  1  1  1    0    1    1    ->  H  L  L  L    H     L
     0    1  1  1  1    X    0    0    ->  Z  Z  Z  Z    Z     H
     1    1  1  1  1    0    1    0    ->  Z  Z  Z  Z    Z     H
     1    1  1  1  1    0    1    1    ->  H  H  H  H    L     L
     0    1  1  1  1    X    0    X    ->  Z  Z  Z  Z    Z     L
     0    0  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     H
     1    0  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     H
     1    0  0  0  0    0    1    1    ->  L  L  L  L    L     L
     1    0  0  0  0    X    0    X    ->  Z  Z  Z  Z    Z     L


TEST_VECTORS "Test standard input port I4-7 operation"
(  CLK   I4 I5 I6 I7   A0   CS   RD    -> D0 D1 D2 D3   D4   INTR  )
     0    1  0  1  0    1    0    0    ->  Z  Z  Z  Z    Z     X
     0    1  0  1  0    1    1    0    ->  Z  Z  Z  Z    Z     X
     0    1  0  1  0    1    1    1    ->  H  L  H  L    X     X
     0    0  1  0  1    1    1    0    ->  Z  Z  Z  Z    Z     X
     0    0  1  0  1    1    1    1    ->  L  H  L  H    X     X
     0    0  1  0  1    1    0    0    ->  Z  Z  Z  Z    Z     X
```

6.9 8-Bit Loadable Up/Down Counter with Carry-out or Borrow-in

DEVICE PEEL22CV10Z
FILE V10CNT8.APL

This application uses the PEEL22CV10 as an 8-bit Up/Down Loadable counter. The four controls are:

CLR (Synchronous Clear)—When CLR is set to High, all outputs (Q7-Q0 and CO_BI) will be set to Low on next clock.

UP (Up/Down control)—When UP is set to High, outputs Q7-Q0 will count up on each clock. When UP is set to Low, outputs Q7-Q0 will count down.

LOAD (Load data)—When LOAD is set High, outputs Q7-Q0 will follow the data of D7-D0 on next clock and the output CO_BI will be set to Low.

!OE (Output Enable)—When OE is set to High, all outputs (Q7-Q0 and CO_BI) will be High Impedance. When OE is set Low, all outputs will be enabled.

Note: After counting up 255, the count will go to 0 and the CO_BI will be set High on next clock. The High will remain on the CO_BI pin until LOAD or CLR goes High. The table below describes the operation:

Operation Table

CLK	CLR	UP	LOAD	!OE	D7—D0	Q7—Q0	C0_BI
C	1	X	X	0	X	LOW	0
C	0	1	0	0	X	COUNT UP	0
C	0	1	0	0	X	255→0	1
C	0	0	0	0	X	COUNT DOWN	0
C	0	0	0	0	X	0→255	1
C	0	X	1	0	DATA IN	DATA IN	0
X	X	X	X	1	X	HIGH-Z	0

C = Clock Pulse (0→1)
X = Don't Care

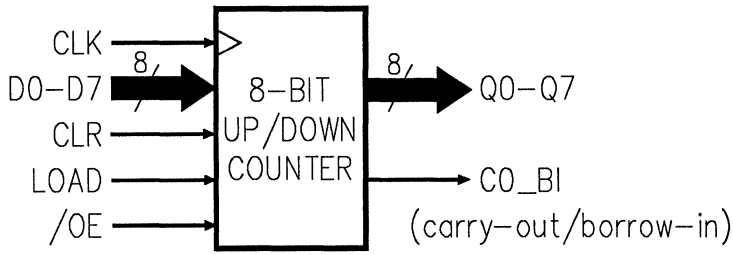


Figure 6.9a – Block diagram for V10CNT8.APL

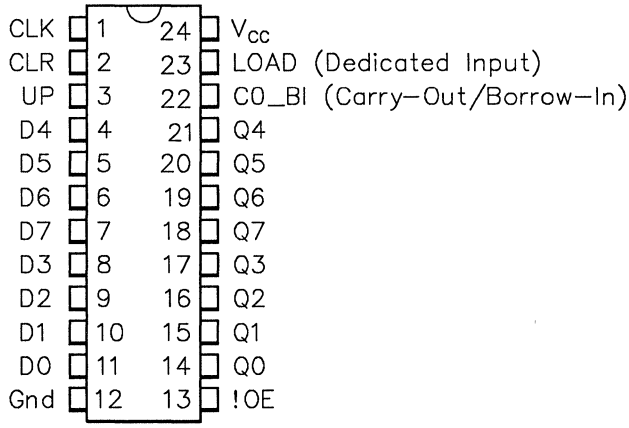


Figure 6.9 b – Pinout for V10CNT8.APL

```
TITLE 'APEEL FILE: PEEL22CV10 8-Bit Up/Down Loadable Counter with
       Carry-Out or Borrow-In.
DESIGNER: James Khong                DATE: 6/9/88'

Peel22CV10


"PIN ASSIGNMENTS

CLK       pin 1
CLR       pin 2
D0        pin 3
D1        pin 4
D2        pin 5
D3        pin 6
D4        pin 7
D5        pin 8
D6        pin 9
D7        pin 10
UP        pin 11
OE        pin 13
LOAD      pin 23      "Used as input only. Default macro configuration
                       "is defaulted to 'pos com feed_pin'.

Q0        pin 14 = pos reg    "All positive registered outputs.
Q1        pin 15 = pos reg
Q2        pin 16 = pos reg
Q3        pin 17 = pos reg
Q7        pin 18 = pos reg
Q6        pin 19 = pos reg
Q5        pin 20 = pos reg
Q4        pin 21 = pos reg
CO_BI     pin 22 = pos reg    "Carry-Out / Borrow-In.

AC         node 25            "Asynchronous Clear node - not used.
SP         node 26            "Synchronous Preset node.


Equations

Enable Q7     = !OE
Enable Q6     = !OE
Enable Q5     = !OE
```

```
Enable Q4    = !OE
Enable Q3    = !OE
Enable Q2    = !OE
Enable Q1    = !OE
Enable Q0    = !OE
Enable CO_BI = !OE

SP = !CLR & !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7 & !UP & !LOAD;

Q7 = !CLR &  Q0 & Q7 & !UP & !LOAD #  "Count Down.
     !CLR &  Q1 & Q7 & !UP & !LOAD #
     !CLR &  Q2 & Q7 & !UP & !LOAD #
     !CLR &  Q3 & Q7 & !UP & !LOAD #
     !CLR &  Q4 & Q7 & !UP & !LOAD #
     !CLR &  Q5 & Q7 & !UP & !LOAD #
     !CLR &  Q6 & Q7 & !UP & !LOAD #
     !CLR & !Q0 & Q7 &  UP & !LOAD #  "Count Up.
     !CLR & !Q1 & Q7 &  UP & !LOAD #
     !CLR & !Q2 & Q7 &  UP & !LOAD #
     !CLR & !Q3 & Q7 &  UP & !LOAD #
     !CLR & !Q4 & Q7 &  UP & !LOAD #
     !CLR & !Q5 & Q7 &  UP & !LOAD #
     !CLR & !Q6 & Q7 &  UP & !LOAD #
     !CLR &  Q0 & Q1 &  Q2 & Q3 & Q4 & Q5 & Q6 & !Q7 & UP & !LOAD #
     !CLR &  D7 & LOAD                "Load Data.

Q6 = !CLR & !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !UP & !LOAD #
     !CLR &  Q0 & Q6 & !UP & !LOAD #
     !CLR &  Q1 & Q6 & !UP & !LOAD #
     !CLR &  Q2 & Q6 & !UP & !LOAD #
     !CLR &  Q3 & Q6 & !UP & !LOAD #
     !CLR &  Q4 & Q6 & !UP & !LOAD #
     !CLR &  Q5 & Q6 & !UP & !LOAD #
     !CLR & !Q0 & Q6 &  UP & !LOAD #
     !CLR & !Q1 & Q6 &  UP & !LOAD #
     !CLR & !Q2 & Q6 &  UP & !LOAD #
     !CLR & !Q3 & Q6 &  UP & !LOAD #
     !CLR & !Q4 & Q6 &  UP & !LOAD #
     !CLR & !Q5 & Q6 &  UP & !LOAD #
     !CLR &  Q0 & Q1 & Q2 & Q3 & Q4 & Q5 & !Q6 & UP & !LOAD #
     !CLR &  D6 & LOAD

Q5 = !CLR & !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !UP & !LOAD #
     !CLR &  Q0 & Q5 & !UP & !LOAD #
     !CLR &  Q1 & Q5 & !UP & !LOAD #
```

```
      !CLR &  Q2 & Q5 & !UP & !LOAD #
      !CLR &  Q3 & Q5 & !UP & !LOAD #
      !CLR &  Q4 & Q5 & !UP & !LOAD #
      !CLR & !Q0 & Q5 &  UP & !LOAD #
      !CLR & !Q1 & Q5 &  UP & !LOAD #
      !CLR & !Q2 & Q5 &  UP & !LOAD #
      !CLR & !Q3 & Q5 &  UP & !LOAD #
      !CLR & !Q4 & Q5 &  UP & !LOAD #
      !CLR &  Q0 & Q1 &  Q2 & Q3 & Q4 & !Q5 & UP & !LOAD #
      !CLR & D5 & LOAD

Q4 = !CLR & !Q0 & !Q1 & !Q2 & !Q3 & !Q4 & !UP & !LOAD #
      !CLR &  Q0 & Q4 & !UP & !LOAD #
      !CLR &  Q1 & Q4 & !UP & !LOAD #
      !CLR &  Q2 & Q4 & !UP & !LOAD #
      !CLR &  Q3 & Q4 & !UP & !LOAD #
      !CLR & !Q0 & Q4 &  UP & !LOAD #
      !CLR & !Q1 & Q4 &  UP & !LOAD #
      !CLR & !Q2 & Q4 &  UP & !LOAD #
      !CLR & !Q3 & Q4 &  UP & !LOAD #
      !CLR &  Q0 & Q1 & Q2 & Q3 & !Q4 & UP & !LOAD #
      !CLR & D4 & LOAD

Q3 = !CLR & !Q0 & !Q1 & !Q2 & !Q3 & !UP & !LOAD #
      !CLR &  Q0 & Q3 & !UP & !LOAD #
      !CLR &  Q1 & Q3 & !UP & !LOAD #
      !CLR &  Q2 & Q3 & !UP & !LOAD #
      !CLR & !Q0 & Q3 &  UP & !LOAD #
      !CLR & !Q1 & Q3 &  UP & !LOAD #
      !CLR & !Q2 & Q3 &  UP & !LOAD #
      !CLR &  Q0 & Q1 &  Q2 & !Q3 & UP & !LOAD #
      !CLR &  D3 & LOAD

Q2 = !CLR & !Q0 & !Q1 & !Q2 & !UP & !LOAD #
      !CLR &  Q0 & Q2 & !UP & !LOAD #
      !CLR &  Q1 & Q2 & !UP & !LOAD #
      !CLR & !Q0 & Q2 &  UP & !LOAD #
      !CLR & !Q1 & Q2 &  UP & !LOAD #
      !CLR &  Q0 & Q1 & !Q2 & UP & !LOAD #
      !CLR &  D2 & LOAD

Q1 = !CLR & !Q0 & !Q1 & !UP & !LOAD #
      !CLR &  Q0 &  Q1 & !UP & !LOAD #
      !CLR & !Q0 &  Q1 &  UP & !LOAD #
      !CLR &  Q0 & !Q1 &  UP & !LOAD #
      !CLR &  D1 & LOAD
```

