LogiCaps
Version 1.6
September 1987 P25-01800-02

Changes are made periodically to the information contained in this manual. These changes will be incorporated into subsequent editions.

Altera Corporation
3525 Monroe Street
Santa Clara, CA 95051
(408) 984-2800
TELEX: 888496

Read This First...

Your LogiCaps documentation consists of two main parts:

- ***User Guide***
- ***Reference Guide***

The ***User Guide*** contains the following sections:

Introduction provides a functional description of LogiCaps.

Installation provides software installation instructions.

Getting Started provides detailed descriptions of the screen displays, windows, and mouse and keyboard commands.

Design Guidelines lists a number of design guidelines and restrictions for designing with LogiCaps and A+PLUS.

LogiCaps Tutorial guides you step by step through the creation of a logic design, and shows how to submit the design to A+PLUS, plot it, print it, and program the EPLD.

MacroFunction Tutorial provides a general description of Altera MacroFunctions and shows how to use them in a design.

The ***Reference Guide*** contains the following sections:

LogiCaps Commands provides a detailed description of each LogiCaps command.

Standard MacroFunctions lists the 30 standard MacroFunctions and provides the symbol block diagram, syntax description, function table, and logic schematic for each.

In addition, the manual contains a section with *Error Messages*, as well as an *Appendix*, *Glossary* , and *Index*. At the back of the manual, you find a *Customer Comment Form* and a *Problem Report Card*.

Manual Updates

Altera documentation is updated with Change Pages, Section Reprints, and a READ.ME file.

Change Pages are issued for minor changes to the manual. New information is identified with vertical change bars in the margins next to the changed text. In addition, the date of issue is printed at the bottom of each page.

Section Reprints are issued if a section requires a substantial number of changes. The date of issue is indicated at the bottom of each page.

A **READ.ME File** is provided on the LogiCaps **INSTALL** diskette. This file contains information about recent changes to the software that are not yet reflected in the manual.

Printing Conventions

The following notational conventions are used throughout this manual:

Times Bold	—	all A+PLUS commands, prompts, and messages
	—	all user input, including keyboard keys
Times Light	—	most file output as displayed on screen
Helvetica Italics Bold	—	all references to Altera manual titles
Helevetica Italics Light	—	all references to sections within Altera manuals
☞	—	information that requires special attention

Contents

User Guide

Section 1: Introduction

Section 2: Installation

Section 3: Getting Started

Section 4: Design Guidelines

Section 5: LogiCaps Tutorial

Section 6: MacroFunction Tutorial

Section 7: LogiCaps Commands

Section 8: Standard MacroFunctions

Section 8: Standard MacroFunctions (Continued)

Illustrations

Figure		Page

Tables

Table		Page
4-1	Legal Pin Name and Node Name Characters in LogiCaps	4-4
8-1	MacroFunction Categories	8-5

SECTION 1

Introduction

LogiCaps, Altera's schematic capture package, offers you a number of unique features that make its use simple and fast. To assist you in becoming rapidly proficient in the use of LogiCaps, this ***User Guide*** provides the following information:

- A functional description
- Installation instructions
- Instructions for invoking LogiCaps and a thorough overview of the individual system components
- A list of design guidelines
- Two tutorial sessions guiding you through the use of LogiCaps and MacroFunctions

Why You Should Use LogiCaps

LogiCaps is truly user-friendly. If you have any experience designing a circuit with schematic symbols, you will quickly feel comfortable using LogiCaps. Even if you are relatively new to designing circuits with a schematic capture interface, you will find this method to be most efficient. Regardless of your level of experience, you will appreciate the large number of mouse functions, the hierarchical menu structure, the split-screen facility, and the simple mnemonic commands.

Before starting with LogiCaps, you should familiarize yourself with the Altera-supplied primitives described in the ***A+PLUS Reference Guide***, and you should also read the introductory chapters (Sections 1 to 3) in the ***A+PLUS User Guide***.

Functional Description

The LogiCaps schematic capture package enables you to create, edit, save, and retrieve schematic drawings with simple mouse and keyboard commands. The primitive library (shown in *Appendix A*) provides the schematic primitives that must be used exclusively when you are drawing schematics with LogiCaps. The Standard MacroFunction library (also shown in *Appendix A*) provides 30 Altera-supplied MacroFunctions that you may use in your schematics. The complete TTL MacroFunction library is available in the ***TTL MacroFunctions*** manual. The ***ADLIB*** manual provides information on how to create your own custom-made MacroFunctions.

LogiCaps outputs an Altera Design File (ADF) that can be directly entered into the Altera Design Processor (ADP). The ADP processes the ADF and converts it into a standard JEDEC file that is then entered into LogicMap II to program an Altera EPLD.

Figure 1-1 shows a block diagram of LogiCaps and A+PLUS.

The most important features of the LogiCaps schematic capture interface are as follows:

- Menu-Driven/Mouse-Driven Design Entry
- Multi-Level Zoom Capability
- Dual Editing Windows
- Orthogonal Rubberbanding
- Tag-and-Drag Editing
- On-Line Help Documentation
- Direct ADF Output for A+PLUS
- Plotter and Printer Interface Software
- Support for IBM Color/Graphics Monitor Adapter (CGA)
- Support for IBM Enhanced Graphics Adapter (EGA)
- Support for Hercules Graphics Card

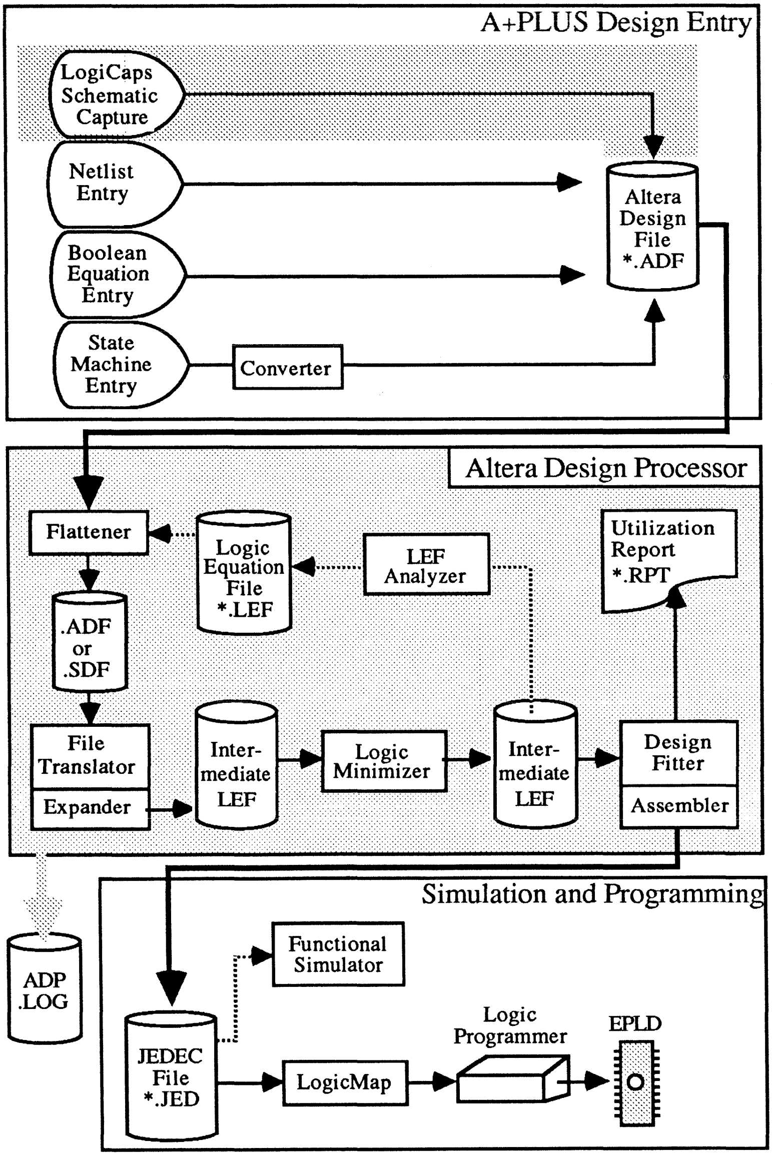


Figure 1-1. LogiCaps/A+PLUS Block Diagram

SECTION 2

Installation

This section provides the information needed to install LogiCaps. For detailed information on how to install the hardware required to program an Altera part, refer to the ***LogicMap II*** manual.

Hardware and Software Requirements

To run LogiCaps you need the following hardware and software products:

- IBM XT or AT and compatible computers (hard disk version)
- MS-DOS version 2.0 or a later version
- 640 Kbytes RAM
- IBM Color/Graphics Monitor Adapter (CGA)
  or:
  IBM Enhanced Graphics Adapter (EGA)
  or:
  Hercules Graphics Card
- A+PLUS software
- Altera Programming Card (optional)
- Logitech C7 or Mouse Systems PC mouse or compatibles (optional)

Mouse Installation

Connect your serial mouse to the COM1 or COM2 serial port of your computer as described in your mouse documentation. No software installation is required; LogiCaps contains its own mouse interface software; *you should not use the mouse drivers that are supplied with your mouse.*

You may want to create a batch file to set your mouse parameters manually. Note that you then must load these parameters manually each time before you call LogiCaps.

LogiCaps accepts the following parameters:

Port:	COM1 or COM2
Baud rate:	1200 baud
Report rate:	150
Buttons:	3
Protocol:	5b (five byte packed binary; Mouse Systems PC Mouse compatible)

For complete details consult the *User Manual* for your mouse.

Software Installation

If you have purchased the PLCAD-SUPREME, PLCAD4, or PLDS2 development system, you should follow the installation instructions described in *Installation* of the ***A+PLUS User Guide***.

If you have purchased the PLE40 development system, install your software according to the following instructions.

If You Have Purchased PLE40...

...you should have the following distribution diskettes:

- **INSTALL**
  Contains the installation procedures for all A+PLUS modules.
- **LOGICAPS**
  Contains the LogiCaps program.
- **UTILITIES**
  Contains the LogiCaps utilities.

Backups of Altera Distribution Diskettes

Before installing Altera software, you must make backups of all Altera distribution diskettes in case one of the distribution diskettes fails to work. Note that the copy-protected diskettes are specially formatted. You can copy the files from these diskettes onto a backup diskette, but you cannot start the copy-protected programs with the backup diskette. So, if you lose or damage an original copy-protected diskette, you must use the backup of that program and copy it onto a disk that has the special formatting.

☞ The DOS **DISKCOPY** command cannot be used to duplicate the distribution diskettes because they are copy-protected and **DISKCOPY** is unable to read them.

To make backups of all Altera distribution diskettes, you must go through the following steps:

1. Boot DOS.

2. Format a blank diskette for each distribution diskette with the DOS **FORMAT** command. (Refer to the DOS Manual.)

3a. If your system has one floppy disk drive and one hard disk drive, put an Altera distribution diskette into drive **A** and type:

    COPY A:*.* B: <Enter>

    You are prompted to replace the Altera distribution diskette with a formatted diskette.

3b. If your system has two floppy disk drives and one hard disk drive, put the Altera distribution diskette into drive **A** and the formatted diskette into drive **B** and type:

    COPY A:*.* B: <Enter>

4. Repeat step 3 for each Altera distribution diskette.

5. Store your backup diskettes in a safe place.

LogiCaps Installation

Once you have completed the following installation procedure, you will be able to run LogiCaps directly from your hard disk.

Before installing A+PLUS software on your hard disk, you must ensure that it has at least 1 Mbyte of free disk space on the hard disk and 640 Kbytes of RAM memory, otherwise installation will not be successful. Available space is verified with the DOS **CHKDSK** command. (For information regarding DOS, refer to *IBM Disk Operating System, Version 3.30 User's Guide*; *IBM Disk Operating System, Version 3.30 Reference;* and *IBM DOS Technical Reference*.)

IBM AT high density 1.2 Mbyte diskette drives are not compatible with the standard 360 Kbyte diskette drives. If you anticipate moving A+PLUS from an AT to an XT computer, you should use a 360 Kbyte diskette drive, if available, to perform the installation.

This procedure assumes that your hard disk is drive **C**. (If you have another hard-disk drive, substitute the appropriate letter.)

1. Boot the computer from the hard disk.

   Remove any write-protect tabs that may be on the distribution diskettes.

2. Put the Altera-provided **INSTALL** diskette into drive **A** or **B** and enter:

   A:INSTALL <Enter> (if you use drive A)
   B:INSTALL <Enter> (if you use drive B)

   The program will perform some basic checks after which the Installation Main Menu, shown in Figure 2-1, is displayed.

```
            A L T E R A Programmable Logic User System
                  Software Configuration Program
               Copyright (C) 1987 Altera Corporation
 version 5.0

       Main Menu

 [1]  Install Software
 [2]  De-Install Software
 [3]  Change Programming Hardware & A+PLUS menu configuration.

 Press a number key to select an option:
 Press <Esc> to exit to DOS.
```

Figure 2-1. Installation Main Menu

Item [1]	Guides you through A+PLUS installation.
Item [2]	De-installs the software from the current system so that you can move the software to another computer.
Item [3]	Allows you to change the A+PLUS hardware installation file called EPLD.SYS. This file is used only if you want to change the address location of the programming card or disable the color display option.

3. Press **<1>** to select option [1]. A set of prompts will guide you step by step through your system configuration.

4. Next, you are prompted step by step through the actual installation procedure.

   This installation process has a master menu that provides installation for all optional products. If you try to install an option which you have not purchased, you will be returned to the Main Menu.

   You are guided through the following process:

   a. You create a directory on your hard disk that contains all files from the installation diskette. You are also asked to confirm the installation results.

   b. You are requested to enter the address location of your programming card. Press **<Enter>**.

   c. You are asked to indicate whether you wish color or monochrome display.

   d. You are asked to enter the name of your editor. LogiCaps is the default editor. Press **<Enter>**.

5. Now, the Installation Menu is displayed. See Figure 2-2.

Installation Menu for:
Development Systems and Optional Software products

Installation Menu

[1] Install A+PLUS Software.
[2] Install LogicMap Software.
[3] Install LogiCaps & Utilities.
[4] Install TTL MacroFunctions & Altera Design Librarian.
[5] Install State Machine Converter.
[6] Install Functional Simulator.
[7] Install FutureNet interface & library.
Press a number key to select an option:
Press <Esc> to exit.

Figure 2-2. Installation Menu

a. Press <3> to install LogiCaps.

   You are prompted to insert the **LOGICAPS** and **UTILITIES** distribution diskettes.

b. When you are finished, press <Esc> to return to the Main Menu.

6. You may select other menu items from the Main Menu or press <Esc> to return to DOS.

7. **Remove any remaining diskette from the floppy disk drive and press <Ctrl><Alt><Del> to reboot the system before using any of the A+PLUS programs.**

Your CONFIG.SYS and AUTOEXEC.BAT files may have been modified to make LogiCaps run properly. Therefore, after completing the installation procedure, you may want to examine these files to determine whether they are compatible with other software on your system. The original files will have been saved as CONFIG.BAK and AUTOEXEC.BAK. To run A+PLUS, the CONFIG.SYS file *must* contain BUFFERS=12 and FILES=20.

De-Installation

If you wish to install your LogiCaps software on a different computer, you must first reverse the installation from your original system.

☞ If you are using a programming card, you must move this card *after* you have de-installed the software and *before* you re-install it on the new computer.

With the original **INSTALL** distribution diskette in drive **A** or **B**, type:

A: INSTALL <Enter> (if you use drive A)
B: INSTALL <Enter> (if you use drive B)

The Installation Main Menu is displayed (see Figure 1-1).