```
Q0 = !CLR & !Q0 & !LOAD #
     !CLR &  D0 &  LOAD

CO_BI = !CLR & !Q0   & !Q1 & !Q2 & !Q3 & !Q4 & !Q5 & !Q6 & !Q7
             & !UP & !LOAD #
        !CLR &  Q0   &  Q1 &  Q2 &  Q3 &  Q4 &  Q5 &  Q6 &  Q7
             &  UP & !LOAD #
        !CLR & CO_BI & !LOAD  "Latch in the previous CO_BI


Test_vectors
(CLK CLR UP LOAD D7 D6 D5 D4 D3 D2 D1 D0 OE ->Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0 CO_BI)
  C   1  X   X    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  L  L    L
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  L  H    L
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  H  L    L
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  H  H    L
  C   0  1   1    1  1  1  1  1  1  0  1  0 -> H  H  H  H  H  H  L  H    L
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> H  H  H  H  H  H  H  L    L
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> H  H  H  H  H  H  H  H    L
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  L  L    H
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  L  H    H
  C   0  0   1    0  1  0  1  0  1  0  1  0 -> L  H  L  H  L  H  L  H    L
  C   0  1   0    X  X  X  X  X  X  X  X  0 -> L  H  L  H  L  H  H  L    L
  C   0  0   0    X  X  X  X  X  X  X  X  0 -> L  H  L  H  L  H  L  H    L
  C   0  0   0    X  X  X  X  X  X  X  X  0 -> L  H  L  H  L  H  L  L    L
  C   0  0   1    0  0  0  0  0  0  1  0  0 -> L  L  L  L  L  L  H  L    L
  C   0  0   0    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  L  H    L
  C   0  0   0    X  X  X  X  X  X  X  X  0 -> L  L  L  L  L  L  L  L    L
  C   0  0   0    X  X  X  X  X  X  X  X  0 -> H  H  H  H  H  H  H  H    H
  C   0  0   0    X  X  X  X  X  X  X  X  0 -> H  H  H  H  H  H  H  L    H
  C   0  0   0    X  X  X  X  X  X  X  X  0 -> H  H  H  H  H  H  L  H    H
  C   1  1   1    X  X  X  X  X  X  X  X  X -> L  L  L  L  L  L  L  L    L
```

6.10 Change-of-State Port with Interrupt

DEVICE PEEL22CV10Z
FILE NAME: V10ZPORT.APL

This application uses the PEEL22CV10Z as an 8-bit input port of which all of its inputs can detect a change-of-state. When detected, the INTR output is set for interrupting a CPU. The state change is latched by eight pseudo-buried registers which can be read by the CPU on D0-D7 as listed in the address table below. Once read, unless another change has occurred, the INTR will be reset. The NEQ output can be used for status polling of any remaining state change. To reduce the average power consumption, the ZERO-POWER mode is used.

Address			Data Outputs	
A0	CS	RD	D0-7	NEQ
X	1	X	Hi-Z	Hi-Z
X	X	1	Hi-Z	Hi-Z
0	0	0	Read I0-I7 Change	I0-I7 Pending Change Status

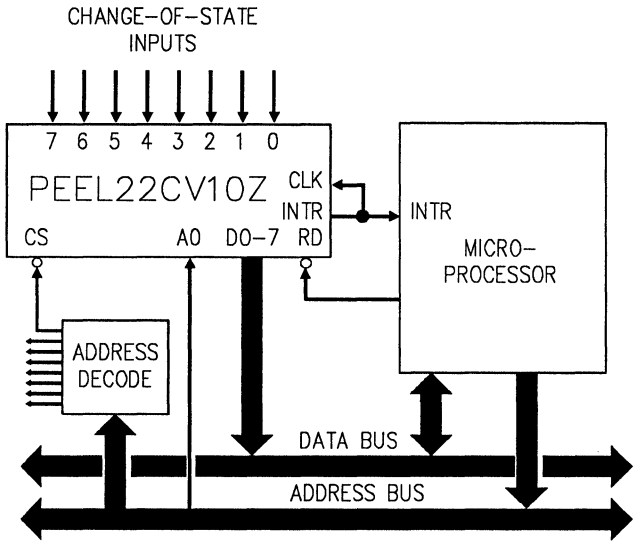


Figure 6.10a – System Interface for V10ZPORT.APL

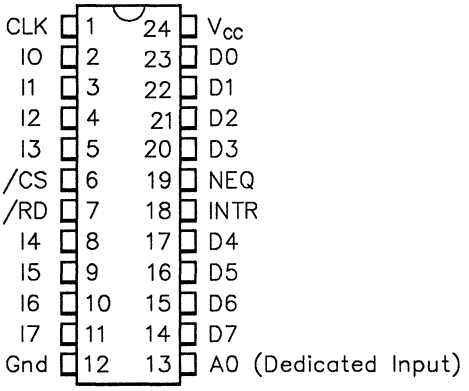


Figure 6.10b – Pinout for V10ZPORT.APL

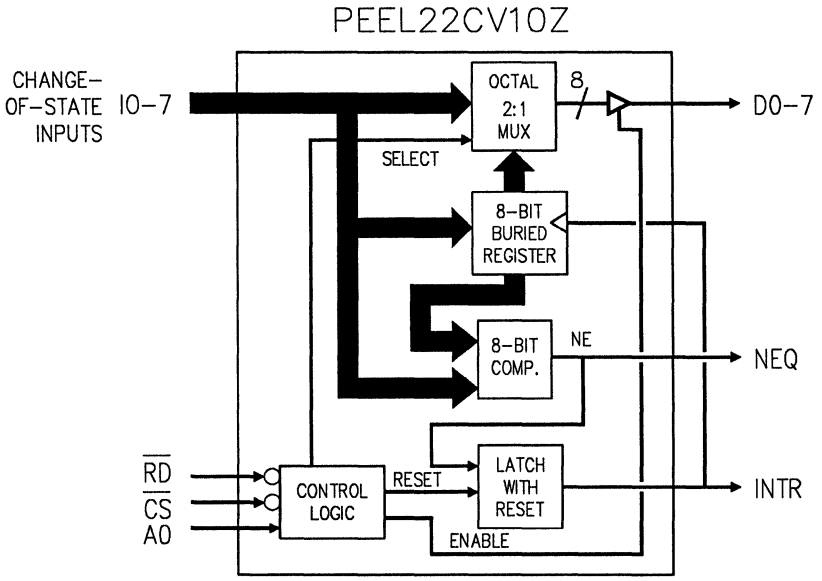


Figure 6.10c – Block diagram showing internal functions of V10ZPORT.APL

```
TITLE 'APEEL FILE: PEEL22CV10Z CHANGE-OF-STATE INPUT PORT WITH INTERRUPT
DESIGNER: Robin Jigour, ICT
DATE: 5/1/87'

PEEL22CV10Z

ZERO_POWER "The key word 'ZERO_POWER' is omitted for non zero-power mode.
            "For zero-power applications, this key word must be specified
            "after the part number declaration but prior to pin list
           "definition.

"PIN DEFINITIONS"

"Inputs"

CLK       pin 1    "Must be connected to pin 18, INTR.
I0        pin 2    "I0-I7 inputs can detect change-of-state.
I1        pin 3
I2        pin 4
I3        pin 5
!RD       pin 6
!CS       pin 7
I4        pin 8
I5        pin 9
I6        pin 10
I7        pin 11
A0        pin 13

"Outputs"

D7        pin 14 = pos com feed_reg    "Pseudo buried registers.
D6        pin 15 = pos com feed_reg
D5        pin 16 = pos com feed_reg
D4        pin 17 = pos com feed_reg
INTR      pin 18 = pos com feed_or
NEQ       pin 19 = pos com feed_or
D3        pin 20 = pos com feed_reg
D2        pin 21 = pos com feed_reg
D1        pin 22 = pos com feed_reg
D0        pin 23 = pos com feed_reg
```

```
EQUATIONS

D0 = I0 & !CS #                  "I0 to D0 register.
     I0 &  CS & !RD #            "I0 to D0 register.
     D0 &  CS & RD & !A0 #       "Read D0 register.
     I0 &  CS & RD & A0          "Read I0.
     Enable D0 = CS & RD         "Enable onto data bus.

D1 = I1 & !CS #                  "I1 to D1 register.
     I1 &  CS & !RD #            "I1 to D1 register.
     D1 &  CS & RD & !A0 #       "Read D1 register.
     I1 &  CS & RD &  A0         "Read I1.
     Enable D1 = CS & RD         "Enable onto data-bus.

D2 = I2 & !CS #                  "I2 to D2 register.
     I2 &  CS & !RD #            "I2 to D2 register.
     D2 &  CS & RD & !A0 #       "Read D2 register.
     I2 &  CS & RD &  A0         "Read I2.
     Enable D2 = CS & RD         "Enable onto data-bus.

D3 = I3 & !CS #                  "I3 to D3 register.
     I3 &  CS & !RD #            "I3 to D3 register.
     D3 &  CS &  RD & !A0 #      "Read D3 register.
     I3 &  CS &  RD &  A0        "Read I3.
     Enable D3 = CS & RD         "Enable onto data-bus.

D4 = I4 & !CS #                  "I4 to D4 register.
     I4 &  CS & !RD #            "I4 to D4 register.
     D4 &  CS & RD & !A0 #       "Read D4 register.
     I4 &  CS & RD & A0          "Read I4.
     Enable D4 = CS & RD         "Enable onto data bus.

D5 = I5 & !CS #                  "I1 to D5 register.
     I5 &  CS & !RD #            "I1 to D5 register.
     D5 &  CS & RD & !A0 #       "Read D5 register.
     I5 &  CS & RD &  A0         "Read I5.
     Enable D5 = CS & RD         "Enable onto data-bus.

D6 = I6 & !CS #                  "I6 to D6 register.
     I6 &  CS & !RD #            "I6 to D6 register.
     D6 &  CS & RD & !A0 #       "Read D6 register.
     I6 &  CS & RD &  A0         "Read I6.
     Enable D6 = CS & RD         "Enable onto data-bus.
```

```
D7 = I7 & !CS #                  "I7 to D7 register.
     I7 &  CS & !RD #            "I7 to D7 register.
     D7 &  CS &  RD & !A0 #      "Read D7 register.
     I7 &  CS &  RD &  A0        "Read I7.
     Enable D7 = CS & RD         "Enable onto data-bus.