Select menu item [2] on the Main menu. The De-Installation Menu is displayed as shown in Figure 2-3.

```
                          A+PLUS
          USE ONLY TO DE-INSTALL VERSIONS INDICATED

                                        De-Installation Menu
[1]  De-install A+PLUS (ver 5.0) Software.
[2]  De-install LogicMap Software.
[3]  De-install LogiCaps (ver 1.6) & Utilities.
[4]  De-install TTL MacroFunctions & Altera Design Librarian.
[5]  De-install State Machine Converter.
[6]  De-install Functional Simulator (ver 2.5).
[7]  De-install FutureNet interface & library.
Press a number key to select an option:
Press <Esc> to exit.
```

Figure 2-3. De-Installation Menu

Press **<2> <Enter>**.

As a result of this de-installation, you can no longer use the LogiCaps software on the hard disk of the first system.

After the initial installation, you cannot use the original distribution diskettes for additional installations unless you run the de-installation program first.

After de-installation, some APLUS files that are not application files will be taken off the directory on your hard disk.

Software Configuration

The first time you run LogiCaps, you are asked a series of configuration questions. The file LOGICAPS.CFG is created to save your configuration. All subsequent invocations of LogiCaps use this file to determine your configuration.

The following configuration questions are asked:

Display type:
C for IBM Color Graphics Adapter
E for IBM Enhanced Graphics Adapter
H for Hercules Graphics Card

Type the appropriate letter and press **<Enter>**.

Mouse port (1=COM1, 2-COM2, default: COM1)?

Press **<Enter>** to select the default or type **2 <Enter>** to select COM2 for your mouse interface.

Programming card port (in HEX, default: 280)?

Press **<Enter>** to select the default configuration. If your port address is different, type this address in HEX and press **<Enter>**.

Mouse speed (1=fast, to 20=slow, default: 7)?

This function allows you to customize your mouse response. Initially select the default by pressing **<Enter>**.

Installation of LogiCaps is now complete and the LogiCaps editor appears on your screen.

To change the LogiCaps configuration, run the program with the following parameter:

LOGICAPS -x

You are prompted to enter your configuration specifications.

SECTION 3

Getting Started

To get started with LogiCaps, you need to know some basic features of the program. The following instructions on how to invoke LogiCaps and the descriptions of the screen display, window concept, keyboard and mouse functions, and command structure will acquaint you with these features well enough to enable you to experiment with LogiCaps. For a hands-on tutorial, consult the *LogiCaps Tutorial*, and for detailed explanations of all functions and commands, consult *LogiCaps Commands* in the ***Reference Guide***.

Invoking LogiCaps

To invoke LogiCaps, type at the DOS prompt:

LOGICAPS [filename] [@macroname] [options] <Enter>

You may use the following optional specifications at the time of invoking LogiCaps:

filename (optional) The name of a drawing file that is automatically loaded.

macroname (optional) The name of a macro recorded and written to a file with the extension .MAC. It is used in place of INIT.MAC.

options Additional configuration specifications that may be entered at the time of LogiCaps invocation. The available options are:

- **–1** Use COM1 for the mouse (default).
- **–2** Use COM2 for the mouse.
- **–l <symbol library>** Select a different symbol library. The default is ALTERA.SYM.
- **–ml** Increase the memory for LINE objects. This option allows you to reallocate memory to increase the memory capacity for lines.
- **–ms** Increase the memory for SYMBOL objects. This option allows you to reallocate memory to increase the memory capacity for symbols.
- **–mt** Increase the memory for TEXT objects. This option allows you to reallocate memory to increase the memory capacity for text entries.

–x Set new file configuration data. Lets you ignore configuration specifications in LOGICAPS.CFG and prompts you for new configuration information.

You use this option if you change your display adaptor hardware (e.g., if you change your display from CGA to Hercules).

To get a screen display of this information, type at the DOS prompt:

LOGICAPS ? <Enter>

The Screen

After you have installed LogiCaps, start the program by typing:

LOGICAPS <Enter>

The main system screen is displayed. See Figure 3-1.

☞ If your screen does not display the LogiCaps editor as shown in Figure 3-2, you may have made an error in the installation procedure. Press **Q** **<Enter>** to exit the program and then use the **-x** option to reconfigure.

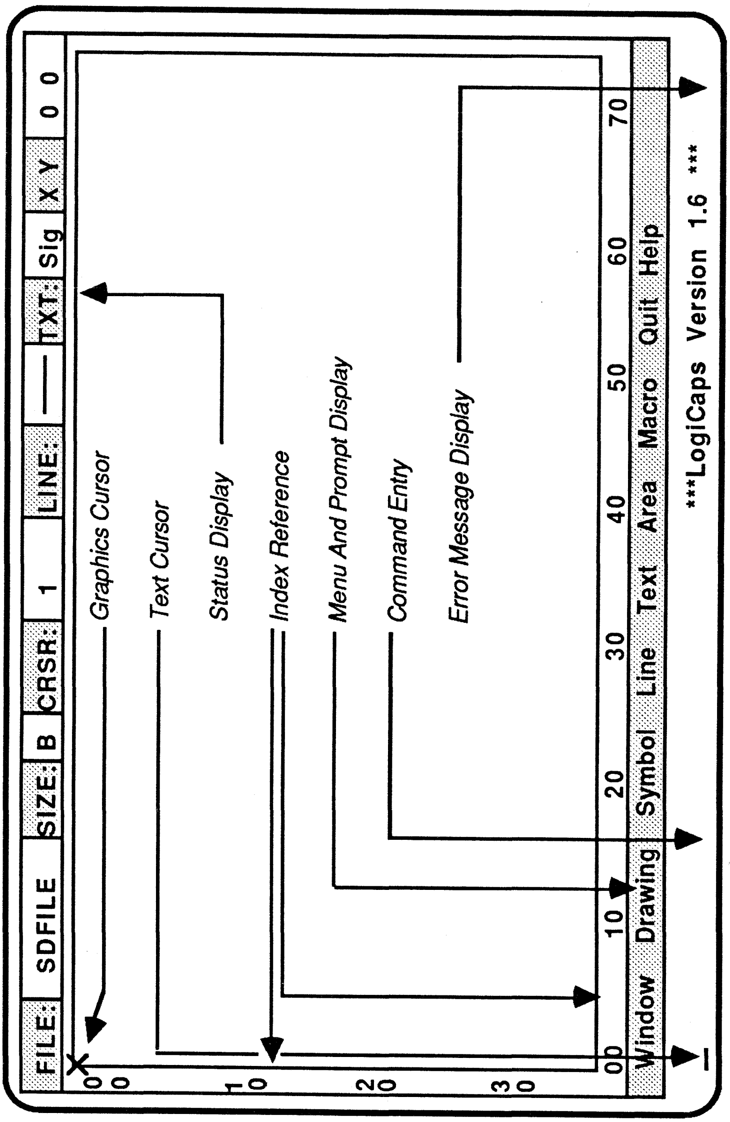


Figure 3-1. LogiCaps Screen

The screen is divided into display fields that show the current status of your design entry. The large central display is a window into your schematic drawing; the fields on the top and bottom of the screen are as follows:

Status Display Line

This line shows the current operating parameters. They are:

FILE: Shows the name of the current file.

SIZE: Shows the drawing size (A, B, C, D, E, AV, BV, CV, DV, EV, or F). Accessed with the **DS** command. ("V" indicates vertical orientation.)

CRSR: Shows the cursor key mode: "1" (default) indicates that the cursor jumps one database unit (DBU) when one of the arrow keys is pressed. (1 DBU corresponds to 0.1 inch.) You may also use the number keys to enter any size jump up to three digits. Moreover, with the **WP** (Window Pan)command the mode may be changed to PAN. Also, when you press the <+> key, the cursor goes into MKY mode and cursor movements correspond to the mouse movements. (Mouse movements do not correspond to database units when you go into reduced drawing modes.)

LINE: Shows the current type of line used for drawing. You may choose solid, dotted, high-density dotted, and dashed lines, as well as fat bus lines. You cycle through the available line types by pressing **<F3>** or executing the **LS** (Line Select) command.

TXT: Shows the current character type (small, medium, large, reverse video medium) used for text entry. You may cycle through the available character types by pressing **<F4>** or by executing the **TS** (Text Select) command.

X Y: Shows the exact location of the graphics cursor within the drawing (X Y coordinates).

Index Reference Numbers

Fixed index numbers are shown along the left and bottom sides of the screen. The numbering—indicating DBUs— starts with (00,00) in the upper left hand corner (1 DBU equals 0.1 inch). The maximum number of DBUs per drawing size are as follows:

Size A	(85,110)	Size AV	(110,85)
Size B	(170,110)	Size BV	(110,170)
Size C	(220,170)	Size CV	(170,220)
Size D	(340,220)	Size DV	(220,340)
Size E	(440,340)	Size EV	(340,440)
Size F	(900,900)		

Menu and Prompt Display Line

The menu/prompt line displays the main menu and individual submenus available during schematic entry. If you select one of these menus by entering the first letter of the menu name, a submenu is displayed. Then you select a function from the submenu by entering the first letter of the subfunction. When a command is accompanied by a prompt, the menu line is cleared of the menu and the appropriate prompt is displayed.

For example, to enter a symbol on the screen, you need to go into the Symbol menu by pressing **S**. Then the symbol submenu is displayed and you press the first letter of the function you wish to execute. Thus, if you want to enter the AND2 symbol, type on the command line:

SE AND2 <Enter>

The AND2 symbol is displayed, with the upper left hand corner of the symbol being the location of the cursor. To delete it (Symbol Delete), place the cursor inside of the symbol and type:

SD <Enter>

The available main menu entries and their general purpose are as follows:

Window: Change the display and move around the display screen.

Drawing: Load and save drawings and to output the Altera Design File (ADF).

Symbol: Work with primitive symbols and the symbol library.

Line: Enter and edit lines.

Text: Enter and edit signal names, documentation information, and general text.

Area: Define and work with an area within the drawing.

Macro: Record a keystroke sequence and store it in a file so that the entire sequence may be called up as one macro.

Quit: Exit LogiCaps.

Help: Show Help information on the Main menu and for quick reference for key functions.

☞ HELP may be called up at any time.

Command Entry Line

The field below the menu/prompt line is reserved for keyboard entry of commands.

Commands are selected by pressing the key that corresponds to the first letter of the command name.

Most commands are not executed until you press **<Enter>** after you have typed the command letters.

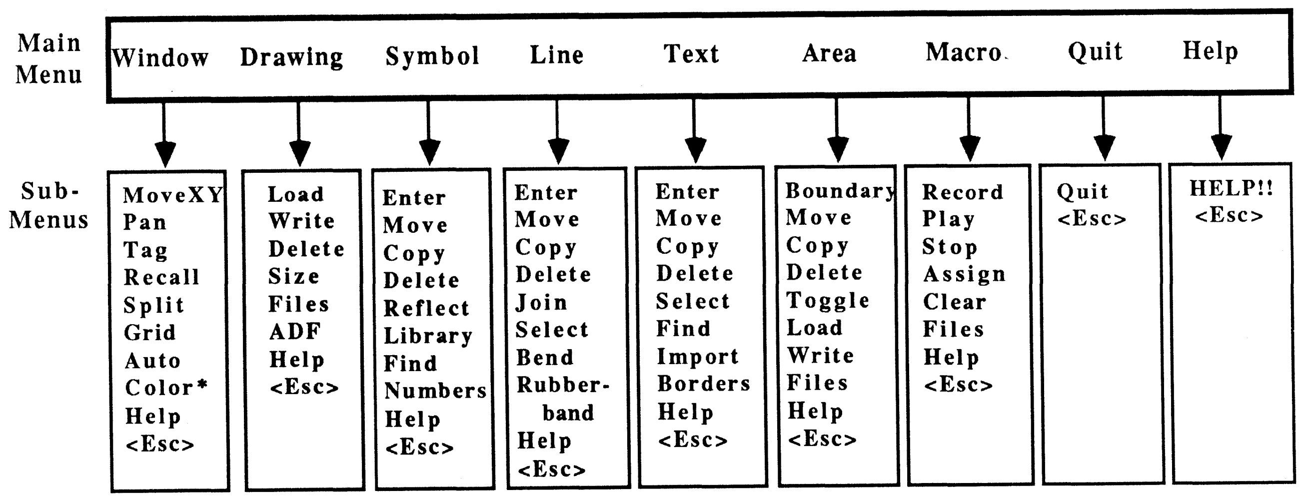
Whenever a command is displayed on the command line, you may re-execute it by pressing **<Enter>**.

<Esc> always clears the command line and returns you to Main menu.

Error Message Display Line

The field to the right of the command entry line is reserved for messages generated by LogiCaps. The section tabbed *LogiCaps Messages* contains an alphabetical list of all messages.

Figure 3-2 shows the complete LogiCaps menu structure. Refer also to *Appendix B* for a quick reference guide to all available commands.



*Not available with Hercules Graphics Card

Figure 3-2. LogiCaps Menu Structure

The Window

With the windowing facility, you can look at any portion of your drawing regardless of the display screen limitations. For example, you can zoom in on a small portion of your design so that this portion fills the entire screen. Or you can zoom out so that the entire drawing page (e.g., size A, B, C, D, etc.) is displayed with your partial or completed design showing in correct proportion and location. You can also zoom in on intermediate levels.

Moreover, with the split-screen facility, you can choose to see both—the zoomed in and zoomed out versions—displayed simultaneously, and you can enter changes to the active screen half while these changes are simultaneously reflected on the inactive screen half. (LogiCaps lets you toggle which half is active.)

Finally, with the Grid command, you can specify any desired grid spacing and adjust it to your drawing so that objects in the drawing may be lined up with the grid lines.

Zoom Levels

With LogiCaps, you can zoom in to full scale mode, zoom out to maximum reduced scale mode, or zoom to a range of intermediate zoom modes. Zoom commands—accessed with the **<PgUp>** and **<PgDn>** keys—are nested, so that you do not need to interrupt your current activity, exit from a command, and then re-enter the command. Moreover, you can jump directly from full scale to maximum reduced scale mode and back to full scale mode by pressing the **<End>** key.

The number of available intermediate zoom levels depends on the specified drawing size. For example, in a size B drawing you can zoom to full scale, maximum reduced scale, and two intermediate scales, while a size F drawing has eight intermediate zoom levels in addition to the maximum reduced scale and full scale ones.

Alphanumeric text is readable only in full scale, 3/4 scale, and 1/2 scale modes. In the two reduced scale modes alphanumeric characters will not be proportionately scaled on the screen or on the printed design. However, they will be scaled proportionately on the plotted design.

Figure 3-3 shows a portion of a schematic in full scale mode; Figure 3-4 shows the schematic in fully reduced mode. In this mode, the screen display shows the proportions of the drawing page. Shaded areas to the left and bottom of the drawing area indicate non-drawing areas.

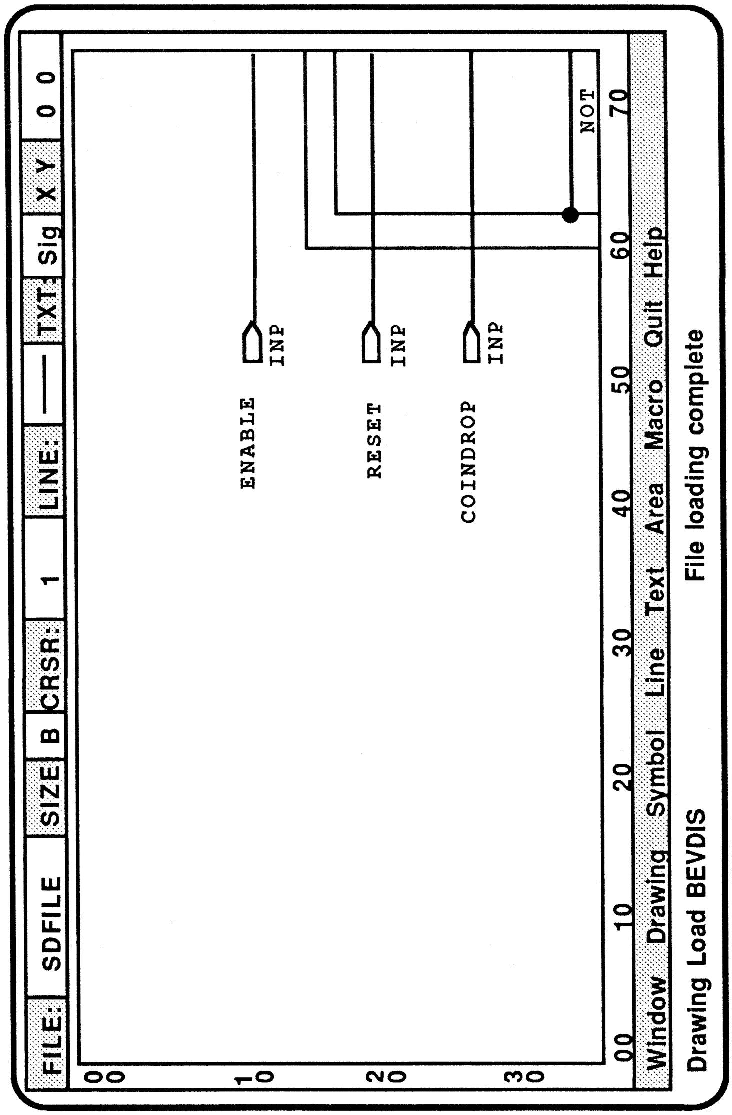


Figure 3-3. Full Scale Display Mode

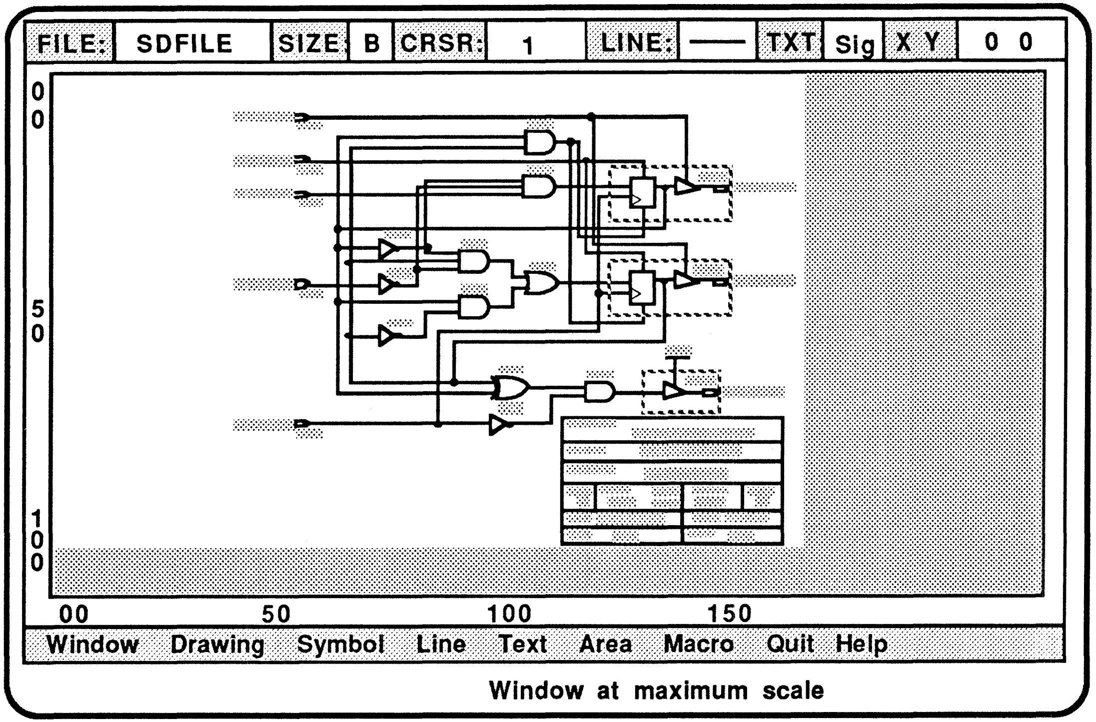


Figure 3-4. Maximum Reduction Display Mode

Split Screen

The split screen facility, accessible with a keyboard command or the **<F10>** key, lets you split the display window vertically. You may then use the "less than" key (<) to move the vertical split line to the left and the "greater than" key (>) to move the vertical split line to the right. The portion that happened to contain the cursor when the command was invoked is enclosed by a solid line box, indicating that this is the active window in which changes may be made. The other portion is enclosed by a dashed line box, indicating that it is the inactive window that will reflect all the changes that are entered into the active one. The **<Tab>** key toggles the windows from active to inactive and vice versa.

When you are in Window Split mode, you may put each window portion into a different zoom level. This facility enables you to work in a full scale drawing section, while verifying your schematic entry in the reduced mode.

Figure 3-5 shows a split window with one half zoomed out to reduced mode and one zoomed in to full scale mode.

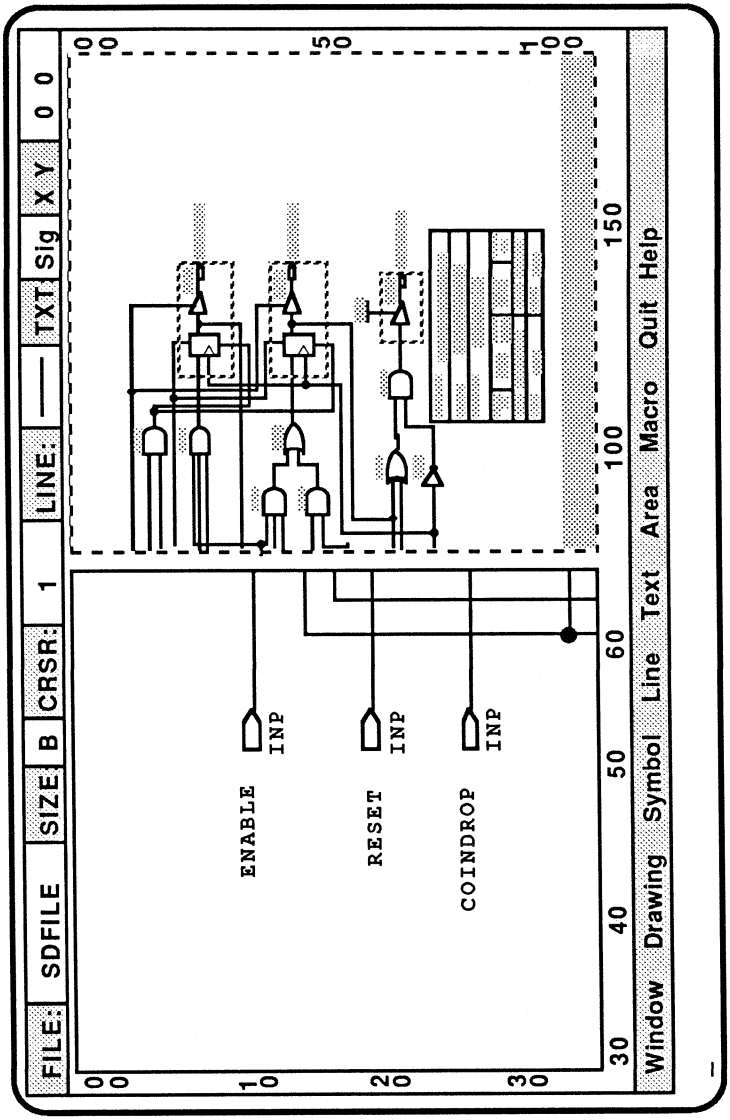


Figure 3-5. Split Window Display

Grid

When the grid is initially displayed with the **WG** (Window Grid) command or the <**F9**> key, it is set at a default setting of 10 DBUs horizontally and 10 DBUs vertically starting at coordinates (00,00).

However, you may specify any other grid settings. When you do that, the grid sets itself up so that the cursor will be located at the intersection of two grid lines. This feature allows you to line up already entered objects with the new grid by moving the cursor to a desired location rather than having to move all the currently entered objects.

While the index reference numbers are fixed and correspond to the default grid starting at (00,00), a specified grid pattern is cursor-dependent, setting the spacing and grid location at the cursor position at the time of invoking the **WG** command.

Grid lines may be spaced one or more DBUs apart. However, as you zoom out and a drawing is reduced, only the grid line intersection points are visible not the entire grid lines. (However, if the grid lines are spaced close together, even the intersection points may disappear.)

With the <**F9**> function key or the **WG TOG** (Window Grid Toggle) command, the current grid may be toggled OFF and ON.

Figure 3-6 shows a grid pattern set to user specification. Note that the grid lines intersect at the cursor location. This feature allows you to easily line up the grid with the drawing.

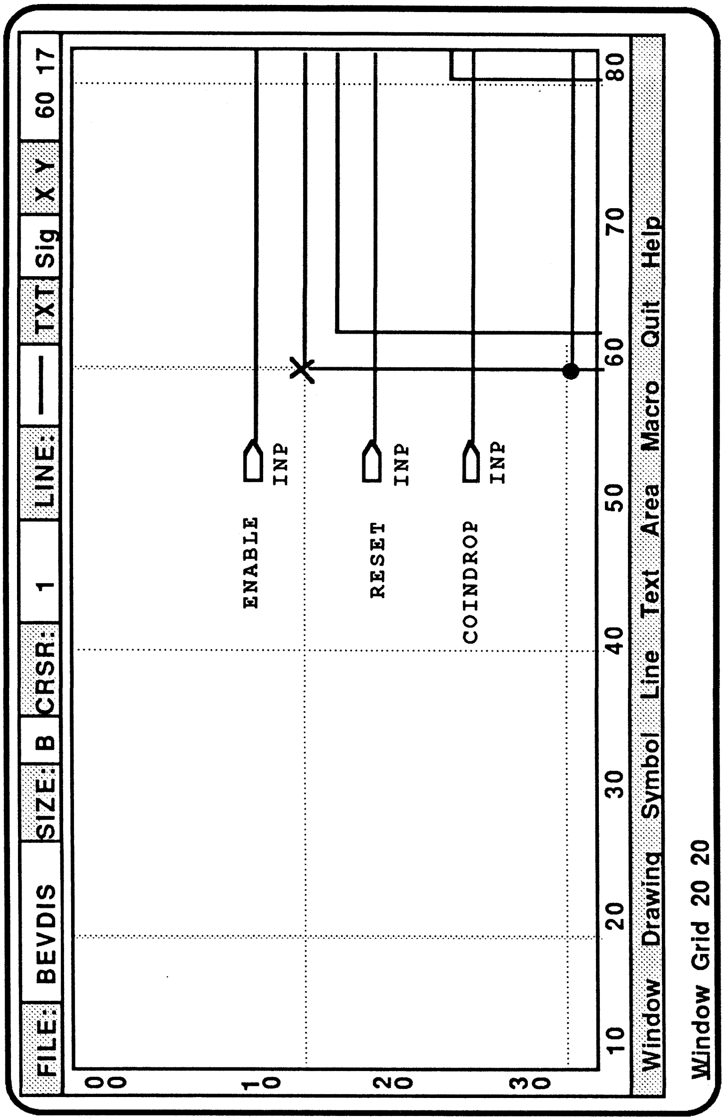


Figure 3-6. User-Specified Grid

Mouse Commands

The most common actions performed during circuit design with LogiCaps—i.e., entering, editing, deleting lines or connection dots, moving and copying symbols, rubberbanding—can be executed with the mouse exclusively.

The mouse has three buttons as shown in Figure 3-7.

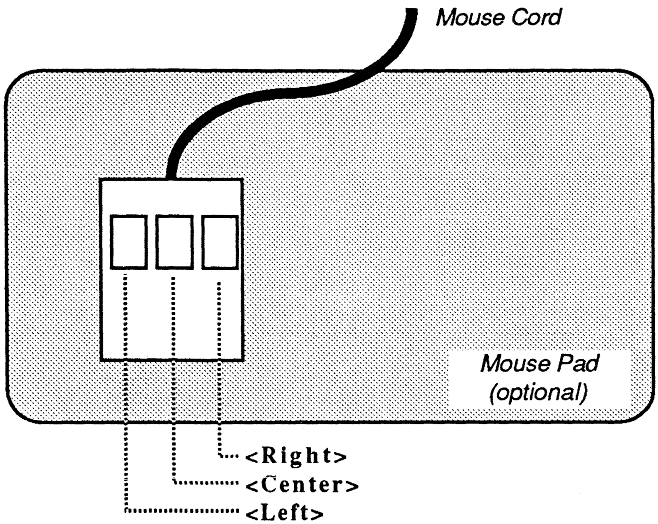


Figure 3-7. The Mouse

The mouse buttons perform these general functions:

<Left>: Draw, tag, and move lines.
Tag and move symbols, areas, and text fields.

<Center>: Enter and delete connection dots.
Tag and copy symbols, areas, and text fields.

<Right>: Cancel commands.
Delete lines.

The ***Reference Section*** contains a complete list and detailed explanations of the mouse functions.

To illustrate the mouse button functions, here is a mini-tutorial on the use of the mouse:

<Left>: MOVE and DRAW functions

1. Position the cursor somewhere inside the drawing area and type on the command line:

   SE AND2 <Enter>

   An AND2 symbol is displayed at the cursor location.

2. Place the cursor inside the AND2 gate on your screen.

3. Press **<Left>**.The boundary line of the symbol flashes, indicating that the symbol has been tagged.

4. Move the flashing dotted box to a different coordinate location. The coordinates are shown in the rightmost box of the status line.

5. Press **<Left>**. The symbol is at the new location.

6. Place the cursor on the output pin at the boundary line and press **<Left>**. The graphics cursor is replaced by the line drawing cursor. Move the cursor right and press **<Left>** again. A wire line is displayed.

7. Press **<Right>** to cancel line draw mode. The line draw cursor is replaced by the graphics cursor.

<Center>: COPY function

1. Place the cursor inside the AND2 symbol.

2. Press **<Center>**. The boundary line of the symbol flashes.

3. With the mouse move the flashing box to a new location, making sure that it doesn't overlap the original symbol boundary.

4. Press **<Center>**. The symbol is copied.
5. Press **<Right>** to cancel Symbol Copy mode.

A flashing line and a flashing border around a symbol, area or text entry indicate that that particular entry has been tagged and may now be moved or copied.

<Right>: DELETE function

1. Place the cursor on the wire line.

2. Press **<Right>**. The line is deleted.

As a safety provision, symbols cannot be deleted with a mouse command, but only with the keyboard command (**SD <Enter>**).

Keyboard Commands

All functions available in LogiCaps can be executed by the following sets of keyboard commands:

- Function key and control key commands
- Arrow key commands
- Alphanumeric command-line commands

The function keys (**<F1>** to **<F10>**) and control keys (**<Tab>**, **<Home>**, **<+>**, **<PgUp>**, **<PgDn>**, **<End>**, **<*>**, **<Space>**, and **<Esc>**) provide shortcuts to the most frequently used functions such as redrawing the screen or zooming in and out.

The arrow keys move the cursor in the respective direction either by increments of one database unit (DBU), or by specified larger increments up to three-digit numbers, or by PAN and MKY modes.

The alphanumeric keys are used for selecting a command from a menu and entering text. These command-line commands are based on a simple hierarchical structure. Since most commands are mnemonic—e.g., WM means "Window Move" and DE means "Drawing Erase"—and since the individual menus and their subfunctions are always displayed on the screen, you always have immediate access to all commands.

Any command shown on the prompt line may be repeated by simply pressing **<Enter>**.