NEQ = I0 & !D0 # !I0 & D0 #      "Compare I0-7 with D0-7 registers.
      I1 & !D1 # !I1 & D1 #      "NEQ=1 if I0-7 and D0-7 are not equal.
      I2 & !D2 # !I2 & D2 #
      I3 & !D3 # !I3 & D3 #
      I4 & !D4 # !I4 & D4 #
      I5 & !D5 # !I5 & D5 #
      I6 & !D6 # !I6 & D6 #
      I7 & !D7 # !I7 & D7

INTR = NEQ  & !CS #     "Latch not-equal status for uP interrupt and
       NEQ  & !RD #     "PEEL clock. Clear interrupt when registers
       NEQ  &  A0 #     "are read and there are no more input state
       INTR & !CS #     "changes, that is, when NEQ, CS, RD and A0
       INTR & !RD #     "are all 0.
       INTR &  A0


TEST_VECTORS "Test change-of-state input port I0-7 operation"
(CLK I0 I1 I2 I3 I4 I5 I6 I7 A0 CS RD  -> D0 D1 D2 D3 D4 D5 D6 D7 NEQ INTR)
  0  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   X  0
  0  0  0  0  0  0  0  0  0  0   1  1 -> X  X  X  X  X  X  X  X   L  0
  0  1  0  0  0  0  0  0  0  X   0  0 -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
  1  1  0  0  0  0  0  0  0  X   0  0 -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
  1  1  0  0  0  0  0  0  0  0   1  0 -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
  1  1  0  0  0  0  0  0  0  0   1  1 -> H  L  L  L  L  L  L  L   L  L
  0  1  0  0  0  0  0  0  0  0   1  1 -> H  L  L  L  L  L  L  L   L  L
  0  1  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  L
  0  0  0  0  0  0  0  0  0  X   0  0 -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
  1  0  0  0  0  0  0  0  0  X   0  0 -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
  1  0  0  0  0  0  0  0  0  0   1  0 -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
  1  0  0  0  0  0  0  0  0  0   1  1 -> L  L  L  L  L  L  L  L   L  L
  0  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  L
  0  1  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
  1  1  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
  1  1  1  1  1  1  1  1  1  X   0  0 -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
  1  1  1  1  1  1  1  1  1  0   1  0 -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
  1  1  1  1  1  1  1  1  1  0   1  1 -> H  L  L  L  L  L  L  L   H  L
  0  1  1  1  1  1  1  1  1  X   0  0 -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
  1  1  1  1  1  1  1  1  1  0   1  0 -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
```

```
1  1  1  1  1  1  1  1  1  0   1  1 -> H  H  H  H  H  H  H  H   L  L
0  1  1  1  1  1  1  1  1  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  L
0  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
1  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
1  0  0  0  0  0  0  0  0  0   1  1 -> L  L  L  L  L  L  L  L   L  L
0  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  L
1  1  1  1  1  1  1  1  1  0   1  1 -> H  L  L  L  L  L  L  L   H  L
0  1  1  1  1  1  1  1  1  X   0  0 -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
1  1  1  1  1  1  1  1  1  0   1  0 -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
1  1  1  1  1  1  1  1  1  0   1  1 -> H  H  H  H  H  H  H  H   L  L
0  1  1  1  1  1  1  1  1  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  L
0  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   H  H
1  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  H
1  0  0  0  0  0  0  0  0  0   1  1 -> L  L  L  L  L  L  L  L   L  L
0  0  0  0  0  0  0  0  0  X   0  X -> Z  Z  Z  Z  Z  Z  Z  Z   L  L
```

6.11 16 to 4 Priority Encoder

DEVICE: PEEL173
FILE NAME: PRI173.APL

This PEEL173 application implements a 16 to 4 priority encoder with high-impedance outputs. If any D0-DF input goes low, the GS (group strobe) output will go high and the binary value of the highest priority input will be placed on the E0-E3 (Encoded) outputs when enabled by OE (output enable). The D0 input is highest priority and DF lowest. When the E0-E3 outputs are disabled (a function of OE or !GS) they assume a high-impedance state. This makes it possible to interface the encoded outputs onto a system bus where GS might serve as an interrupt line to a uP and OE as the chip select. The high-impedance control also allows multiple PEEL173 priority encoders to be bussed together for creating wider (32, 48 or 64 bit etc.) priority encoders. To add additional encoders, the highest priority OE must be tied low, and the GS must control the next highest priority OE. The multiple GS can be further encoded to identify which device is driving the E0-E3 lines.

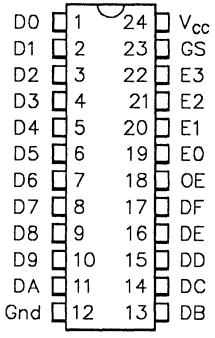


Figure 6.11a - Pinout for PRI173.APL

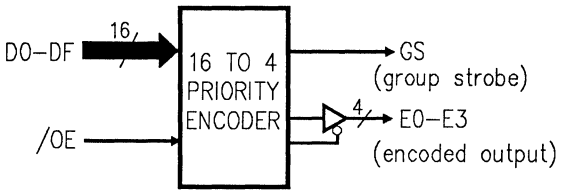


Figure 6.11b - Block diagram for PRI173.APL

```
TITLE 'APEEL FILE: PEEL173 16 TO 4 PRIORITY ENCODER
DESIGNER: Robin Jigour, ICT
DATE: 10/15/87'

PEEL173


"PIN DEFINITIONS"


"Inputs"

D0        pin  1
D1        pin  2
D2        pin  3
D3        pin  4
D4        pin  5
D5        pin  6
D6        pin  7
D7        pin  8
D8        pin  9
D9        pin 10
DA        pin 11
DB        pin 13
DC        pin 14          "Pins 14-18, default polarity = pos.
DD        pin 15
DE        pin 16
DF        pin 17
OE        pin 18


"Outputs

E0        pin 19 = pos
E1        pin 20 = pos
E2        pin 21 = pos
E3        pin 22 = pos
GS        pin 23 = neg
```

```
EQUATIONS

GS = !(D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9
       & DA & DB & DC & DD & DE & DF)

E0 = D0 & !D1 #
     D0 & D1 & D2 & !D3 #
     D0 & D1 & D2 & D3 & D4 & !D5 #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & !D7 #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & !D9 #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9 & DA & !DB #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9 & DA & DB & DC
        & !DD #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9 & DA & DB & DC
        & DD & DE
Enable E0 = !OE & GS

E1 = D0 & D1 & !D2 #
     D0 & D1 & D2 & !D3 #
     D0 & D1 & D2 & D3 & D4 & D5 & !D6 #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & !D7 #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9 & !DA #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9 & DA & !DB #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9 & DA & DB & DC & DD
Enable E1 = !OE & GS

E2 = D0 & D1 & D2 & D3 & !D4 #
     D0 & D1 & D2 & D3 & D4 & !D5 #
     D0 & D1 & D2 & D3 & D4 & D5 & !D6 #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & !D7 #
     D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7 & D8 & D9 & DA & DB
Enable E2 = !OE & GS

E3 = D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7
Enable E3 = !OE & GS
```

```
TEST_VECTORS
(D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF OE ->E3 E2 E1 E0  GS )
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0 ->Z  Z  Z  Z   L
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 ->Z  Z  Z  Z   L
  0  X  X  X  X  X  X  X  X  X  X  X  X  X  X  X  0 ->L  L  L  L   H
  1  0  X  X  X  X  X  X  X  X  X  X  X  X  X  X  0 ->L  L  L  H   H
  1  1  0  X  X  X  X  X  X  X  X  X  X  X  X  X  0 ->L  L  H  L   H
  1  1  1  0  X  X  X  X  X  X  X  X  X  X  X  X  0 ->L  L  H  H   H
  1  1  1  1  0  X  X  X  X  X  X  X  X  X  X  X  0 ->L  H  L  L   H
  1  1  1  1  1  0  X  X  X  X  X  X  X  X  X  X  0 ->L  H  L  H   H
  1  1  1  1  1  1  0  X  X  X  X  X  X  X  X  X  0 ->L  H  H  L   H
  1  1  1  1  1  1  1  0  X  X  X  X  X  X  X  X  0 ->L  H  H  H   H
  1  1  1  1  1  1  1  1  0  X  X  X  X  X  X  X  0 ->H  L  L  L   H
  1  1  1  1  1  1  1  1  1  0  X  X  X  X  X  X  0 ->H  L  L  H   H
  1  1  1  1  1  1  1  1  1  1  0  X  X  X  X  X  0 ->H  L  H  L   H
  1  1  1  1  1  1  1  1  1  1  1  0  X  X  X  X  0 ->H  L  H  H   H
  1  1  1  1  1  1  1  1  1  1  1  1  0  X  X  X  0 ->H  H  L  L   H
  1  1  1  1  1  1  1  1  1  1  1  1  1  0  X  X  0 ->H  H  L  H   H
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  X  0 ->H  H  H  L   H
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0 ->H  H  H  H   H
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0 ->Z  Z  Z  Z   L
```

6.12 8-Bit Greater-Than/Less-Than Magnitude Comparator

DEVICE: PEEL273
FILE NAME: MAG273.APL

This application uses the PEEL273 to implement a magnitude comparator that compares two groups of eight inputs (A0-7 and B0-7) to provide both greater-then (AB) and less-then (A) outputs. The comparator can be cascaded by connecting the A_GT_B and A_LT_B outputs of up to eight PEEL273s to the An and Bn inputs a next level PEEL273. Thus nine PEEL273s would allow for a 64-bit magnitude comparator. Comparators, especially magnitude comparators use many exclusive-OR functions for testing for equality or inequality. Because of this, the sum-of-product equations for just one output of the magnitude comparator requires 2 to the n-1 product terms per bit. With eight bits that comes to 255 product terms per sum, more then are available with even the PEEL273. However, by using multi-level logic many of the high order exclusive-OR functions (EQ4-EQ7) are first implemented and then used in the final magnitude comparator output equations, thus, reducing product terms to 39.

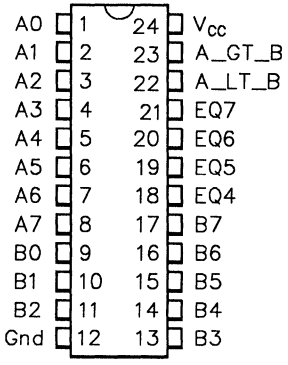


Figure 6.12a – Pinout for MAG273.APL

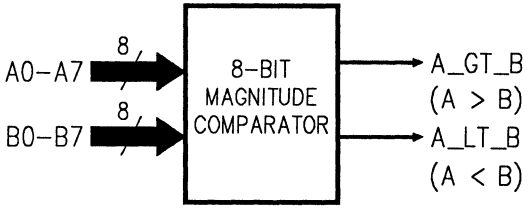


Figure 6.12b – Block diagram for MAG273.APL

```
TITLE 'APEEL FILE: PEEL273 8-BIT GREATER-THAN/LESS-THAN
       MAGNITUDE COMPARATOR
DESIGNER: Robin Jigour, ICT
DATE: 10/11/87'

PEEL273



"PIN DEFINITIONS"

"Inputs"

A0        pin  1
A1        pin  2
A2        pin  3
A3        pin  4
A4        pin  5
A5        pin  6
A6        pin  7
A7        pin  8
B7        pin  9
B6        pin 10
B5        pin 11
B4        pin 13
B3        pin 14          "Pins 14-17, default polarity = pos.
B2        pin 15
B1        pin 16
B0        pin 17


"Outputs

E4        pin 18 = NEG "E4-E7 are used to minimize equation product terms.
E5        pin 19 = NEG
E6        pin 20 = NEG
E7        pin 21 = NEG
A_LT_B    pin 22 = POS
A_GT_B    pin 23 = POS
```

```
EQUATIONS

Enable A_GT_B = 1       "Enable outputs (unconditionally).
Enable A_LT_B = 1
Enable E7     = 1
Enable E6     = 1
Enable E5     = 1
Enable E4     = 1

E7  = !( A7 & !B7 #      "A7=B7  (XNOR)
         !A7 &  B7)

E6  = !( A6 & !B6 #      "A6=B6  (XNOR)
         !A6 &  B6)

E5  = !( A5 & !B5 #      "A5=B5  (XNOR)
         !A5 &  B5)

E4  = !( A4 & !B4 #      "A4=B4  (XNOR)
         !A4 &  B4)