When you are in command line editing or Text Entry mode, the **<Del>**, **<Ins>**, **<Backspace>**, **<←>**, **<→>**, **<End>**, **<Home>**, **<Ctrl><←>**, **<Ctrl><→>**, and **<Ctrl><End>** keys have the following functions:

<Del> Deletes the character at the cursor and moves the text to the right of the cursor to the left.

<Ins> Toggles between insert and exchange modes. In exchange mode it types right over the character at the cursor. In insert mode it creates a space and moves the text to the right of the cursor to the right.

<Backspace> Deletes the character to the left of the cursor and moves the text to the right of the cursor to the left.

<←> and <→> Positions the cursor without affecting the text.

<End> Moves the cursor to the end of the text.

<Home> Moves the cursor to the beginning of the text.

<Ctrl><←> Moves the cursor to the end of the previous word.

<Ctrl><→> Moves the cursor to the beginning of the next word.

<Ctrl><End> Moves the cursor right to the next pin number.

For full details on all available keyboard commands refer to Section 7 in the ***Reference Guide***.

SECTION 4

Design Guidelines

Before designing circuits for Altera parts, you should note the following format conventions for entering logic designs with LogiCaps:

- All Primitive symbols

    1. have inputs on the left of the symbol;

    2. have outputs on the right of the symbol;

    3. show the part mnemonic and the symbol.

    To facilitate design entry, LogiCaps has a Symbol Reflect function that lets you flip the symbol horizontally so that inputs are on the right side and outputs are on the left. This feature, however, does not affect design processing by the Altera Design Processor (ADP).

- All I/O Primitive symbols

  1. have P and Oe input lines on the top of the symbol, C input at the bottom of the symbol, and all other input lines and clock input lines to the left of the symbol;

  2. have combinatorial and registered output lines on the right (these are the PIN_NAME designations representing a pin on the Altera part to be programmed; Input Primitives have PIN_NAME on the left);

  3. have combinatorial and registered feedback lines on the bottom of the symbol.

- Clear (C), Preset (P), and Output Enable (Oe) inputs to primitives are optional and do not require connections. If left unconnected, C and P will default to GND, and Oe will default to VCC.

- Embedded blanks are not allowed in a pin name.

- When connecting a wire to a pin stub, be sure the line is drawn right up to the stub and no gap is left.

- T-type connections of lines require a connection dot.

- Lines and text fields that are not connected to symbols or other lines may be freely used; they will not show up in the ADF.

- Lines connected to VCC and GND symbols are assigned the VCC and GND signal name. Any other names assigned to such lines are ignored.

- Text fields within symbols must not be edited, moved, copied, or deleted unless the entire symbol is moved, copied, or deleted. They contain information essential to generating the Altera Design File (ADF).

- To enter a Boolean Equation into a schematic with the EQN1 or EQN8 primitive, you must type the desired equation over the field "% Arbitrary Boolean Equation; %". Be sure to use the syntax required for equations according to the EQUATIONS: section in the ADF File Format, as described in *Altera Design File Format* in the ***A+PLUS Reference Guide***.

Unused text fields in the EQN8 primitive will appear in the ADF generated from a design. However, they will be treated as comments and will not have any impact on the ADF.

An equation may exceed the width of the box and still be interpreted correctly.

Example of an equation entered into an EQN1:

Node A = (Node B + Node C) * /NodeD ;

where nodes A, B, C, and D are internal nodes used elsewhere in the design.

- Pin names may be up to eight characters long and may contain any character except '%', ',', '=', '@', '(', and ')'.
- A pin number is assigned to a pin name or buried register output by entering an at-symbol (@) and the specific number after the name. Name, @, and number together may not exceed 11 characters.
- PIN_NAME text fields within I/O and Input primitives should be edited by placing the cursor on the word and executing the **TE** (Text Enter) command. You must edit the existing field, not replace it with a new field, to generate a valid ADF.
- Text fields in the TITLE symbols must be edited, not replaced, so that the ADF can be generated correctly.
- With the **TE** (Text Enter) command, signals in different areas or pages of a schematic may be wired together without being physically connected. Simply use the **TE** command to give the same name to all signals that should be tied together.
- Table 4-1 shows a list of all legal pin name and node name characters for the LogiCaps program.

1. When generating the JEDEC file, the ADP will convert an asterisk (*) in an input or output pin name to a tilde (~). Ensure that this conversion does not create duplicate pin names (e.g., if you were to use both the * and ~ symbols in pin names).

2. The Functional Simulator (FSIM) uses periods (.) and pin name extensions to reference internal nodes of I/O primitives. Therefore, if you use periods in pin name assignments, you may get unpredictable results when running FSIM.

Table 4-1. Legal Pin Name and Node Name Characters in LogiCaps

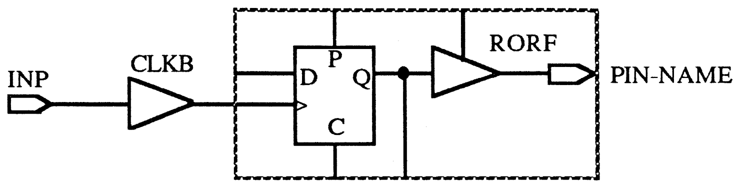
Legal Pin Name Characters			
!	:	S	j
"	;	T	k
#	<	U	l
$	>	V	m
&	?	W	n
'	A	X	o
*	B	Y	p
+	C	Z	q
-	D	[	r
.	E	\	s
/	F	]	t
0	G	^	u
1	H	_	v
2	I	`	w
3	J	a	x
4	K	b	y
5	L	c	z
6	M	d	{
7	N	e	|
8	O	f	}
9	P	g	~
	Q	h	
	R	i	

Legal Node Name Characters		
A	X	u
B	Y	w
C	Z	x
D	a	y
E	b	z
F	c	0
G	d	1
H	e	2
I	f	3
J	g	4
K	h	5
L	i	6
M	j	7
N	k	8
O	l	9
P	m	
Q	n	
R	o	
S	p	
T	q	
U	r	
V	s	
W	t	

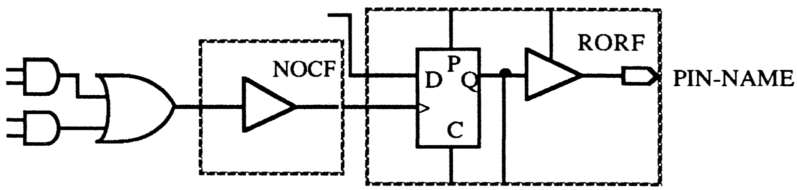
- Clk inputs to primitives can be obtained in three ways (legal clocking configurations for all Altera EPLDs are described in *Appendix A* of the ***A+PLUS Reference Guide***:

  (1) A clock driven by an input pin is assumed to be synchronous unless all dedicated clock pins have already been used. (The EP600, EP900, and EPB1400 [BUSTER] parts have two clock pins; the EP1800 has four.)

  (2) To force a clock input to be asynchronous, feed the signal to a CLKB (Asynchronous Clock Buffer) primitive and connect the CLKB output to the input of a flipflop. Example:



  (3) A clock driven by Boolean logic is always asynchronous. (Use of the CLKB primitive is optional.) BUSTER (EPB1400) macrocells allow Boolean equations feeding the Clock inputs to I/O primitives to contain up to two product terms. In the EP600, EP900, and EP1800 parts, an asynchronous clock with logic that requires two product terms as input can be implemented by connecting the logic as an input to an NOCF primitive, and using the output of the NOCF as the clock input to the register. Example:



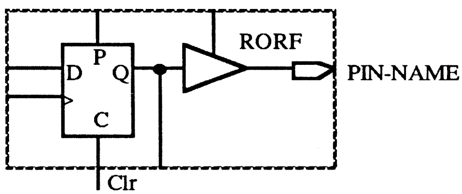
- Clock (Clk), Read Strobe (Rs) and Write Strobe (Ws) inputs to Bus I/O primitives (BUSX, LBUSI, LBUSO, LINP8, RBUSI and RINP8) are optional. If they are connected, they must be connected directly to a pin. If they are left unconnected, you must retain the commas that delimit the fields. When the Clk, Rs, and Ws are left unconnected, the logic connected to the Output Latch Enable (Ole), Output Enable (Oe), and Write Enable (We), respectively, will have full control of the flipflop. (For additional information, refer to *Altera Primitive Library* in the ***A+PLUS Reference Guide*** and the ***Altera Data Book***.)

- Read Strobe (Rs) and Write Strobe (Ws) input pins for BUSTER (EPB1400), as well as Rs and Ws inputs to Bus I/O primitives are predefined as Active Low signals.

- BUSTER (EPB1400) macrocells allow Boolean equations feeding Clock (Clk) and Output Enable (Oe) inputs to I/O primitives to contain up to two product terms; Clear (C) inputs are limited to one product term. The logic feeding the Ouput Latch Enable (Ole), Output Enable (Oe), Read Enable (Re), and Write Enable (We) inputs to Bus I/O primitives (BUSX, LBUSI, LBUSO, LINP8, RBUSI and RINP8) may contain up to two product terms.

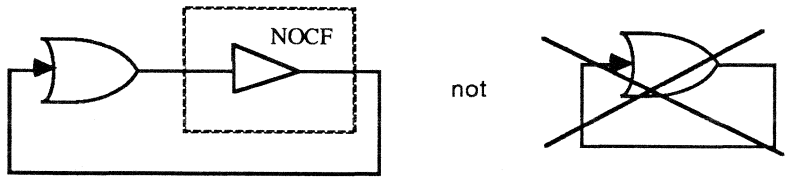
  The architecture of other Altera parts allows Boolean equations feeding Clear, Clock, Preset, Latch Enable, and Output Enable inputs to I/O primitives (i.e., C, Clk, P, Le, and Oe) to contain only one product term. See *Appendix A* in the ***A+PLUS Reference Guide*** for specific information.

  Example:



Clr = Oa1 * Fbk2' * Oa3;

- Boolean equations cannot feed back to themselves without first going through an I/O primitive or buried register. This means that the left-hand side of an equation can never appear in the right-hand side or in an intermediate term of the right-hand side of the same equation. For example, cross-coupled latches cannot be constructed within a macrocell using only Boolean expressions. To obtain combinatorial feedback, you must use an I/O primitive (e.g., COCF, NOCF, ROCF, COIF, ROIF, TOIF). Example:



- Automatic part selection will not be successful if you specify pin assignments, or if there are too many inputs, outputs, or macrocells.

- Combinatorial feedback from a macrocell to itself may cause unpredictable results.

- The Altera Design Processor does not support the product-term sharing feature of the EP1210. This feature is accessed through LogicMap II.

- Dual I/O feedback is supported by the global macrocells of the EP1800 and all generic BUSTER macrocells. To implement dual I/O feedback, use the MacroFunctions CO2F, JO2F, RO2F, SO2F, and TO2F.

- The ADP helps you to locate an error in the schematic by giving the name of an output primitive (i.e., Altera-defined output symbol) in the schematic. If you had not named the output node, the ADP will name the node by (1) giving the **X,Y** position in the schematic or by (2) giving the symbol number of a TTL MacroFunction in the design.

  (1) The node name will be of the form **.pxxxyyy**, where **xxx** is the **X**-coordinate and **yyy** is the **Y**-coordinate. The **p** indicates multiple sheets of schematics, so that the first file listed after the **File name** prompt of the ADP will indicate **p=0**, the second will indicate **p=1**, etc. For example, the name **.1037008** indicates the node at the location **X=37,Y=8** in the second file submitted to the ADP. To locate the error in your LogiCaps schematic, you simply type **WM 37,8**, and the cursor will move to the specified location.

  (2) If the error is located in a node within a MacroFunction, the node name will be of the form **.pMmmmNn**, where **mmm** is the symbol number and **n** is an internal node name. The **p** indicates multiple sheets of schematics. For example, the node name

.0M004N1 indicates a node within MacroFunction 4 (displayed with the **SF** [Symbol Find] command) in the first schematic submitted to the ADP.

Although you don't have direct access to a node within a MacroFunction, you may view the internal logic by referring to the description of the Standard MacroFunctions in the *Reference Section* or to the ***TTL MacroFunctions*** manual.

SECTION 5

LogiCaps Tutorial

LogiCaps provides a large number of functions. You will probably use the common ones, such as entering symbols and drawing lines, most of the time; others you may use occasionally. All available functions are described in detail in the *Reference Guide*.

To quickly acquaint you with the functions you are most likely to use, the sessions are organized as follows:

Session 1: Symbols and Lines

1. Invoking LogiCaps
2. Using the main HELP function
3. Entering a symbol
4. Moving a symbol
5. Copying a symbol
6. Drawing a straight and an orthogonal line
7. Entering a connection dot
8. Saving the design and exiting to DOS

Session 2: Areas and Text

1. Loading the drawing file
2. Zooming
3. Defining an area boundary
4. Moving and toggling an area
5. Saving an area
6. Locating a specified symbol and entering a pin name and number
7. Saving a previously saved design

Session 3: Rubberbanding and Window Display Commands

1. Rubberbanding
2. Moving around the drawing
3. Setting the window color
4. Splitting the window display

Session 4: Grid, Macros, and Title Block

1. Specifying the grid size
2. Recording a macro
3. Entering the title block
4. Stopping and playing back a macro

Session 5: ADFs

Generating the Altera Design File (ADF)

Session 6: Plotting and Printing the Design

1. Plotting the circuit design
2. Printing the circuit design

Session 7: Entering the ADF into A+PLUS

1. Submitting the ADF to the ADP
2. Programming the EPLD

Session 8: Entering Bus Lines

1. Using Bus Lines in BUSTER Designs
2. Using Bus Lines for Reference in EPLD Designs

Sample Session 1

In this session you learn how to :

1. Invoke LogiCaps
2. Use the HELP function
3. Enter a symbol
4. Move a symbol
5. Copy a symbol
6. Draw a line
7. Enter a connection dot
8. Save the design and exit to DOS

1. Invoking LogiCaps

At the DOS prompt, type:

LOGICAPS <Enter>

After the LogiCaps copyright information has been displayed, the LogiCaps screen is opened. The flashing graphics cursor (X) is located in the upper left hand corner of the screen. The flashing text cursor (_) is positioned under the first letter of the command line prompt.

There are two types of cursors:
(1) the graphics cursor that looks like a flashing X;
(2) the text cursor that looks like a flashing underscore.

2. Using the HELP Function

On the command line, type:

H <Enter>

The main Help screen is displayed with information about the functions assigned to the function keys.

Press **<Space>** to move through the entire Help file.

Press any other key to return to the normal drawing screen.

To get help information regarding any of the other main menu entries, type the first letter of the menu entry plus **H** for Help. For example, to get on-line information about the Window command, type:

WH <Enter>

The screen display changes to information about the **W** (Window) command. If you want to know specifics about the Area command, you type **AH <Enter>**, etc.

Press any key to return to the drawing screen.

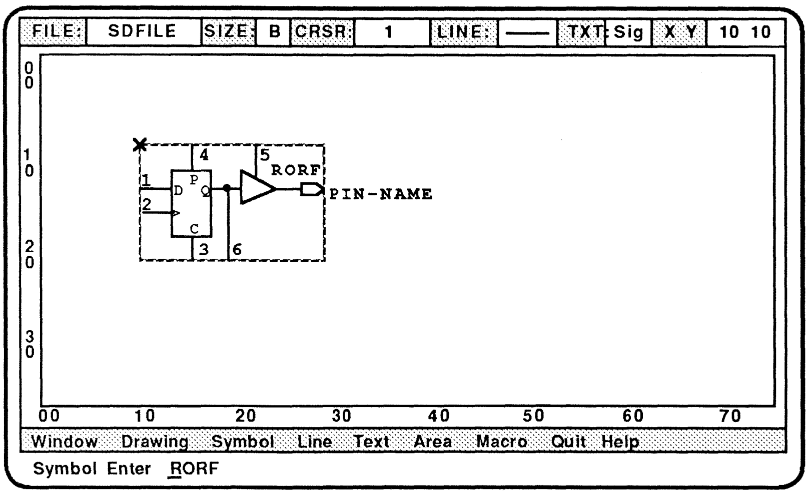
3. Entering a Symbol

The last field on the status line (X Y) shows the coordinate numbers for the current cursor location. Move the graphics cursor until the coordinates displayed on the status line show (10,10). Then type the Symbol Enter command:

SE RORF <Enter>

The RORF primitive is displayed with the cursor located in the upper left hand corner of the symbol at (10,10).

Type **SE ?** if you wish to see the complete list of available symbols.



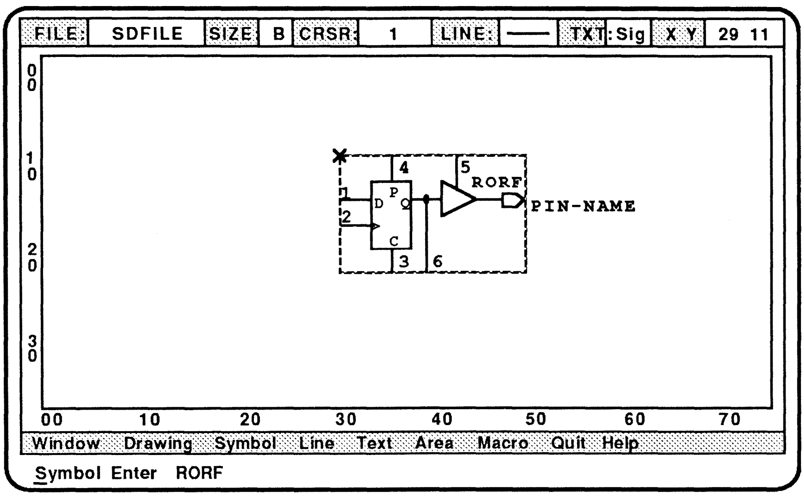
4. Moving a Symbol

Place the cursor inside the RORF symbol and press **<Left>**. The symbol boundaries flash indicating that Symbol Move mode is active.

As you move the mouse, a flashing box cursor the size of the RORF symbol boundary is displayed. Move this box until the status line indicates coordinates (29,11).

☞ The coordinates displayed in the status line reflect the position of the the upper left corner of the movable box cursor.

Press **<Left>** to execute the move of the symbol to the new location and exit Symbol Move mode.



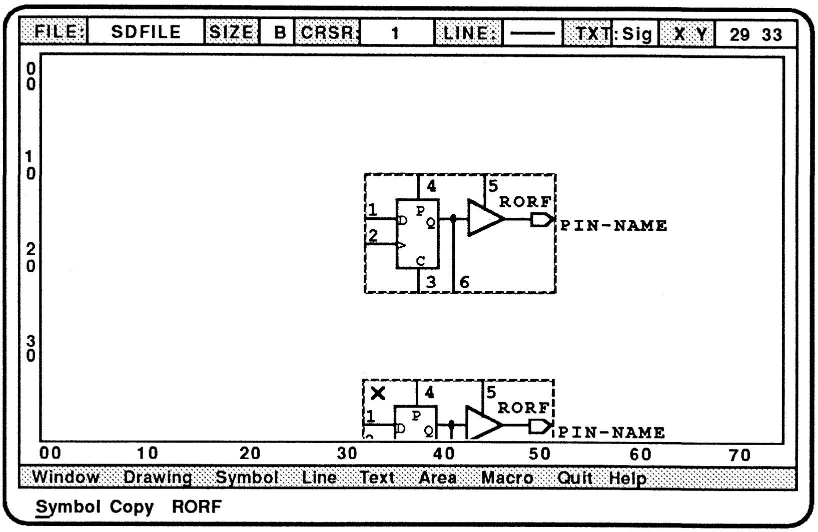
5. Copying a Symbol

Place the cursor inside the RORF symbol and press **<Center>**. The symbol boundaries flash indicating that Symbol Copy mode is active.

As you move the mouse, a flashing box cursor the size of the RORF symbol is dragged along. Move this box to (29,33).

Press **<Center>** to copy the RORF at this new location.

Then press **<Right>** to exit Symbol Copy mode. The graphics cursor is restored.



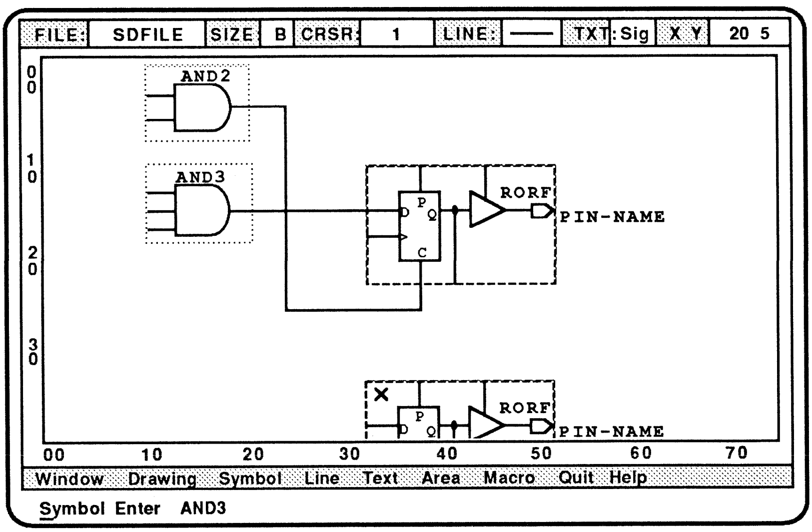
6. Drawing a Straight and an Orthogonal Line

With the **SE** command, enter an AND2 symbol at location (8,1) and an AND3 at (8,11). Place the cursor on the output pin stub of the AND2 at the point where it intersects with the boundary symbol. Press **<Left>** to go into Line Draw mode, indicated by a large flashing line cursor.

Move to (23,5). (In this case, you may wish to use the **<→>** key instead of the mouse, and just move right by 6 data base units.) Press **<Left>**. Move to (23,25), press **<Left>** again; then move to the point where the Clear pin stub meets the boundary line of the symbol (35,23). Notice that the line automatically makes a 90-degree bend without you having to press **<Left>** again to start a new segment of the line. Note also that the lines are all dotted, i.e., the Line Draw mode is active but the command has not yet been executed.

If you press **<Center>** now, the line will bend in the opposite direction. You can toggle the **<Center>** key so that the line first goes horizontal then vertical or first vertical and then horizontal. Refer also to the **LB** (Line Bend) command in the ***Reference Guide***.

Press **<Left>** three times to draw the line segments and exit Line Draw mode. Move to (17,15). Press **<Left>**. Move to (29,15). Press **<Left>** twice to draw the line segment. See the illustration below:



7. Entering a Connection Dot

Before you can enter the connection dot with the **LJ** (Line Join) command, you should complete the following moves:

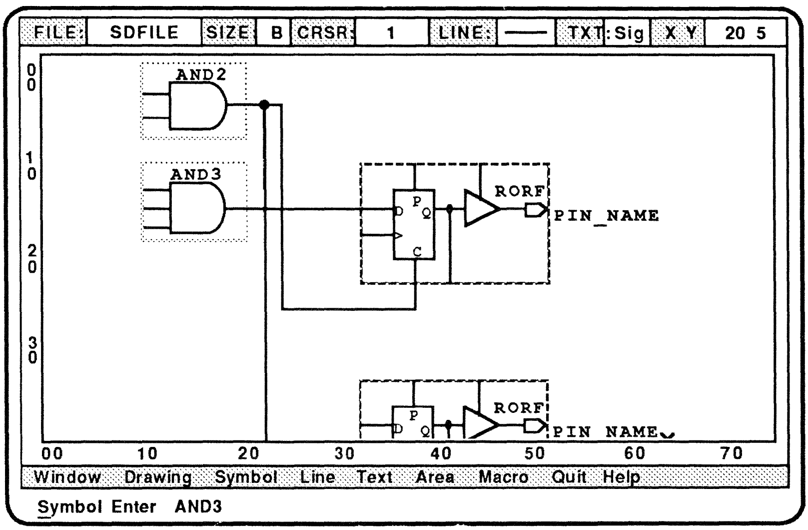
Move the cursor over the bottom window boundary. The window automatically pans to a new view, displaying the second RORF symbol.

Autopan is enabled when you invoke LogiCaps. To disable this function, type **WA** (Window Autopan) and press **<Enter>**. Refer to **WA** in the ***Reference Guide***.

Place the cursor on the location where the Clear pin stub (C) meets the symbol boundary (35,45). Press **<Left>** and move to (20,47). Press **<Center>** to toggle the **LB** function, if necessary. Press **<Left>**, then autopan to (20,5). (If **WA** is disabled, type **WA <Enter>** to toggle it.) Press **<Left>** three times to exit Line Draw mode.

To execute the **LJ** command, press **<Center>** to enter a connection dot. (Or you may type **LJ** on the command line and then press **<Enter>**.) Your partial drawing should look like the one shown below.

To delete the dot, press **<Center>** again.



8. Saving the Design and Exiting to DOS

To save this partial drawing and write it to a file type:

DW BEVDIS <Enter>

The drawing is saved under the filename BEVDIS.SD. The extension is added for you automatically.

To exit to DOS type:

Q <Enter>

If you are working with a file that you have edited since saving or loading it, you are prompted with:

Save changes Y, N, or <Esc>?

Type **Y** (Yes) if you want to save the changes that you entered since the last time you loaded or saved the file. If you don't want to save the changes, press **N** (No). If you want to abort the Quit command, press **<Esc>**.

Sample Session 2

In this session you learn how to :

1. Load a drawing file
2. Zoom out and in
3. Define an area boundary
4. Move and toggle an area
5. Save an area
6. Locate a specified symbol and enter a pin name and pin number
7. Save a previously saved design

1. Loading a Drawing File

To load an existing drawing file, you may (1) load the file when you invoke LogiCaps, or you may (2) first invoke LogiCaps and then load the file. In both cases, you don't need to type the extension .SD.

(1) Type at the DOS prompt:

LOGICAPS BEVDIS <Enter>

LogiCaps is invoked and the BEVDIS file is loaded at the same time.

(2) Alternatively, you may first invoke LogiCaps and then load the file by typing:

DL BEVDIS <Enter>

The partial drawing is displayed on the screen.

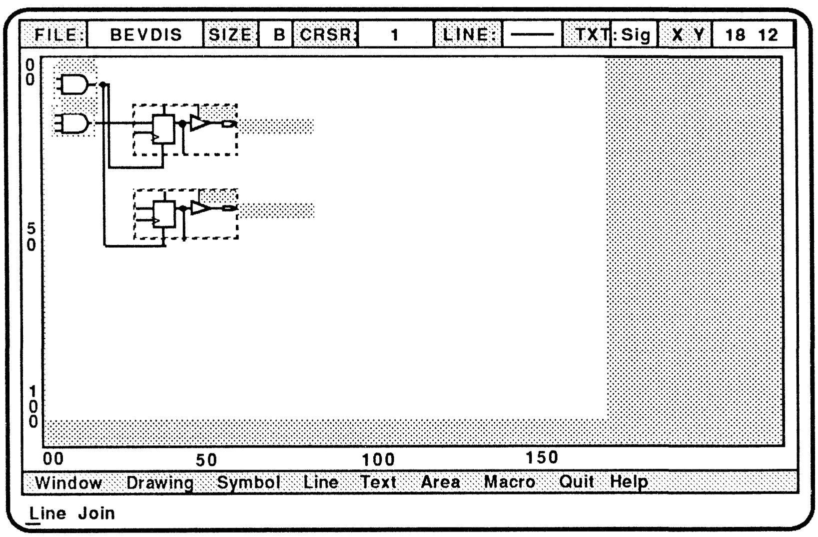
☞ The **DF** command displays a list of all drawing files with the extension .SD.

2. Zooming Out and In

To enter additional objects into the drawing, you must reposition the current window on the drawing sheet. Before you can do that, you must zoom out to reduced scale in order to display the entire drawing on the screen.

Press **<PgUp>** three times. Each time you press **<PgUp>**, the drawing scale is further reduced. The third time, the screen shows a black area—containing the reduced drawing—approximately representing the proportions of a size B drawing. (Size B is the default). The area not covered by the drawing is shaded.

The number of intermediate zoom levels varies with the specified drawing size. For example, a size B drawing allows four zoom levels: (1) full scale, (2) 2 intermediate levels, and (3) maximum reduced scale. A size F drawing allows 10 zoom levels. You reach these levels with the **<PgUp>** and **<PgDn>** keys.



3. Defining an Area Boundary

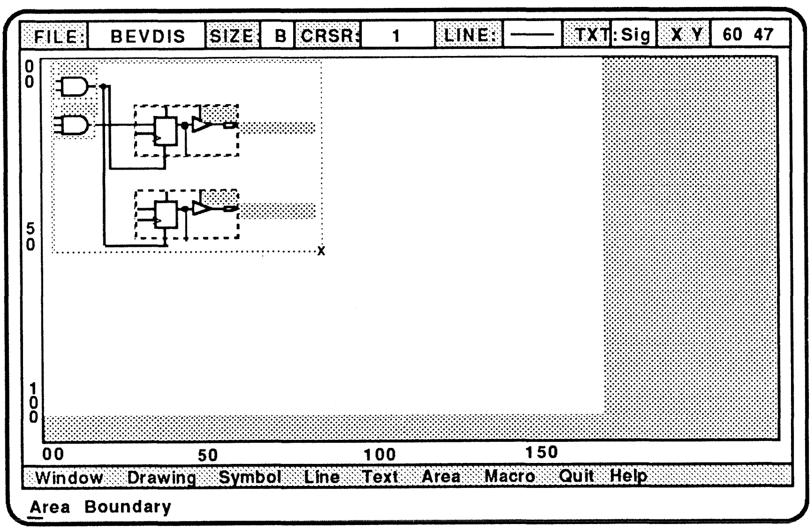
With the drawing zoomed out to maximum reduced drawing mode, use the cursor keys to position the cursor at (6,1) and press **<F6>** or type on the command line:

AB <Enter>

A flashing cross cursor is displayed. As soon as you move the mouse, the cross turns into a box cursor. Move this box cursor until the box contains the entire partial drawing. (The coordinates should not be smaller than (64,48)).

Press **<Left>** to define the area and exit Area Boundary mode. The graphics cursor is now displayed at the lower right hand corner of the area.

You can adjust the area boundary by placing the graphics cursor on an edge or corner of the boundary and then invoking the **AB** command again.



4. Moving and Toggling an Area

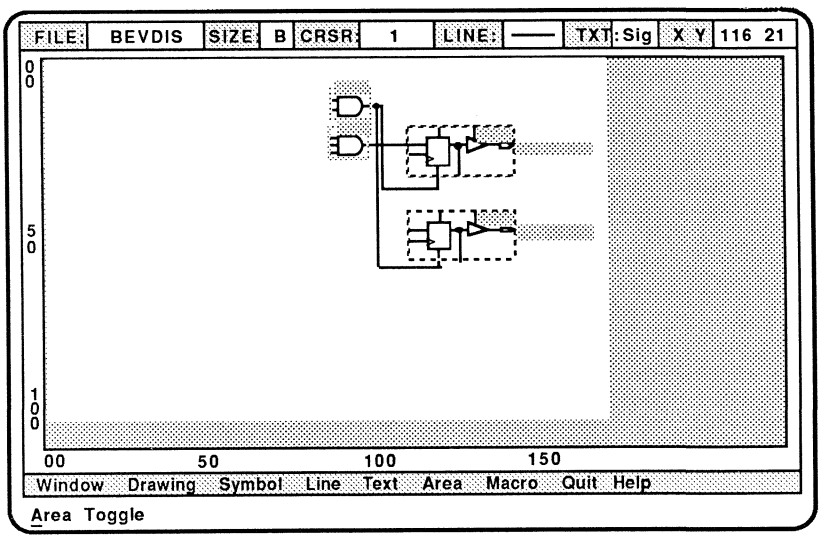
While you are still in maximum reduced drawing mode, place the cursor at (29,5). Press **<Left>** twice to enter Area Move mode, indicated by a flashing area boundary.