A_GT_B = A7 & !B7 #
     E7 & A6 & !B6 #
     E7 & E6 & A5 & !B5 #
     E7 & E6 & E5 & A4 & !B4 #
     E7 & E6 & E5 & E4 &  A3 & !B3 #
     E7 & E6 & E5 & E4 & !A3 & !B3 &  A2 & !B2 #
     E7 & E6 & E5 & E4 &  A3 &  B3 &  A2 & !B2 #
     E7 & E6 & E5 & E4 & !A3 & !B3 & !A2 & !B2 &  A1 & !B1 #
     E7 & E6 & E5 & E4 & !A3 & !B3 &  A2 &  B2 &  A1 & !B1 #
     E7 & E6 & E5 & E4 &  A3 &  B3 & !A2 & !B2 &  A1 & !B1 #
     E7 & E6 & E5 & E4 &  A3 &  B3 &  A2 &  B2 &  A1 & !B1 #
     E7 & E6 & E5 & E4 & !A3 & !B3 & !A2 & !B2 & !A1 & !B1 & A0 & !B0 #
     E7 & E6 & E5 & E4 & !A3 & !B3 & !A2 & !B2 &  A1 &  B1 & A0 & !B0 #
     E7 & E6 & E5 & E4 & !A3 & !B3 &  A2 &  B2 & !A1 & !B1 & A0 & !B0 #
     E7 & E6 & E5 & E4 & !A3 & !B3 &  A2 &  B2 &  A1 &  B1 & A0 & !B0 #
     E7 & E6 & E5 & E4 &  A3 &  B3 & !A2 & !B2 & !A1 & !B1 & A0 & !B0 #
     E7 & E6 & E5 & E4 &  A3 &  B3 & !A2 & !B2 &  A1 &  B1 & A0 & !B0 #
     E7 & E6 & E5 & E4 &  A3 &  B3 &  A2 &  B2 & !A1 & !B1 & A0 & !B0 #
     E7 & E6 & E5 & E4 &  A3 &  B3 &  A2 &  B2 &  A1 &  B1 & A0 & !B0
```

```
A_LT_B = !A_GT_B & !E7  #
         !A_GT_B & !E6  #
         !A_GT_B & !E5  #
         !A_GT_B & !E4  #
         !A_GT_B &  A3 & !B3  #
         !A_GT_B & !A3 &  B3  #
         !A_GT_B &  A2 & !B2  #
         !A_GT_B & !A2 &  B2  #
         !A_GT_B &  A1 & !B1  #
         !A_GT_B & !A1 &  B1  #
         !A_GT_B &  A0 & !B0  #
         !A_GT_B & !A0 &  B0


TEST_VECTORS
( A7 A6 A5 A4 A3 A2 A1 A0   B7 B6 B5 B4 B3 B2 B1 B0   ->A_GT_B   A_LT_B )
   0  0  0  0  0  0  0  0    0  0  0  0  0  0  0  0   ->   L       L
   0  0  0  0  0  0  0  1    0  0  0  0  0  0  0  0   ->   H       L
   0  0  0  0  0  0  0  0    0  0  0  0  0  0  0  1   ->   L       H
   1  0  1  0  1  0  1  0    1  0  1  0  1  0  1  0   ->   L       L
   1  0  1  0  1  0  1  0    0  1  0  1  0  1  0  1   ->   H       L
   0  1  0  1  0  1  0  1    1  0  1  0  1  0  1  0   ->   L       H
   1  1  1  1  1  1  1  1    1  1  1  1  1  1  1  1   ->   L       L
   1  1  1  1  1  1  1  1    1  1  1  1  1  1  1  0   ->   H       L
   1  1  1  1  1  1  1  0    1  1  1  1  1  1  1  1   ->   L       H
```

6.13 10-Bit Expandable Equality Comparator

DEVICE: PEEL273
FILE NAME: EQU273.APL

This application uses the PEEL273 as an equality comparator for two 10-bit values (A0-9 and B0-9). When EXP (expand) is low, the PEEL273 works in a single mode, directly driving the EQU (equal) output high when A=B and low when A=/B. When EXP is high the EQU output simulates an open-drain output allowing expansion to multiple PEEL273 comparators with EQU outputs tied in parallel with a pull-up resistor (not to exceed 625 ohms). Thus, using two PEEL273s, a 20-bit equivalency comparator could be achieved. When using multiple comparators the EQU line will pull high only when all A and B inputs of each comparator are equal. The truth table is listed below:

Inputs A0-9		B0-9	EXP	Output EQU
A	=	B	0	1
A	≠	B	0	0
A	=	B	1	Z
A	≠	B	1	0

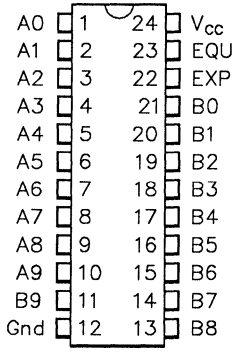


Figure 6.13a – Pinout for EQU273.APL

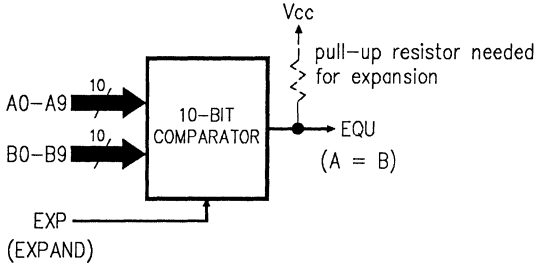


Figure 6.13b – Block diagram for EQU273.APL

```
TITLE 'APEEL FILE:  PEEL273 10-Bit Expandable Comparator
DESIGNER:  Robin Jigour
DATE:  10/15/87'

PEEL273


"PIN DEFINITIONS"

"Inputs"

A0        pin  1
A1        pin  2
A2        pin  3
A3        pin  4
A4        pin  5
A5        pin  6
A6        pin  7
A7        pin  8
A8        pin  9
A9        pin 10
B9        pin 11
B8        pin 13
B7        pin 14          "Pins 14-22, default polarity = pos.
B6        pin 15
B5        pin 16
B4        pin 17
B3        pin 18
B2        pin 19
B1        pin 20
B0        pin 21
EXP       pin 22


"Outputs

EQU       pin 23 = neg


EQUATIONS

EQU       = !(A0 & !B0 # !A0 & B0 #     "Compare A to B.
              A1 & !B1 # !A1 & B1 #     "If A /= B, EQU=0.
              A3 & !B3 # !A3 & B3 #     "If A  = B, EQU=1.
              A4 & !B4 # !A4 & B4 #
```

```
              A5 & !B5 # !A5 & B5 #
              A6 & !B6 # !A6 & B6 #
              A7 & !B7 # !A7 & B7 #
              A8 & !B8 # !A8 & B8 #
              A9 & !B9 # !A9 & B9)

Enable EQU = !EXP #                    "If EXP=0 enable EQU, if=1 then
          A0 & !B0 # !A0 & B0 #     "compare A to B.
          A1 & !B1 # !A1 & B1 #     "If A /= B, EQU is enabled and = 0.
          A3 & !B3 # !A3 & B3 #     "If A  = B, EQU is disabled so
          A4 & !B4 # !A4 & B4 #     "that EQU=1 via pull-up resistor.
          A5 & !B5 # !A5 & B5 #
          A6 & !B6 # !A6 & B6 #
          A7 & !B7 # !A7 & B7 #
          A8 & !B8 # !A8 & B8 #
          A9 & !B9 # !A9 & B9