Move the boundary box cursor until the status line displays coordinates (129,18) and press **<Left>**. The area is moved to the new location.

Now enter the Area Toggle command to remove the area boundary. Press **<F5>** or on the command line type:

AT <Enter>

The area boundary is removed from the display and the area is considered "undefined." (See the illustration.) Subsequent execution of the **AT** command will redisplay the area boundary at its previous position. (At this time, the area should be redefined.) Note that only one area may be defined at any given time.



5. Saving an Area

First, use the **AT** command to redefine the area which you "undefined" in the previous step.

To save the defined area with the Area Write command type:

AW AREA1 <Enter>

This command saves the defined area with the filename AREA1.SDA.

Whenever you save a file with the **AW** command, you can load it again with the **AL** (Area Load) command. It will be displayed at the current cursor position without interfering with the screen display, provided there is no symbol overlap.

If you have saved a drawing with the **DW** (Drawing Write) command, you could load it as an area with the **AL** command so that the entire drawing will be displayed at the current cursor position without erasing the current screen display. To do this, change the filename from <filename>.SD to <filename>.SDA.

Now remove the area boundary by typing:

AT <Enter>

6. Locating a Specified Symbol and Entering a Pin Name and Pin Number

Press <**PgDn**> three times to get to full scale mode. While the screen is at full scale, execute the Symbol Numbers command by typing:

SN <Enter>

Just inside the upper left hand corner of each symbol boundary a number is displayed indicating the sequence in which you entered the symbol. (The **SN** command toggles between Display Enabled and Display Disabled).

With the **SF** (Symbol Find) command locate the desired RORF symbol by typing:

SF 1 <Enter>

The graphics cursor is displayed in the upper left corner of the desired RORF symbol.

Disable symbol numbers by typing:

SN <Enter

Place the graphics cursor on the PIN-NAME text field of that RORF and type the Text Enter command:

TE

The graphics cursor is replaced by the text cursor located under the first letter of the text field (P). Type:

DROPCUP@12

The name DROPCUP is assigned to pin 12.

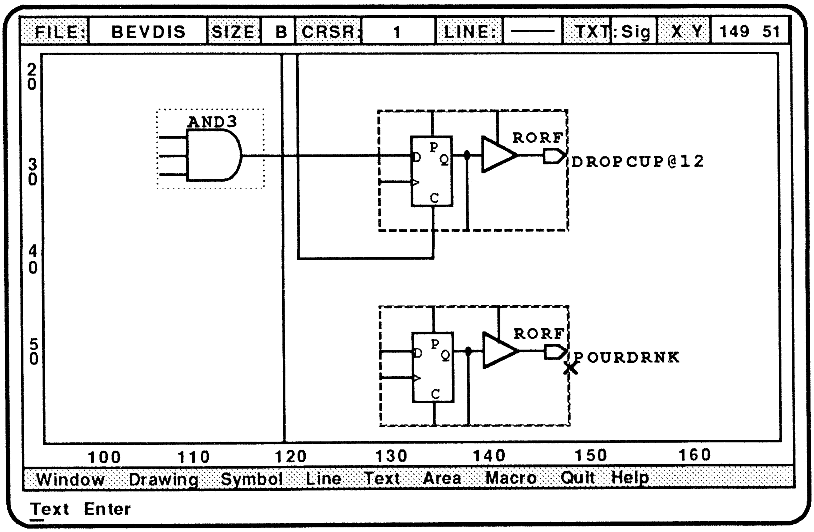
Press <**Esc**> to exit Text mode, move to the PIN-NAME text field in the other RORF and press <**Enter**> again to go back into Text mode.

Type:

POURDRNK

Note that you do not need to re-enter the **TE** command the second time. As long as it is displayed on the command line, you press **<Esc>** to exit Text Enter mode and **<Enter>** to return to Text Enter mode.

The illustration shows the partial drawing entered so far.



7. Saving a Previously Saved Design

Use the Drawing Write command by typing:

DW <Enter> Y

Note that the filename BEVDIS is displayed as default because it is the name of the file most recently saved with the DW command. If you want to save the file under a different name, simply type over the displayed name.

If the filename already exists, you are prompted with:

OK to overwrite existing file? Y[N]

Type **Y** if you want to overwrite the existing file, **N** if you don't.

Press **Q <Enter>** to exit.

Sample Session 3

In this session you learn how to:

1. Use the rubberbanding feature
2. Move around the drawing
3. Set the window color
4. Split the window display

1. Rubberbanding

Load your design by typing from DOS:

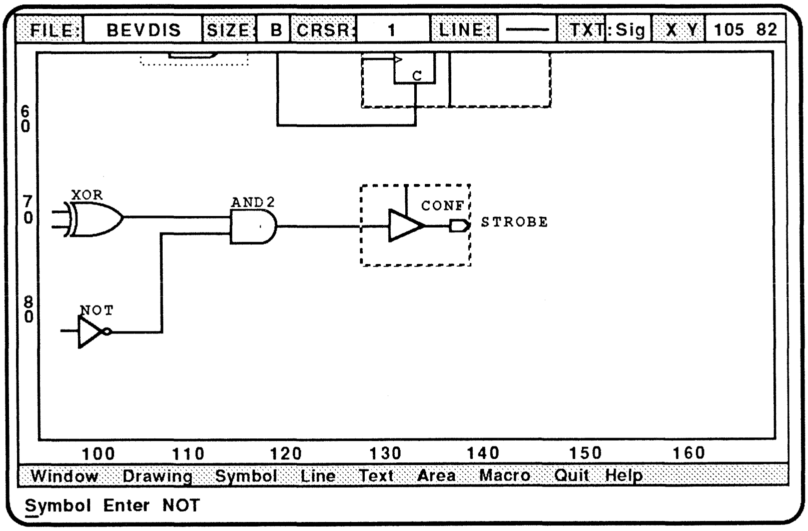
LOGICAPS BEVDIS <Enter>

The rubberbanding feature enables you to move objects in the drawing without disconnecting the lines. Before using this feature, complete the following steps:

At (134,71) enter a CONF symbol. At (115,71) enter an AND2 symbol. At (96,70) enter an XOR symbol. At (97, 84) enter a NOT symbol.

To center the window on the symbols just entered, move to (128,75) and press **<Home>**. Draw a line between the AND2 output and the CONF input. Then draw a line between the XOR output and the top input pin stub (1) of the AND2 symbol. (Verify the pin stub number by typing **SN <Enter>**.)

Place the cursor at (115,76), where input 2 pin stub meets the symbol boundary. Press **<Left>** and move to (110,76). Press **<Left>** again; then move to (105,87) where the NOT output pin stub meets the symbol boundary. If necessary, press **<Center>** to toggle the Line Bend function. Press **<Left>** three times to enter the line. See the illustration below.



Execute the Line Rubberband command by typing:

LR <Enter>

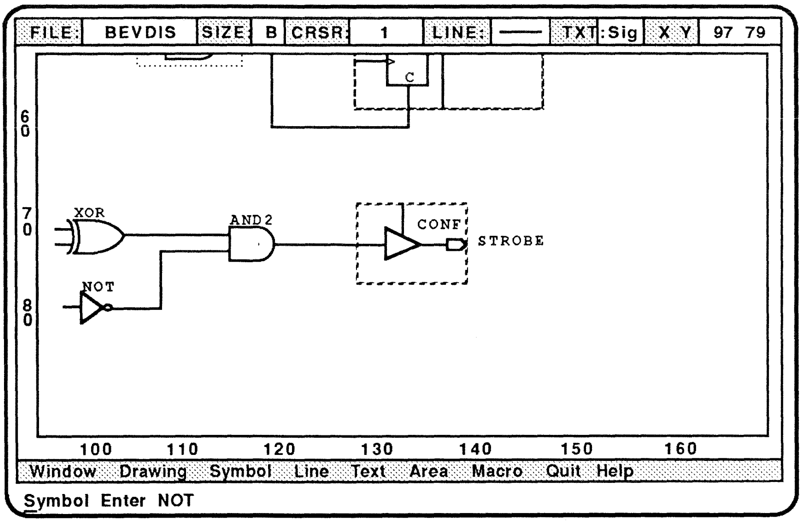
The message line indicates whether the rubberbanding feature is enabled or disabled. Toggle **<Enter>** until the message reads:

Rubber-banding Enabled

Place the cursor inside the NOT symbol and press **<Left>**. The boundary line flashes indicating Symbol Move mode.

Then move to (97,79). The symbol boundary as well as the line connected to it are moved to the new location. Press **<Left>** to exit Symbol Move mode and complete rubberbanding.

The partial view should now look like the one shown in the illustration below:



2. Moving Around the Drawing

There are several ways to jump between two views of the window: **<Tab>** (Leap), **WR** (Window Recall), **WT** (Window Tag), **WP** (Window Pan), **WM** (Window Move), mouse, and cursor keys. Three of these are demonstrated here. For full details on all of these commands refer to the ***Reference Guide***.

<Tab> (Leap)

In full scale mode, place the cursor at (124,27) or anywhere approximately in that area. Press **<Tab>**. Then move so that the cursor is approximately at (124,75) and press **<Tab>** again.

Everytime you press **<Tab>** now, the cursor will jump between these two views on the screen.

☞ When you press **<Tab>**, you save the current cursor position. Subsequently pressing **<Tab>** allows the cursor to jump between the saved location and the current location.

WT (Window Tag)

With the **WT** command you save the zoom level as well as the cursor and window location on the drawing.

Go to (124,27), zoom out to the second intermediate zoom level (press **<PgUp>** twice), and enter:

WT <F1>

Then go to (124,75), press **<PgDn>** twice, and enter:

WT <F2>

The **<F1>** key will now restore the cursor to the first specified location in reduced drawing mode and **<F2>** will restore it to the second specified location in maximum reduced drawing mode.

Press **<F2>**.

Move the cursor to (37,9), execute the **SE** (Symbol Enter) command, and enter an INP symbol.

Place the cursor inside the INP symbol, press **<Center>**. Move to (37,21), then press **<Center>** again.

Make additional copies of the INP symbol at (37,28) and (37,50) and (37,80); then press **<Right>.**

Enter the remaining symbols, lines, and connection dots. Type:

WM

You are prompted to enter the X and Y coordinates and press **<Enter>**.

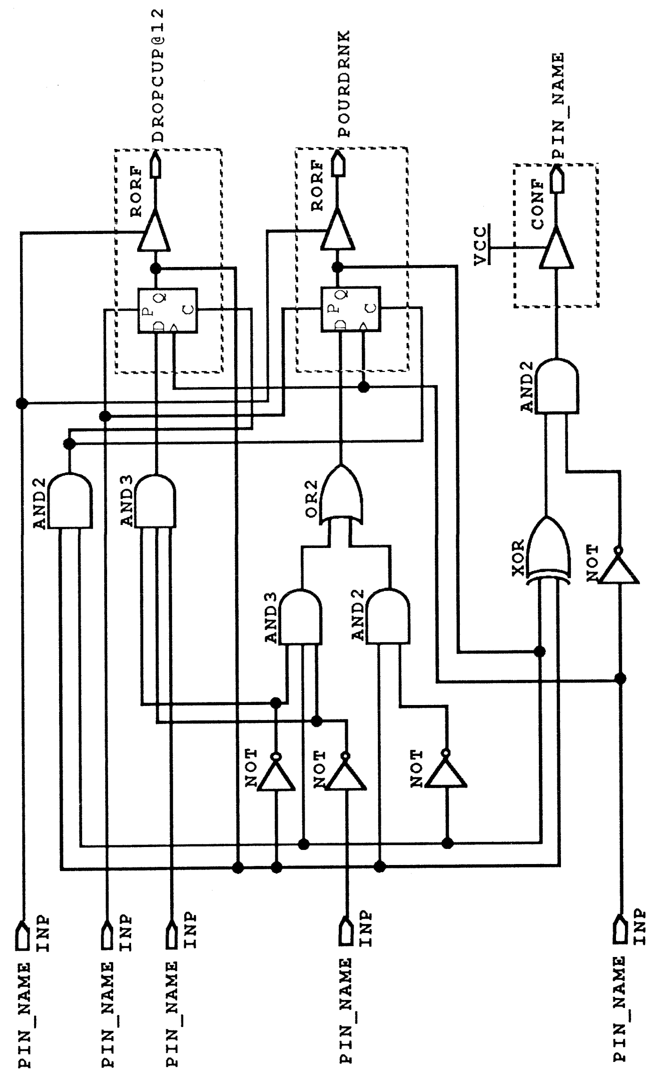
Enter the following symbols at the specified coordinate locations:

NOT	(72,38)
NOT	(72,49)
NOT	(72,59)
AND3	(91,42)
AND2	(91,53)
OR2	(108,46)
VCC	(137,68)

To verify your design, the following list gives the locations of the remaining connection dots:

(60,37)	(87,41)
(60,41)	(84,48)
(60,56)	(93,73)
(63,46)	(89,82)
(63,62)	(128,52)
(127,11)	(125,23)

The following illustration shows the circuit as it should appear now.


DROPCUP@12
POURDRNK
PIN_NAME
RORF
RORF
CONF
VCC
AND2
AND3
OR2
AND3
AND2
AND2
XOR
NOT
NOT
NOT
NOT
PIN_NAME
INP

3. Setting the Window Color

If you have a CGA or an EGA board, type:

WC

to go into Window Color mode.

In CGA mode, you then press **<Enter>** until the screen displays the color of your choice.

In EGA mode, you are prompted for:

Window Color Palette: <n>

where n is the present color palette number. Type **0** and press **<Enter>**. 0 is the palette number for the background color. Press **<Space>** until you have the desired color or type a number between 0 and 63, and then press **<Enter>**.

Invoke **WC** again, and then choose palette number 5 and press **<Enter>**. This number selects color for symbols, symbol names, text, solid and dashed lines, bus lines, and connection dots. (Refer to the **WC** command in the ***Reference Guide*** for available palette numbers.)

In both modes, you can save the new color configuration. Simply enter the **WC** command, and when you are prompted for input, type:

S <Enter>

The **-x** option used when invoking LogiCaps overrides this new configuration and restores the default configuration (see **Invoking LogiCaps**).

4. Splitting the Window Display

LogiCaps enables you to vertically split the screen display into two sections so that you see two independent viewing windows, each with its own index reference.