TEST_VECTORS
(A9 A8 A7 A6 A5 A4 A3 A2 A1 A0  B9 B8 B7 B6 B5 B4 B3 B2 B1 B0  EXP ->EQU)
  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0  0  0  0   0  ->H
  0  0  0  0  0  0  0  0  0  1   0  0  0  0  0  0  0  0  0  0   0  ->L
  1  0  1  0  1  0  1  0  1  0   1  0  1  0  1  0  1  0  1  0   0  ->H
  1  0  1  0  1  0  1  0  1  0   0  1  0  1  0  1  0  1  0  1   0  ->L
  1  1  1  1  1  1  1  1  1  1   1  1  1  1  1  1  1  1  1  1   0  ->H
  1  1  1  1  1  1  1  1  1  1   1  1  1  1  1  1  1  1  1  0   0  ->L
  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0  0  0  0   1  ->Z
  0  0  0  0  0  0  0  0  0  1   0  0  0  0  0  0  0  0  0  0   1  ->L
  1  0  1  0  1  0  1  0  1  0   1  0  1  0  1  0  1  0  1  0   1  ->Z
  1  0  1  0  1  0  1  0  1  0   0  1  0  1  0  1  0  1  0  1   1  ->L
  1  1  1  1  1  1  1  1  1  1   1  1  1  1  1  1  1  1  1  1   1  ->Z
  1  1  1  1  1  1  1  1  1  1   1  1  1  1  1  1  1  1  1  0   1  ->L
```

Appendix A: Editor Commands

The PDK-1 provides two methods for controlling the editor: Keyboard commands, or pull-down menus. These commands are summarized in the sections that follow. For a step-by-step guide to using the editor, please refer to section 4.

Keyboard Commands

The editor can be driven with commands issued from the keyboard. The commands are given by striking certain characters on the keyboard while holding down the Control key. In the sequences shown below, the "^" character represents the **[Ctrl]** key. For example, to execute the command **^K^B**, hold down the **[Ctrl]** key while striking **K**, then **B.**

The keyboard commands of the PEEL editor are like those used by WordStar. For example, the "E D X S Diamond" (see figure A-1) can be used like the numeric key pad to move the cursor around the screen while the Control key is held down. Similarly, the letters adjacent to this diamond allow you to move the cursor from word to word, to scroll up and down through your file, etc

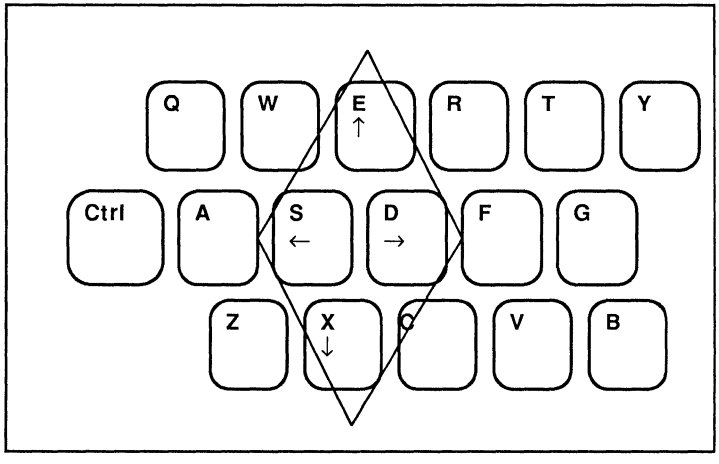


Figure A-1. "E D X S Diamond"

Control Command	Key Pad Command	Description
^A		Left word
^S	left arrow	Left character
^D	right arrow	Right character
^F		Right word
^E	up arrow	Up line
^X	down arrow	Down line
^R	[PgUp]	Up page
^C	[PgDn]	Down page
^W		Scroll up
^Z		Scroll down
[Enter]		New line
^N		Insert line
^G	[Del]	Delete character
^I		Tab
^T		Delete word
^Y		Delete line
^B		Reformat paragraph
^U		Abort current command
^J		Home or Beg/end of line

To enter a character using its ASCII number, hold the **[ALT]** key down while typing in the ASCII number on the key-pad.

Note that the keyboard command sequences do not work while in the pulldown menu mode

Enter/Exit pull-Down Menu Mode

The PDK also provides a convenient system of pull-down menus to drive the editor.

General Command	Pulldown Command	Description
[F10]		Enter pulldown window mode
[Esc]	twice	Exit from pulldow window mode to edit mode

Block Functions

Control Command	Pulldown Command	Description
^K^B	Begin	Begin block
^K^K	End	End block
^K^C	Copy	Copy block
^K^V	Move	Move block
^K^R	Read	Read block
^K^W	Write	Write block
^K^Y	Delete	Delete block
^K^H	Hide/show	Hide/display
	Spell/check	Spell check

Search Functions

Control Command	Pulldown Command	Description
^Q^F	Find	Find
^Q^A	Find/Replace	Find and replace
^L	Next	Next Find or Find/Replace

Go to Functions

Control Command	Pulldown Command	Description
^Q^R	Top of file	Cursor to top of file
^Q^C	End of file	Cursor to end of file
^Q^B	Begin block	Cursor to top of block
^Q^K	End block	Cursor to end of block
^O^N	Line	Go to line number
^O^I	Column	Go to column number
^Q^J	Go marker	Go to marker number
^K^M	Set marker	Set marker number

Text format Functions

Control Command	Pulldown Command	Description
^V	Insert	Toggle insert mode
^O^W	Word wrap	Toggle word wrap
^Q^I	Auto indent	Toggle auto indent
^O^L	Left margin	Set left margin
^O^R	Right margin	Set right margin
^K^T	Tab size	Set tab spacing
^O^S	Undo limit	Set undo limit
	Save settings	Save text format settings

Window Functions

Control Command	Pulldown Command	Description
^O^G	Select	Select second window
^O^O	Open	Open second window
^O^Y	Close	Close second window

File Functions

Control Command	Pulldown Command	Description
^K^D	Open	Close current and Open file
^K^Q	Close	Close file
^K^S	Save	Save file
	save As	Save file under another name
	Directory	List directory
	Logged dir	Log directory of disk
	Print	Print file
	copY	Copy file
	Rename	Rename file
	Erase	Erase file
^K^X		
^K^Q	Quit	Abandon file