Start in full scale mode with the cursor located close to one of the INP symbols. Press **<PgUp>** twice to go to the second intermediate zoom-out level. You should be able to see nearly the entire drawing. On the command line type:

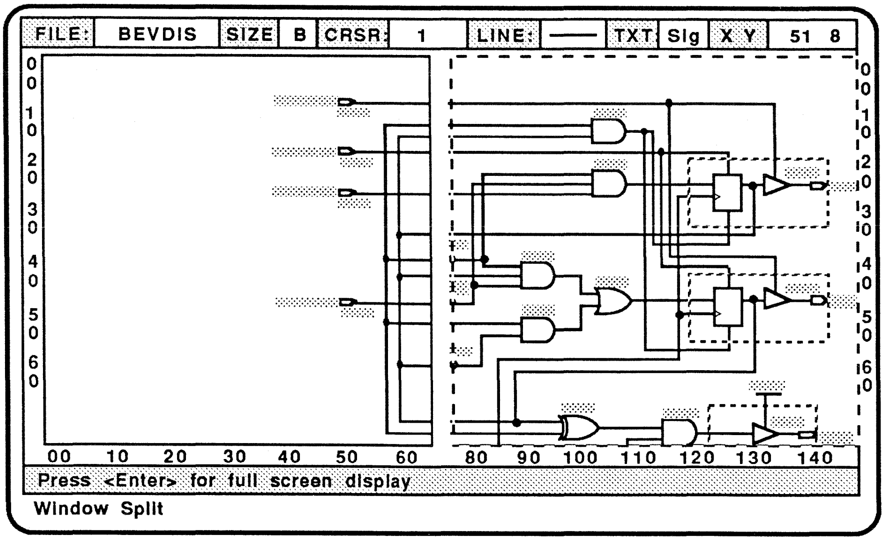
WS <Enter>

The **<F10>** key toggles the Window Split function ON and OFF. You may also move the vertical split line to the left by pressing the "less than" key (<) and to the right by pressing the "greater than" key (>).

The window is split. The left window has a solid boundary indicating that this is the "active" window in which you may enter changes. (Note: If the cursor had been on the right side of the screen when you entered the **WS** command, the right window would be active.) The right window has a dashed boundary indicating that it is inactive, i.e., the changes made in the left window are all reflected in the right window, but you cannot initiate changes in the right window until you activate it. See the illustration below.

Toggle **<Tab>** to reverse the window status and activate the right window. Press **<Tab>** again to activate the left window. Then autopan to the right: the left window display changes, the right one does not. However, the cursor location is reflected in both windows.

Press **<PgUp>**. The drawing on the left is reduced, the one on the right is not; however, the cursor location is reflected again in both windows. Refer to the following illustration.



Now press **<PgDn>** until the window is in full scale mode and pan to the upper left side of the drawing.

Press **<Tab>** to activate the right window and press **<PgUp>**, **<PgDn>**, and **<Home>** until you can see all the INP symbols (third intermediate zoom level).

Press **<Tab>**. Place the cursor inside the PIN_NAME text field of the topmost INP symbol (you can verify the correct symbol by looking at the right window cursor location).

Type:

TE
ENABLE <Del> <Del> <Esc>

(**<Del>** deletes the leftover characters.) Move to the next INP symbol and press **<Enter>** (you don't need to re-enter the **TE** command because it is still displayed on the command line).

Type:

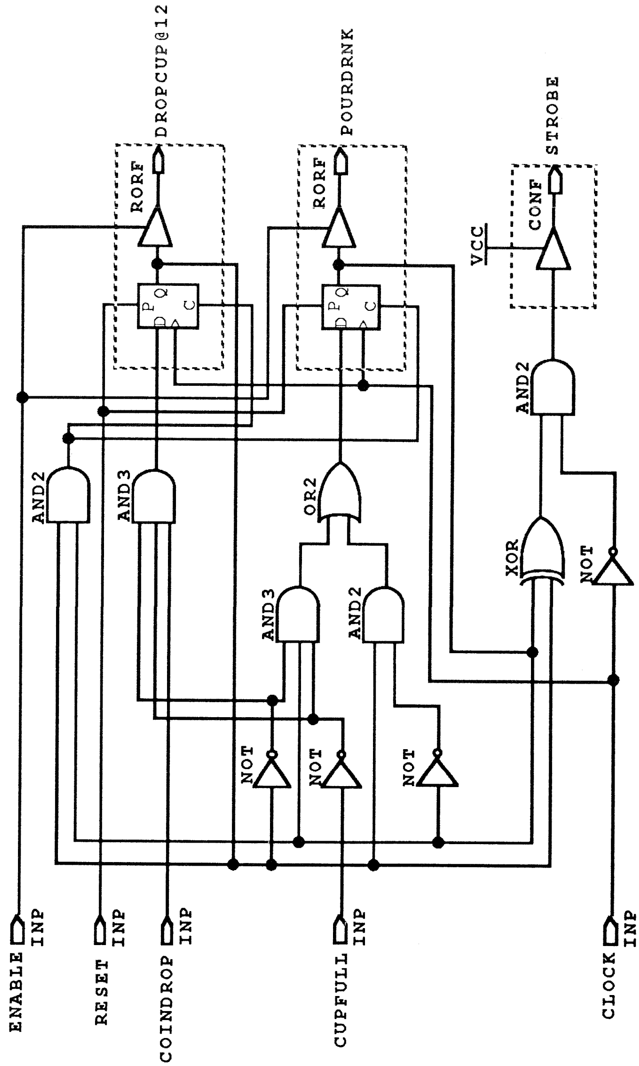
RESET <Del> <Del> <Del> <Esc>

Repeat this process for the remaining INP symbols and the CONF. The following illustration shows the current drawing including the pin names to be entered.

Type **WS <Enter>** again to exit Window Split mode.

Save the design.

The schematic entered so far is shown in the illustration below.


DROPCUP@12
RORF
POURDRNK
RORF
STROBE
CONF
VCC
D P Q C
D P Q C
AND2
AND3
OR2
AND3
AND2
AND2
XOR
NOT
NOT
NOT
NOT
ENABLE INP
RESET INP
COINDROP INP
CUPFULL INP
CLOCK INP

Sample Session 4

In this session you learn how to :

1. Specify the grid size
2. Record a macro
3. Enter the title block
4. Stop and play back a macro
5. Assign a macro to an <F> key

Note:

The macro function enables you to record, save, and play back a sequence of entered commands as one single command. The sequence may include just a few instructions or it may be long enough to incorporate an entire drawing page layout.

Macros are assigned macro filenames that are listed in a macro directory. One macro filename, INIT.MAC, is reserved for system setup. This macro will be executed automatically each time LogiCaps is executed.

General recommendations when recording a macro:

- Use the menu commands rather than the function key commands.

- Be sure to include all the necessary steps in the macro. For example, details such as character type, line type, zoom scale, and cursor position are all included when a macro is recorded. Therefore, set all the different conditions you need after you have entered the Macro Record command, so that they are included in the recording session. When you play back the macro, be sure these conditions are set again.

1. Specifying the Grid Size

Load the file by typing:

LOGICAPS BEVDIS <Enter>

Type:

WM 21,18 <Enter>

When you specify your own grid spacing, the cursor location will always be one of the intersections in the grid lines regardless of where the cursor is located. For this particular sample session, the first grid line intersection must be at (21,18).

To execute the Window Grid command type:

WG

The command line indicates that the default grid is set to (10,10).

Type over the default grid size:

21 18 <Enter>

The specified grid is displayed with the cursor being at (21,18), which is one of the intersections of the vertical and horizontal lines. (Refer to the next step for an explanation regarding this particular grid spacing.)

With the **<F9>** key you can toggle the grid ON or OFF. Toggle it ON.

2. Recording a Macro

In this sample session, you will record the drawing index and title block position of a size B drawing. This macro can then be used for the Beverage Dispenser design and for any other size B drawings.

To execute the Macro Record command and name the macro file type:

MR INDEX <Enter>

Note that the FILE: text field in the status line has changed to RECORDING MACRO.

Be sure the status line displays the correct settings for SIZE, LINE, and TXT:

(1) To set SIZE to B, type **DS B <Enter>**.

(2) To set LINE to a solid single line, type **LS <Enter>**; then toggle **<Enter>** until the solid line is displayed.

(3) To set TXT to small letter size, type **TS <Enter>**; then toggle **<Enter>** until the small letter size is displayed. The message line will display **Small Text**.

Now divide the index reference into 8 sections on the horizontal and 6 sections on the vertical sides:

The vertical lines are divided into 110 DBUs, so the index reference lines must be spaced 18 DBUs apart

The horizontal side of a size B drawing is divided into 170 data base units (DBUs); therefore, the index reference lines must be spaced 21 DBUs apart.

With the **WM** (Window Move) command place the cursor at (0,0). Go into full scale mode by pressing **<PgDn>**.

While in main menu, type 18 to set the cursor spacing to 18 DBUs (verify the setting in the CRSR: field in the status line). Then press **<↓>**. The cursor jumps 18 DBUs.

Press **<Left>**, move the mouse two DBUs to the right (2,18). Press **<Left>** twice to draw the line.

Place the cursor on the line, and type **LC <Enter>** to go into Line Copy mode. Then press **<↓>** and **<Center>** to copy a line 18 DBUs apart. Press **<↓>** and **<Center>** until you reach the bottom of the drawing page (6 dividing line stubs).

Repeat this process on the opposite vertical side.

On the horizontal sides, repeat this process but set the spacing to 21 by typing **21** when in the Main menu. There will be 8 dividing line stubs.

Finally, place the cursor at (2,2), press **<Left>** and draw a box around the drawing.

Now, enter the alphanumeric index designators.

Position the cursor at (10,2) and type:

TE 1 <Esc>

With CRSR: still set to 21, press to jump into the next index field.

Press **<→>** then **<Enter> 2** then **<Esc>**.
Press **<→>** then **<Enter> 3** then **<Esc>**.
Etc.

Repeat this process until you have entered numbers 1 to 8 on the top index reference.

Duplicate these numbers in the bottom index fields. Start at (10,110).

To enter the vertical index designators, set CRSR: to 18, then place the cursor at (0,10), and type

TE A <Esc>.

Press **<↓>** then **<Enter> B** then **<Esc>**.
Press **<↓>** then **<Enter> C** then **<Esc>**.
Etc.

Repeat this process on the opposite side of the drawing page. Start at (168,10).

Without stopping the **MR** command, go to the next step in this tutorial session.

4. Entering the Title Block

Place the cursor at (119,88).

Type:

SE TITLE <Enter>

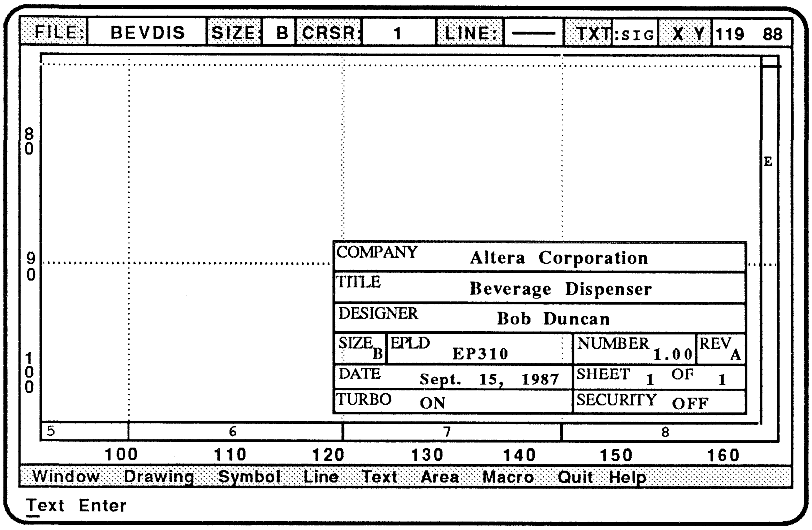
Now use the **TE** (Text Enter) command to change all the fields in the title block to the information shown in the following illustration. If you are not certain as to where to position the cursor to enter the text, execute the **TB** (Text Borders) command and toggle the **<Enter>** key to the message:

Text borders display enabled

Then place the cursor at the bottom left corner of the text field and type **TE**. The graphics cursor is replaced by the text cursor at the first text location in the text field. (See also Sample Session 2).

<table>
<tr><td colspan="4">COMPANY
Altera Corporation</td></tr>
<tr><td colspan="4">TITLE
Beverage Dispenser</td></tr>
<tr><td colspan="4">DESIGNER
Bob Duncan</td></tr>
<tr><td>SIZE
B</td><td>EPLD
EP310</td><td>NUMBER
1.00</td><td>REV
A</td></tr>
<tr><td colspan="2">DATE
Sept. 15, 1987</td><td colspan="2">SHEET **1** OF **1**</td></tr>
<tr><td colspan="2">TURBO
ON</td><td colspan="2">SECURITY
OFF</td></tr>
</table>

The screen should look like this:



Go to the next step while still in RECORDING MACRO mode.

4. Stopping and Playing Back A Macro

Once you have finished drawing the index reference and entering the title block information, execute the Macro Stop command by typing:

MS <Enter>

The macro is saved with the filename INDEX.MAC.

Type **DD <Enter>** to delete the drawing. Then type **DL <Enter>** to load the BEVDIS file again. It will not show the index reference, because you did not save the drawing with the index.

Now type:

MP <Enter>

The name INDEX appears as default.

LogiCaps now plays back the macro in the same way in which you entered the keystrokes. If you wish to interrupt the playback, press **<Esc>**.

Write the drawing to disk.

The completed schematic is shown in Figure 5-1.

(Refer also to Figure 5-3 for a plot of the schematic including the macro.)

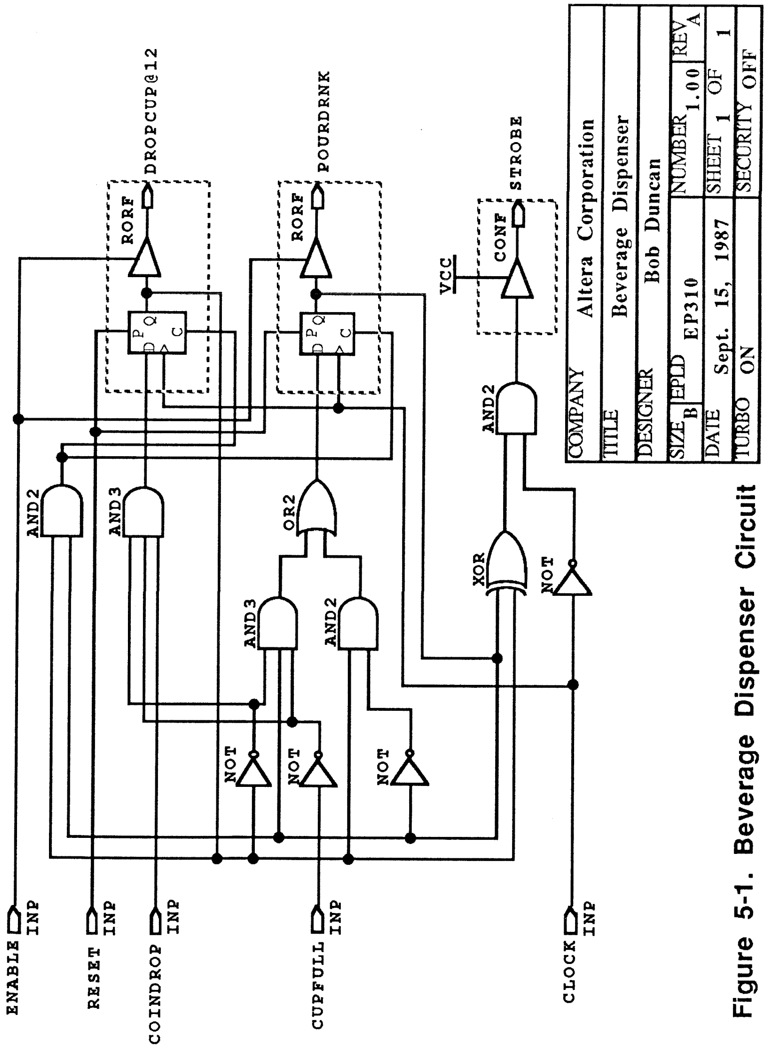


Figure 5-1. Beverage Dispenser Circuit

5. Assigning a Macro to an <F> Key

If you wish to assign a function key to the macro so that you can play it back, use the Macro Assign command by typing:

MA INDEX <Enter>

You are prompted with:

Press <F> key to be assigned

Press **<F1>**. The macro is assigned to the **<F1>** key and can always be played back by pressing **<F1>**. Note that the macro assignment now supersedes the function originally assigned to **<F1>**. (The **MC** (Macro Clear) command returns the function key to its originally assigned function.)