Appendix B: Programming Support

Support as of Q2 1989. For additional information contact ICT or specific PLD programmer manufacturer.

International CMOS Technology (408) 434-0678

	PDS-1 PEEL™ Development System ❖	
ICT Part Number	**Software**	**Adapter**
PEEL18CV8	PDS-1 Software Version V1.20 or greater	Call ICT for PLCC Adapter
PEEL20CG10	PDS-1 Software Version V3.20 or greater	
PEEL22CV10		
PEEL22CV10Z		
PEEL153	PDS-1 Software Version V2.20 or greater	
PEEL253		
PEEL173	PDS-1 Software Version V2.00 or greater	
PEEL273		

Adams-MacDonald (408) 373-3607

	Sprint Plus ❖
ICT Part Number	**Software**
PEEL18CV8	Version V3.10
PEEL153	Version V3.20
PEEL253	

Advin Systems (408) 984-8600

	Sailor-PAL ❖
ICT Part Number	**Software**
PEEL18CV8	Version 8.0
PEEL22CV10	Version 9.4

BP Microsystems (800) 225-2102

	PLD 1100 ❖
ICT Part Number	**Software**
PEEL18CV8	Version 1.07 or greater
PEEL153	
PEEL253	
PEEL173	
PEEL273	
PEEL22CV10	Please call BP Microsys

❖ Systems and updates identifed by this symbol have been qualified by ICT for their ability to program the devices listed.
Contact ICT for information on systems not listed in this table.

Digelec (800) 367-8750, in California call (818) 887-3755

	Model 860 Programmer
ICT Part Number	**Software**
PEEL18CV8	Version A-1.3 or greater ❖
PEEL153	Version A-1.4 or greater
PEEL173	Version A-1.4 or greater
PEEL253	Version A-1.4 or greater
PEEL273	Version A-1.4 or greater
PEEL22CV10	Please call Digelec

DATA I/O (800) 426-1045

	Unisite 40 ❖		
ICT Part Number	**Software**	**Family**	**Pinout**
PEEL18CV8	V1.4 or greater	8D	3A
PEEL20CG10	V2.3 or greater		56
PEEL22CV10			28
PEEL22CV10Z	V2.2 or greater		A3
PEEL153			65
PEEL253			85
PEEL173			76
PEEL273			86

	Model 29B Programmer ❖			
ICT Part Number	**Module**	**Adapter/ Firmware**	**Family**	**Pinout**
PEEL18CV8	LogicPak™ V04	303A-011A /V02 or greater	8D	3A
PEEL20CG10		303A-011A /V09 or greater		56
PEEL22CV10		303A-011A/ V06 or greater		28
PEEL22CV10Z				A3
PEEL153		303A-011A/ V06 or greater		65
PEEL253				85
PEEL173		303A-011A/ V04 or greater		76
PEEL273				86

	Model 60A ❖		*Model 60H* ❖		
ICT Part Number	**Adapter**	**Firmware**	**Firmware**	**Family**	**Pinout**
PEEL18CV8	360A-001	Version V11	Version V12	8D	3A
PEEL20CG10		Version V14 (Summer 1989)	Version V14 (Summer 1989)		56
PEEL22CV10					28
PEEL22CV10					A3
PEEL153		Please call Data I/O	Please call Data I/O		65
PEEL253					85
PEEL173					76
PEEL273					86

INLAB (303) 460-0103

	Model 28U ❖
ICT Part Number	**Firmware**
PEEL18CV8	Version 11.02 or greater
PEEL22CV10	
PEEL153	
PEEL253	
PEEL173	
PEEL273	

Kontron Electronics (800) 227-8834

	MPP-80S Portable Programmer with Universal Module UPM/B	*EPP-80 Base Programmer with Universal Module UPM/B*
ICT Part Number	**Firmware**	**Firmware**
PEEL18CV8	Version 2.1 or greater	Version 2.1 or greater
PEEL153	Please call Kontron	Please call Kontron
PEEL253		
PEEL173	Version 2.2 or greater	Version 2.2 or greater
PEEL273		

Logical Devices (305) 974-0975

	ALLPRO Programmer ❖
ICT Part Number	**Software**
PEEL18CV8	Version 1.44 or greater ❖
PEEL20CG10	Please call Logical Devices
PEEL22CV10	Version 1.47c or greater
PEEL22CV10Z	Please call Logical Devices
PEEL153	Version 1.47c or greater
PEEL253	Please call Logical Devices
PEEL173	Version 1.45 or greater ❖
PEEL273	

Stag Microsystems (408) 988-1118

	PPZ Programmer		*ZL-30A Programmer*	
ICT Part Number	**Module**	**Firmware**	**Module**	**Firmware**
PEEL18CV8	ZM2200	Version 34	30A800	Version 30A26
PEEL153		Version 37		Version 30A32
PEEL253				
PEEL173				
PEEL273				
PEEL22CV10		Please call Stag		Please call Stag

System General (Taiwan) 886-2-7212613

	SGUP-85 ❖
ICT Part Number	**Firmware**
PEEL18CV8	Version 3.0 or greater
PEEL22CV10	
PEEL22CV10Z	Version 3.1 or greater
PEEL153	
PEEL253	
PEEL173	
PEEL273	

INTERNATIONAL CMOS TECHNOLOGY, INC.

APPENDIX C:

PDS-1 PEEL™ Development System

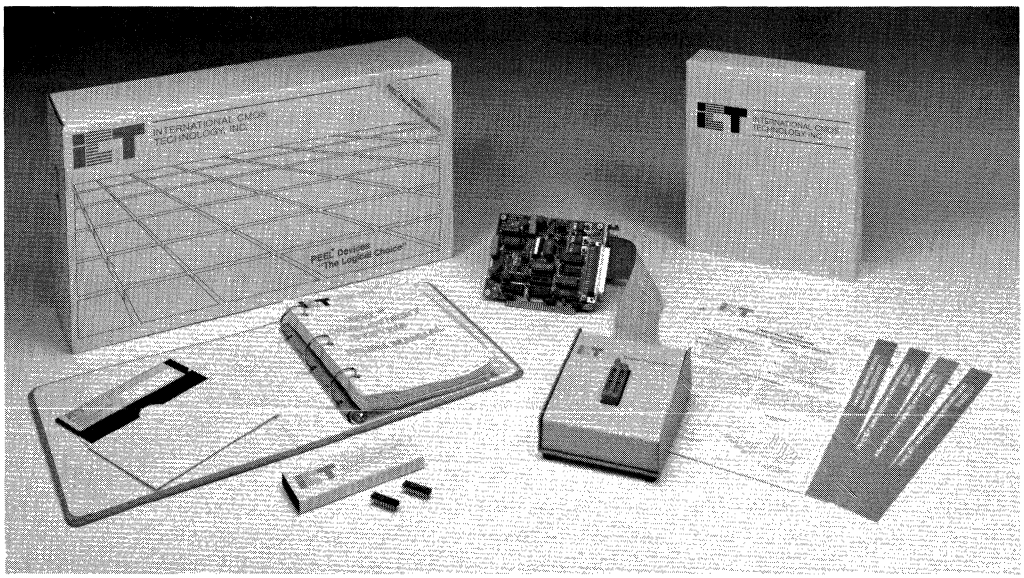
Features

- **Development System for PEEL Devices**
  - — Editor, logic assembler, translator, programmer, and tester all in one system
  - — Runs on PC-compatible computers
- **Standard PLD Programmer Functions**
  - — Program, Load, Verify, Secure
- **APEEL™ Boolean Logic Assembler**
  - — Supports all features of PEEL devices
  - — "PALASM*-like" equations
  - — "ABEL*-like" macro cell definitions
  - — Logic simulation
- **Translates Standard PLDs to PEEL Devices**
  - — Loads PLD or reads JEDEC file
  - — Automatically translates to PEEL device
- **Built-in File Editor**
  - — Edit source, JEDEC, or test-vector files
- **Enhanced Logic-Test Capabilities**
  - — Design verification with in socket
  - — Special features: step, loop, capture,
- **Expandable and Accessible**
  - — New support with software updates
  - — No copy protection

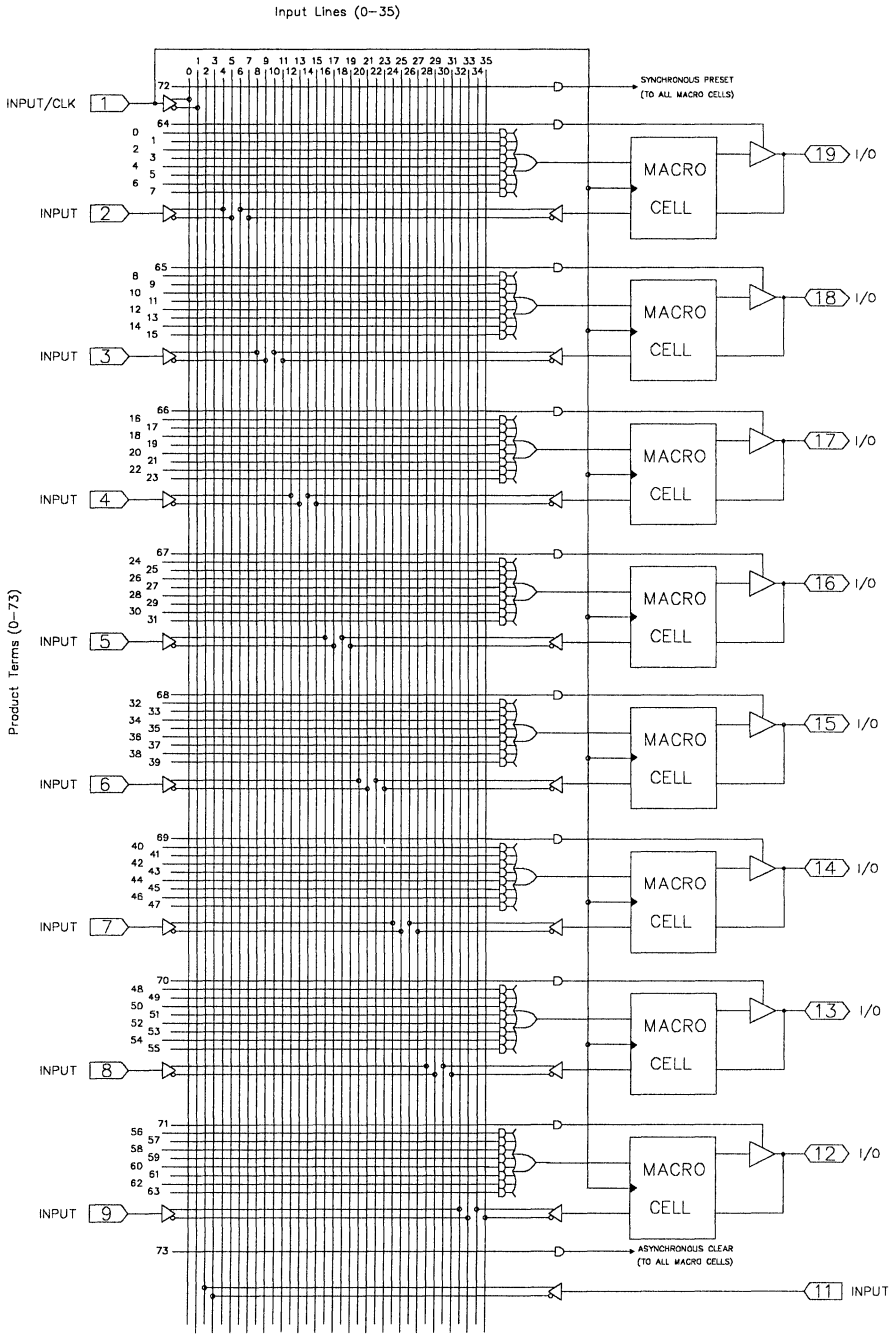
General Description

The PEEL™ Development System is a powerful, yet inexpensive, PC-based system for designing with PEEL (Programmable Electrically Erasable Logic) devices. The PDS-1 is a personal PLD work-station providing everything needed to implement your logic designs from concept to silicon. Several options for desiging with PEEL devices are available with the PDS-1. For example, an existing PLD design (i.e., PAL, GAL or EPLD JEDEC file) can be automatically translated and programmed into a PEEL device. Additionally, the translation capability allows you to use your present PLD logic assembler or compiler to design with PEEL devices.

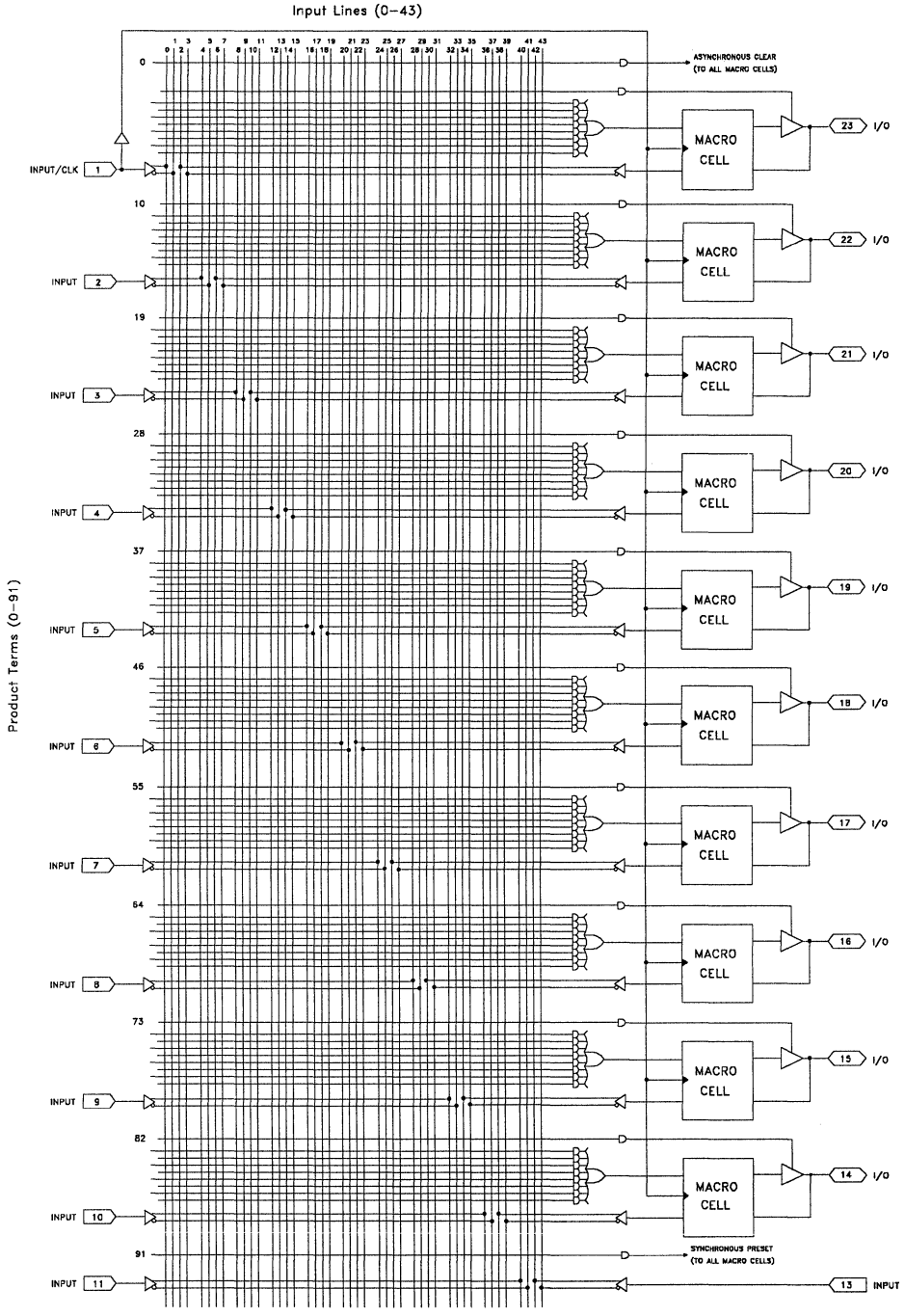
To fully support the advanced features of PEEL devices, the PDS-1 also provides the tools needed to design from start to finish, including a built-in word processor for design entry and editing, the APEEL™ boolean-logic assembler, a complete PEEL-device programmer and enhanced logic tester. The capabilities of the software-controlled programmer will be expanded as new devices are released by ICT. Registered owners are enrolled in the ICT software update service and receive programmer/development-software updates.



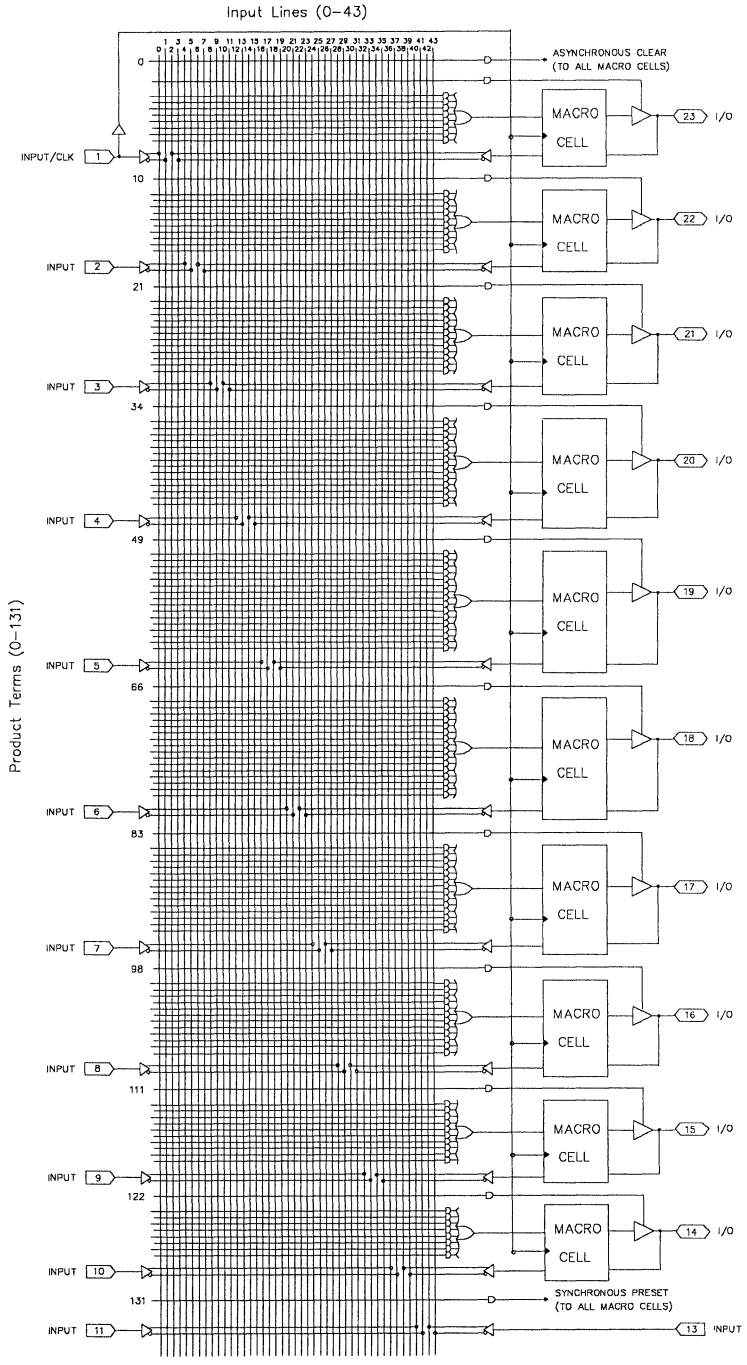
PDS-1, PEEL Development System



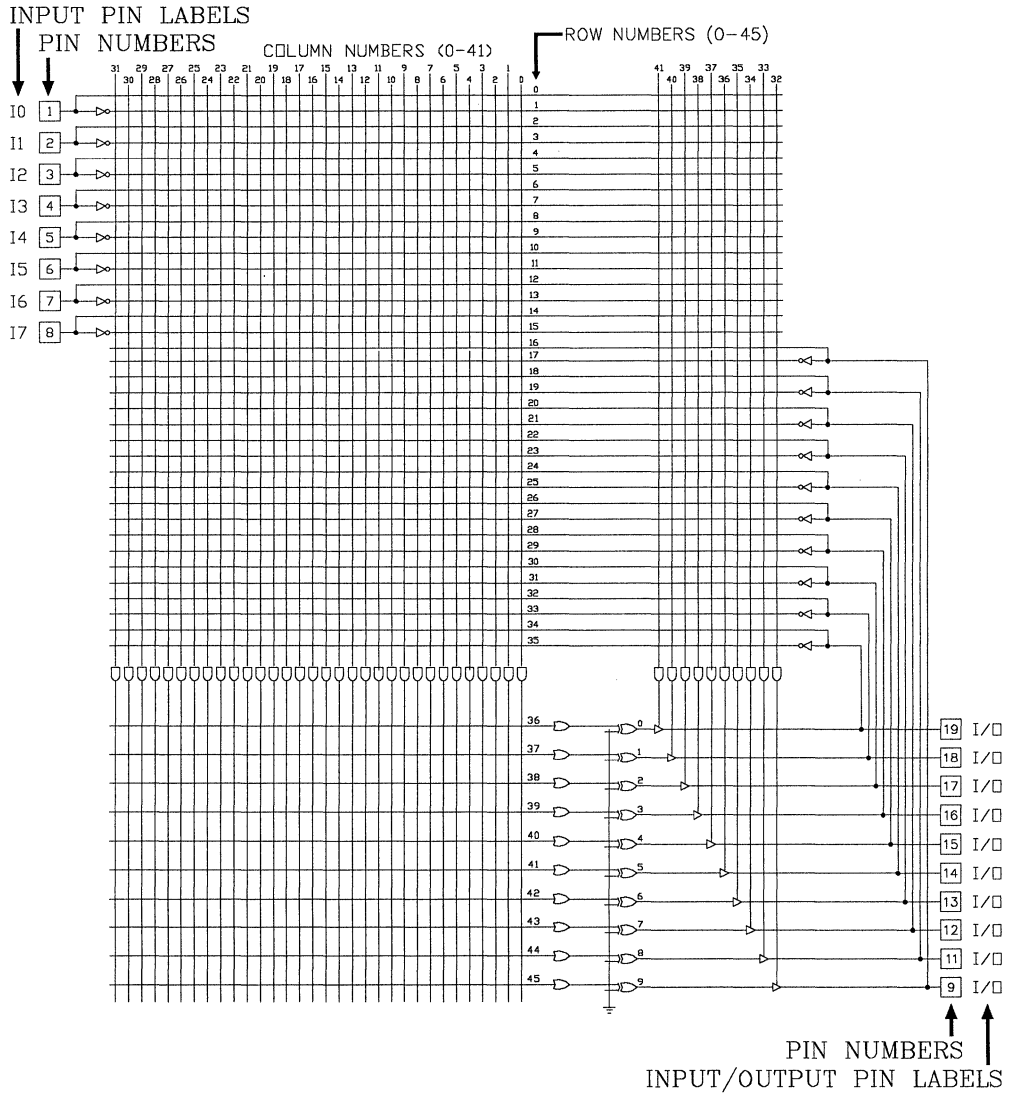
PEEL18CV8 Logic Array Diagram



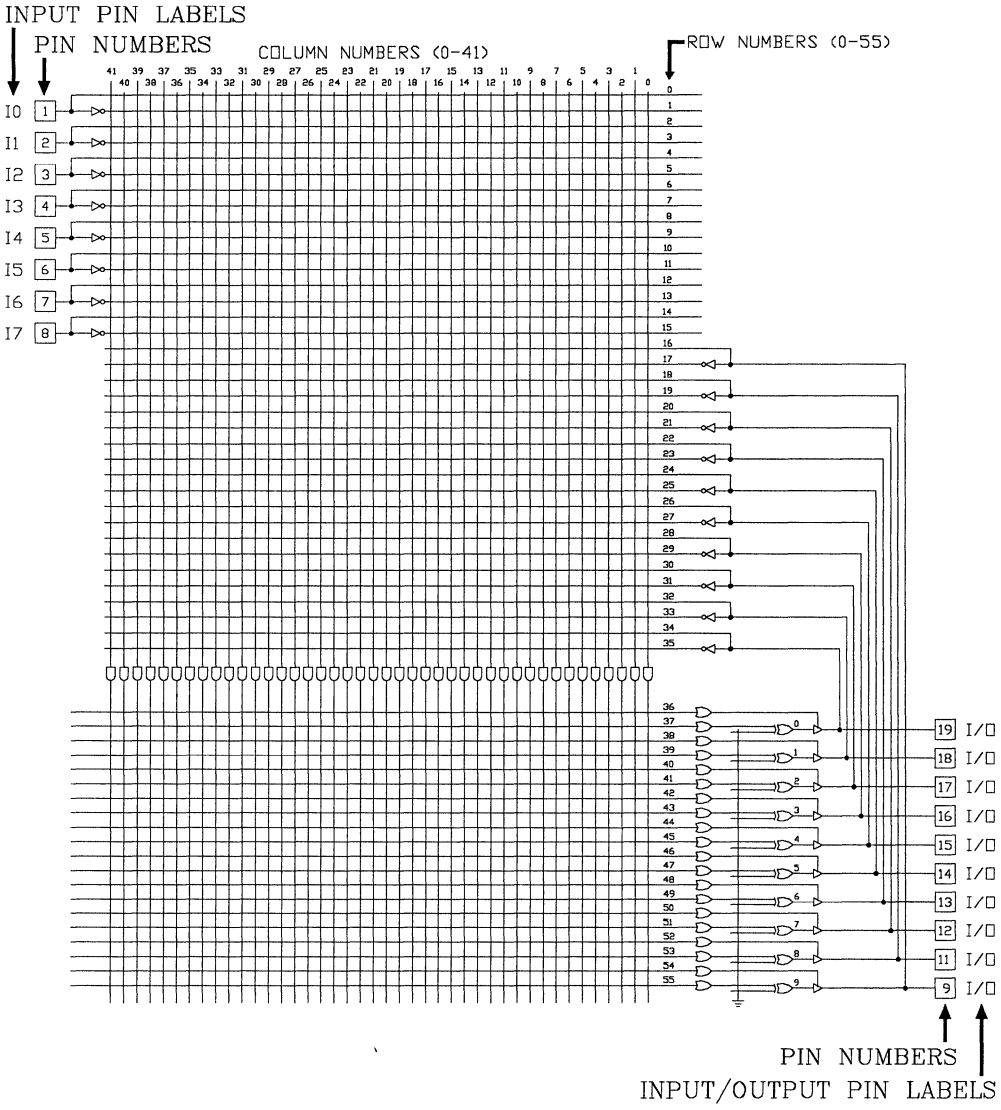
PEEL20CG10 Logic Array Diagram



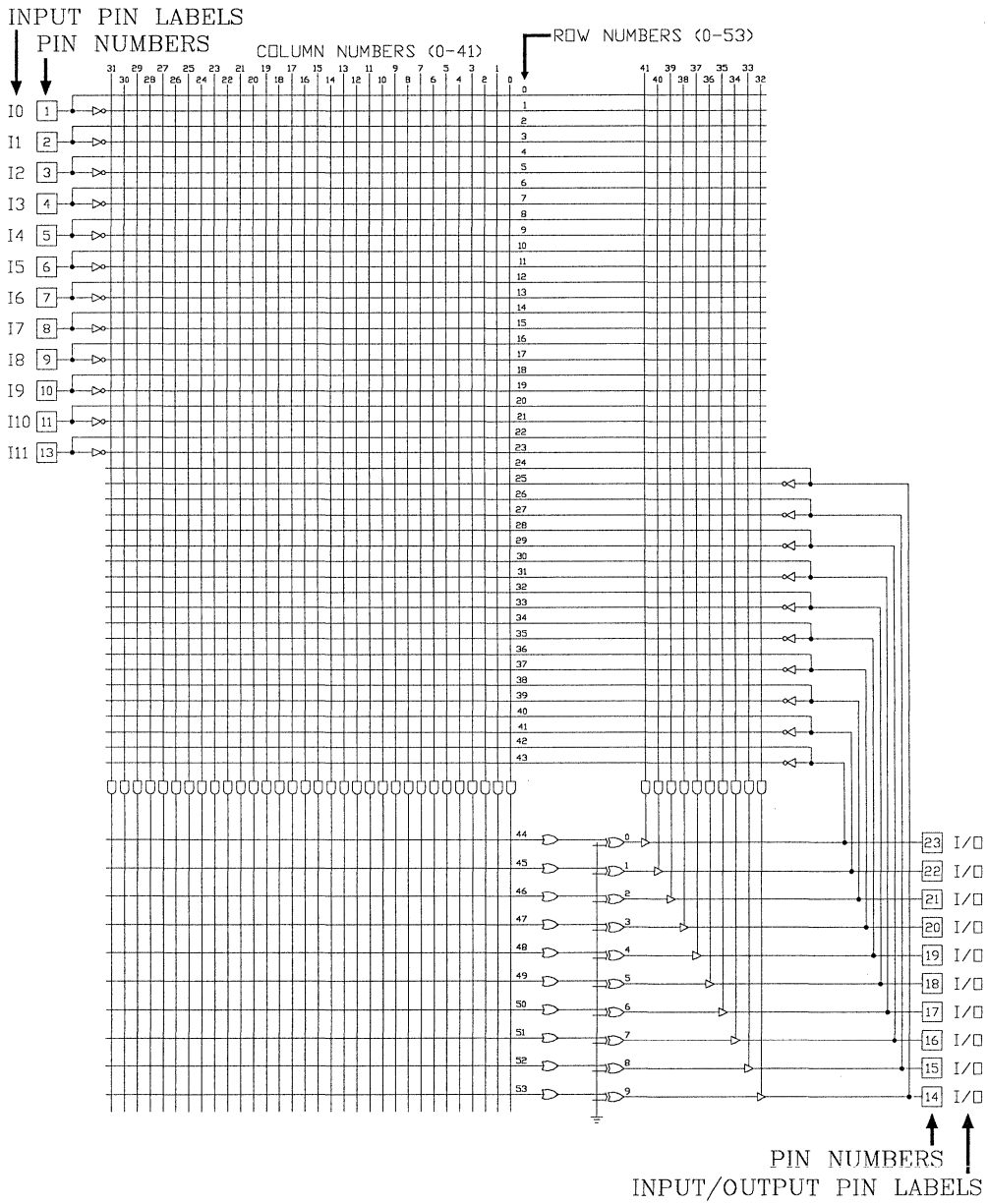
PEEL22CV10 / PEEL22CV10Z Logic Array Diagram



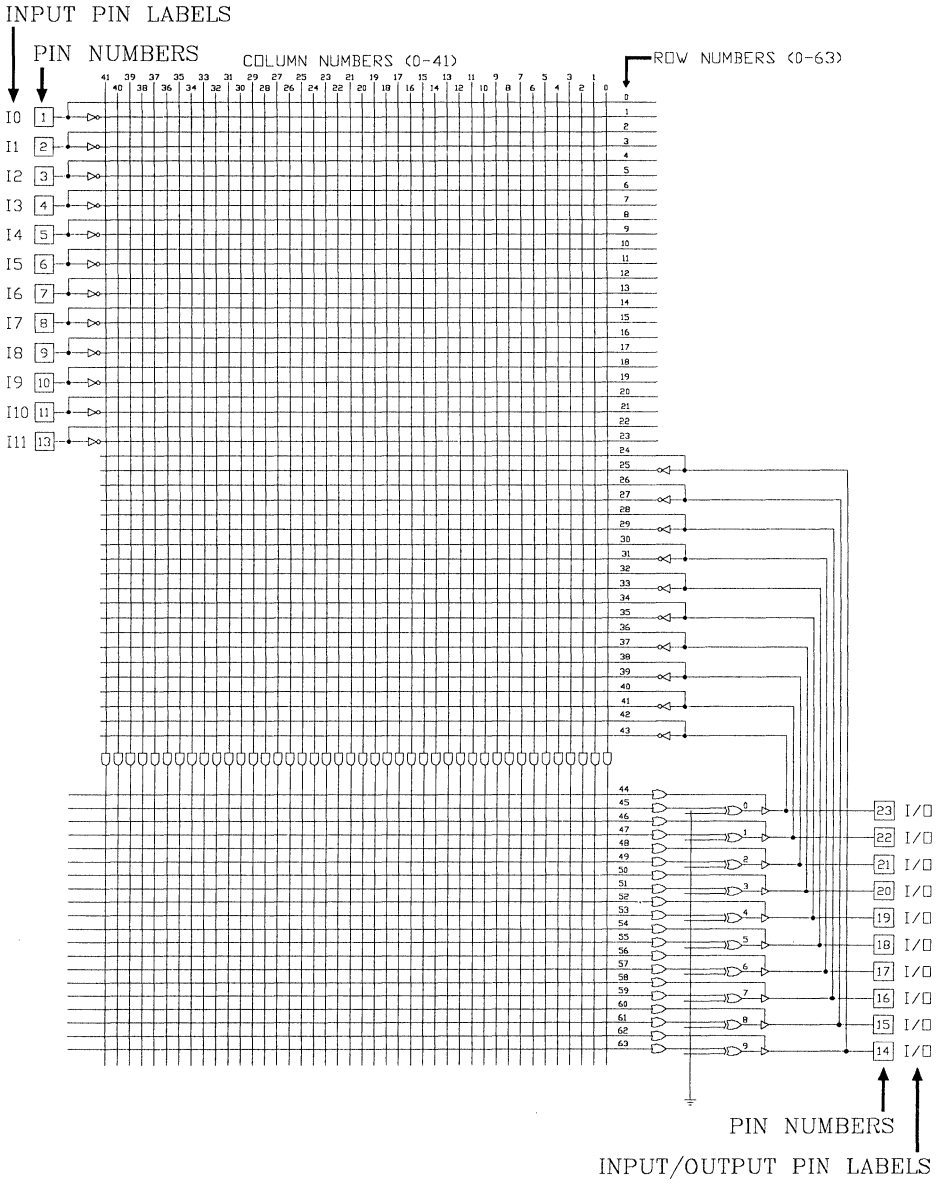
PEEL153 Logic Array Diagram



PEEL253 Logic Array Diagram



PEEL173 Logic Array Diagram



PEEL273 Logic Array Diagram

ICT North American Sales

Headquarters

2125 Lundy Avenue
San Jose, CA 95131
TEL..... (408) 434-0678
FAX..... (408) 434-0688
TWX..... (910) 997-1531

Western Area Sales

2125 Lundy Avenue
San Jose, CA 95131
Tel:..... (408) 434-0678
Fax:..... (408) 434-0688

Central Area Sales

Barrington Pointe
2300 N. Barrington Road
Suite 452
Hoffman Estates, IL 60195
Tel:..... (312) 490-5389
FAX:..... (312) 884-9423

Eastern Area Sales

720 East Main Street
Moorestown, NJ 08057
Tel:..... (609) 722-0442
FAX:..... (609) 234-2308

Sales Representatives

Alabama

Technology Marketing Associates
Huntsville..... (205) 883-7893

Arizona

Oasis Sales, Inc. . (602) 277-2714

California

Epsilon Marketing
Northern Calif..... (408) 241-8111

DynaRep, Inc.
Los Angeles ..... (805) 527-0072
Orange County:.... (714) 545-3255

Littlefield & Smith
San Diego ..... (619) 455-0055

Colorado

Electrodyne ..... (303) 695-8903

Connecticut

Datatrack Engineering, Inc.
..... (203) 271-1311

Florida

Technology Marketing Associates
Orlando..... (407) 857-3760
Deerfield Beach.... (305) 427-1090
Largo ..... (813) 541-1591

Georgia

Technology Marketing Associates
..... (404) 446-3565

Indiana

Electronic Sales & Engineering
..... (317) 849-4260

Illinois

Martan, Inc. ..... (312) 303-5660

Iowa

Contact ICT ..... (312) 490-5389

Kansas

Contact ICT ..... (312) 490-5389

Maine

Compass Technology, Inc.
..... (617) 933-3336

Maryland

Conroy Sales Inc. (301) 296-2444

Massachusetts

Compass Technology, Inc.
..... (617) 933-3336

Michigan

R.C. Nordstrom & Company
..... (313) 559-7373

Minnesota

George Russell & Associates
..... (612) 854-1166

Missouri

Contact ICT ..... (312) 490-5389

New Hampshire

Compass Technology, Inc.
..... (617) 933-3336

New Jersey

Nexus Technology Sales, Inc.
North New Jersey. (201) 947-0151

Tritek Sales
South New Jersey (609) 429-1551

New Mexico

Oasis Sales, Inc. .. (602) 277-2714

New York

Nexus Technology Sales, Inc.
Metro NY ..... (201) 947-0151

Pi'Tronics
NY State ..... (315) 455-7346

North Carolina

Quantum Marketing
..... (919) 846-5728

Ohio

Thompson and Associates
Dayton ..... (513) 435-7733
Beachwood..... (216) 831-6277
Orient..... (614) 877-4304

Oregon

Contact ICT ..... (408) 434-0678

Pennsylvania

Tritek Sales ..... (609) 429-1551

South Carolina

Quantum Marketing
..... (704) 523-8822

Tennessee

Technology Marketing Associates
..... (205) 883-7893

Texas

T.L. Marketing
Dallas..... (214) 484-6800
Austin..... (512) 453-4586
Houston ..... (713) 589-2763

Utah

Utah Component Sales, Inc.
..... (801) 268-8440

Vermont

Compass Technology, Inc.
..... (617) 933-3336

Washington

Contact ICT ..... (408) 434-0678

Wisconsin

Martan, Inc. ..... (414) 241-4955

Canada

Bytewide Marketing
Montreal..... (514) 636-4121
Ottawa ..... (613) 728-0031
Toronto ..... (416) 675-1868

ICT North American Distributor Network

Alabama

Marshall Industries
Huntsville .....(205) 881-9235

Arizona

Marshall Industries
Phoenix.....(602) 496-0290

California

Marshall Industries
El Monte.....(818) 459-5500
Chatsworth.....(818) 407-4100
Irvine.....(714) 458-5301
San Diego.....(619) 578-9600
Rancho Cordova...(916) 635-9700
Milpitas.....(408) 942-4600

Western Microtechnology
Agoura Hills .....(818) 356-0180
Orange.....(714) 637-0200
San Diego.....(619) 453-8430
Saratoga .....(408) 725-1660

Colorado

Marshall Industries
Thornton.....(303) 451-8383

Connecticut

Marshall Industries
Wallingford.....(203) 265-3822

Florida

Marshall Industries
Altamonte Springs (407) 767-8585
Ft. Lauderdale.....(305) 977-4880
St. Petersburg.....(813) 573-1399

Georgia

Marshall Industries
Norcross.....(404) 923-5750

Illinois

Marshall Industries
Schaumburg.....(312) 490-0155

Indiana

Marshall Industries
Indianapolis ..... (317) 297-0483

Kansas

Marshall Industries
Lenexa..... (913) 492-3121

Maryland

Marshall Industries
Silver Springs ..... (301) 622-1118

Massachusetts

Marshall Industries
Wilmington..... (508) 658-0810

Western Microtechnology
Burlington ..... (617) 273-2800

Michigan

Marshall Industries
Livonia ..... (313) 525-5850

Minnesota

Marshall Industries
Minneapolis ..... (612) 559-2211

Missouri

Marshall Industries
Bridgeton ..... (314) 291-4650

New Jersey

Marshall Industries
Mt. Laurel ..... (609) 234-9100
Fairfield..... (201) 882-0320

New York

Marshall Industries
Hauppauge..... (516) 273-2424
Johnson City..... (607) 798-1611
Rochester..... (716) 235-7620

North Carolina

Marshall Industries
Raleigh ..... (919) 878-9882

Ohio

Marshall Industries
Dayton.....(513) 898-4480
Westerville.....(614) 891-7580
Solon.....(216) 248-1788

Oregon

Marshall Industries
Beaverton.....(503) 644-5050

Western Microtechnology
Beaverton.....(503) 629-2082

Pennsylvania

Marshall Industries
Pittsburgh.....(412) 963-0441

Texas

Marshall Industries
El Paso.....(915) 593-0706
Carrollton.....(214) 233-5200
Houston.....(713) 895-9200
Austin .....(512) 837-1991
Brownsville .....(512) 542-4589

Insight Electronics
Austin .....(512) 467-0800
Ft. Worth .....(817) 338-0800
Houston.....(713) 448-0800
Richardson.....(214) 783-0800

Utah

Marshall Industries
Salt Lake City.....(801) 485-1551

Washington

Marshall Industries
Bothell .....(206) 486-5747

Western Microtechnology
Redmond.....(206) 881-6737

Wisconson

Marshall Industries
Waukesha .....(414) 797-8400

Canada

Marshall Industries
Ontario .....(416) 674-2161
Montreal .....(514) 848-9112
Ottawa.....(613) 564-0166

ICT International Sales Representatives

ICT European Sales

EMS, Ltd.
Cardinal Point
Newall Road
Heathrow TW6 2EX
England
Contact: Bob Critchlow
Tel: 44-1-759-4192
FAX: 44-1-564-7625

West Germany

United Semiconductor Engineering
.....(49) 89 339292

Alfred Neye Enatechnik
.....(49) 4106 6120

France

ASAP ..... (33) 16 1 30 43 8233

MISIL..... (33) 1 45 60 00 21

United Kingdom

Sequoia Technololgy Limited
.....(44) 0734 311822

The Netherlands

Acal Auriema .....(31) 40 816565

Denmark

PRC Components (45) 6 62 37 99

Norway

Abemi A/S 472 807200

Finland

Abemi Oy Ab 3580 755 6722

Italy

MicroELit S.P.A. ..... (39) 4817900

Belgium

Acal Auriema..... (32) 2523 6295

Spain

Anatronic, S.A. 242 44 55

Switzerland

Inteltro AG 01 741 41 21

Sweden

Titan.....08 754 9175

Israel

SegTec 972-3-556-7548

ICT Far East Sales

Excel Associates Ltd.
1502 Austin Tower,
22-26A Austin Ave.
Tsimshatsw, Kowloon
Hong Kong
Contact: Mauri Morin
Tel: 03 7210900
FAX: 03 696826

Hong Kong, Korea, Singapore,

Excel Associates..... 03 7210900

Tiawan

Jatron Electronics, Inc.
.....02-307-0152

Sertek International Inc.
.....02-501-0055

Australia

Reptechnic..... 621-436-3422

Japan

Motono International Co., Ltd.
.....03-404-5750

Tokyo Denshi Hanbai Co., Ltd.
.....03-348-3401

Sanei Electronic Industry Co., Ltd.
.....426-44-7338

Paltek Corporation 03-707-5455

Shin-Nichi Electronics Co., Ltd.
.....03-833-2431

INTERNATIONAL CMOS TECHNOLOGY, INC.

2125 Lundy Avenue
San Jose, California 95131

Telephone (408) 434-0678
TWX (901) 997-1531
FAX (408) 434-0688

2125 Lundy Avenue . San Jose, CA 95131 (408) 434-0678