Sample Session 5

In this session you learn how to :

Generate the ADF

Note:

To enable the Altera Design Processor (ADP) to process the design BEVDIS, the design must be converted into an Altera Design File (ADF). LogiCaps automatically converts the schematic into an ADF.

Generating the Altera Design File (ADF)

Load the schematic with the Drawing Load command:

DL BEVDIS <Enter>

Then type:

DA <Enter>

The message line indicates that LogiCaps is tracing the wires in the drawing and is propagating the node names. Then the ADF is diplayed and simultaneously written to disk under the name BEVDIS.ADF.

Once the ADF is generated, you are asked to **Press any key to continue**. At this time, you may wish to exit LogiCaps and print out the ADF from DOS.

The ADF is now ready for processing by A+PLUS.

☞

1. The header is generated from the title block in the drawing.

2. The ADF includes the symbol reference numbers as comments enclosed in % symbols next to the associated symbol entry.

3. Unlabeled nodes are assigned names in the format ..XXXYYY, where XXX and YYY are locations within a symbol near the node in the drawing.

4. The ADF is terminated with END$.

Figure 5-2 shows the ADF for BEVDIS.SD.

```
BOB DUNCAN
ALTERA CORPORATION
Sept. 15, 1987
1.00
B
EP310
BEVERAGE DISPENSER

OPTIONS: TURBO = ON   SECURITY = OFF
PART: EP310
INPUTS:
   ENABLE, RESET, COINDROP, CUPFULL, CLOCK
OUTPUTS:
   DROPCUP, POURDRNK, STROBE

NETWORK:
DROPCUP, ..139036    =   RORF(..129028,..129030,..135036,
                         ..135026,..142026) % SYM 1 %
POURDRNK, ..139058   =   RORF(..129050,..129030,..135036,..135026,
                         ..142026)  %SYM 2 %
..135036   = AND(..139036,..139058,..) %SYM 3 %
..129028   = AND(..108026,..108028,..108030) % SYM 4%
STROBE     = CONF(..134075,VCC) % SYM 5 %
..134075   = AND(..115074,..115076) % SYM 6 %
..115074   = XOR(..139058,..139036) % SYM 7 %
..115076   = NOT(..129030) % SYM 8 %
..142026   = INP(ENABLE) % SYM 9 %
..135026   = INP(RESET) % SYM 10 %
..108030   = INP(COINDROP) % SYM 11 %
..056052   = INP(CUPFULL) % SYM 12 %
..129030   = INP(CLOCK) % SYM 13 %
..108026   = NOT(..139036) % SYM 14 %
..108028   = NOT(..056052) % SYM 15 %
..078062   = NOT(..139058) % SYM 16 %
..098046   = AND(..108026,..139058,..108028) % SYM 17 %
..098057   = AND(..139036,..078062) % SYM 18 %
..129050   = OR(..098046,..098057) % SYM 19 %

END$
```

Figure 5-2. ADF for BEVDIS.SD

Sample Session 6

In this session you learn how to:

1. Plot the design
2. Print the design

Note:

Refer to *Appendix C* for details on the supported plotters and printers.

1. Plotting the Circuit Design

LogiCaps supports the HP7440/GE, 7475, 7550, 7570, 7580, 7585 drafting plotters.

From DOS type:

lcplot BEVDIS <Enter>

BEVDIS.SD is plotted to serial port 1, with all options turned OFF. Figure 5-3 shows the plotted design.

The following options are available, with defaults given in parentheses. (If you enter **lcplot <Enter>**, a list of available options is displayed.)

-b	Plot symbol number and border (OFF).
-m	Maximize drawing to sheet (OFF).
-o filename	Plot to file, filename (OFF).
-p[12]	Plot to port 1 or 2 (port 1).
-q[0..4]	Plot in the specified quadrant. Numbering of quadrants follows the Cartesian coordinate system. (0 = center of drawing sheet).
-s	Plot sheet border (OFF).
-#	Plot symbol pin numbers (OFF). Pin numbers assigned to pin names with the @ symbol will not be plotted.
-fa[1..4]	Pen acceleration (system default).
-ff[1..8]	Pen force (system default).
-fp[1..8]	Pen number (pen 1).
-fv[1..60]	Pen velocity (system default).

For example, if you enter at the LOGICAPS prompt,

lcplot -p2 -q3 -# BEVDIS.SD

the BEVDIS schematic is plotted to serial port 2 in quadrant 3 of the plotting paper with symbol pin numbers included.

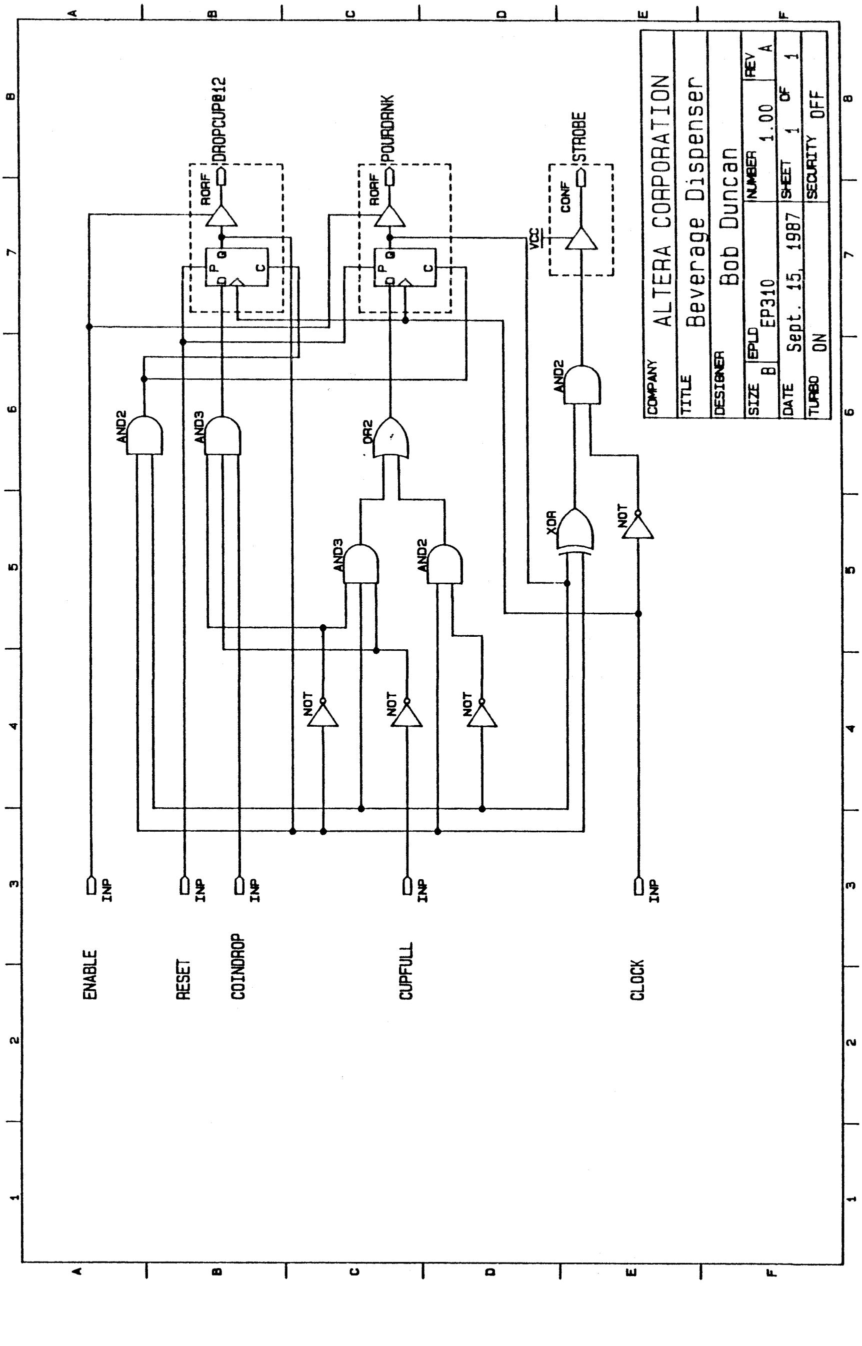


Figure 5-3. Plot of BEVDIS

2. Printing the Circuit Design

LogiCaps supports the Epson FX/MX series and compatible printers.

From DOS type:

lcprint BEVDIS.SD <Enter>

BEVDIS.SD is printed with all options turned OFF.

The following options are available, with defaults given in parentheses. (If you enter **lcprint <Enter>**, a list of available options is displayed.)

-b	Print symbol numbers and borders (OFF).
-e	Expand the scale to approximately 1:1.
-f	Fast print (OFF). This command prints the design at twice the speed but with about half the resolution.
-g	Print grid lines (OFF).
-m	Maximize printing area (OFF). This command positions the printed design close to the upper left corner of the sheet to reduce white space around the design.
-o name	Print to file or device (LPT1:).
-r	Reduce scale (OFF). This command reduces the size of the printed design by 50 %. A second **-r** command in the options string would reduce the design by 75 %.
-s	Print sheet border (OFF).
-w	Adjusts the setting to accommodate wide-carriage printers, e.g., Epson FX100 (OFF).
-#	Print symbol pin numbers (OFF). Pin numbers assigned to pin names with the @ symbol will not be printed.

For example, if you enter the following information at the LOGICAPS prompt,

lcprint -o LPT2: -r -# -f BEVDIS.SD <Enter>

the BEVDIS schematic is printed at accelerated speed to the second parallel printer at 50 % reduction with the symbol pin numbers included.

☞ Unlike a plotted design, a printed design will not correspond to the sheet size specified in the title block because the print size depends on the resolution of the printer. Printed designs are generally smaller. Refer to *Appendix C* for further information.

Sample Session 7

In this session you learn how to:

1. Submit the ADF to the ADP
2. Program the EPLD

1. Submitting the ADF to the ADP

To enter the design sample ADF into the Altera Design Processor (ADP), you must first go into A+PLUS by entering from DOS:

APLUS <Enter>

The APLUS Menu is displayed.

Press **<F4>** to display the ADP Menu.

You are prompted for **Input Format.**

Type:

A <Enter>

Then you are automatically prompted through the remaining functions on the ADP Menu.

For **File Name** enter:	**BEVDIS <Enter>**
For **Minimization** enter:	**Y** or **<Enter>**
For **Inversion Control** enter:	**N** or **<Enter>**
For **LEF Analysis** enter:	**N** or **<Enter>**

For a detailed explanation of these parameters, refer to *A+PLUS and ADP Reference* in the ***A+PLUS Reference Guide***.

After entering all parameters, you are prompted with:

Do you wish to run under the above conditions [Y/N]?

Type **Y** to execute ADP. When the ADP has finished processing the sample design, you are prompted with:

Would you like to implement another design [Y/N]?

Enter **N** to return to the APLUS Menu.

2. Programming the EPLD

Finally, you submit your design to the LogicMap program. While you are still in the APLUS Menu, press <F5> to select LogicMap II. If you have the Logic Programmer card plugged in, the program will come up on the screen.

If you do not have the Logic Programmer card plugged in, the following message is displayed:

Programmer self test failed

Device must not be in socket for this test to pass!

Enter:
C to continue without programming card
T to run diagnostics again
Q to return to operating system

When the LogicMap II System Level Window is displayed, you are asked to wait until the calibrating process is completed. Then the System Level Help Window is opened.

An EP310 is the part used for this design. However, do not put the EPLD into the socket of the programming unit until you are prompted to do so.

With the box cursor select the Program Device function and enter the filename BEVDIS. You are prompted to:

Select Device for Programming

Type:

EP310 <Enter>

LogicMap II automatically checks if the EPLD is erased. Once you have answered all the prompts, programming takes five to ten seconds.

Sample Session 8

In this session you learn how to :

1. Use a bus line in a design with BUSTER primitives and MacroFunctions
2. Use a bus line in an EPLD design (for reference only)

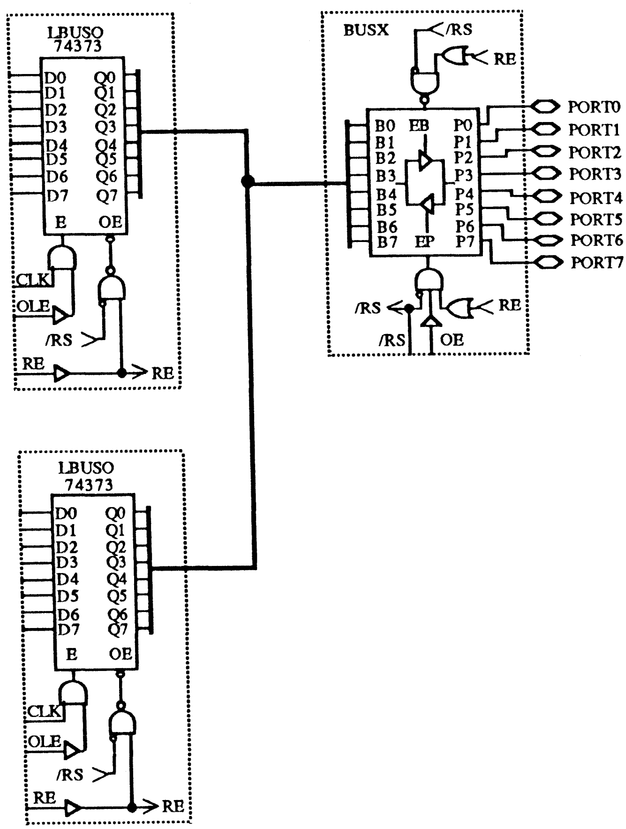
1. Using Bus Lines in BUSTER Designs

Designs for the BUSTER part (EPB1400) may contain bus lines that have full connectivity. Use the <F3> key or the **LS** (Line Select) command to toggle to the bus line selection.

When designing with BUSTER primitives and MacroFunctions, you should keep in mind the following guidelines:

1. A bus is tri-statable.
2. A bus may not connect to an individual signal wire.
3. Only one bus per design is allowed; i.e., all bus lines must connect so that in effect the design contains only one bus line.

Refer to the illustration below:



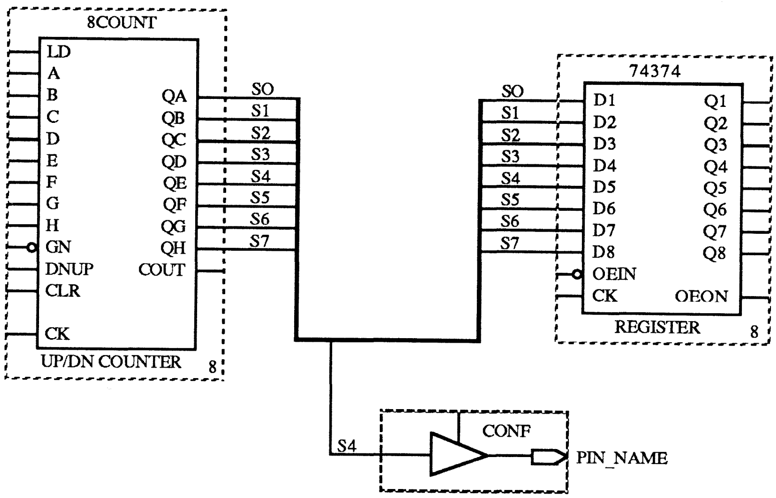
2. Using Bus Lines for Reference in EPLD Designs

Designs for EPLD parts other than BUSTER may contain bus lines that are used for documentation reference *only*. They do *not* establish connectivity. Use the <F3> key or the **LS** (Line Select) command to toggle to the bus line selection.

When you use bus lines for reference only, you should keep in mind the following guidelines:

1. A bus is not tri-statable.
2. A bus may connect to an individual signal wire solely for documentation purposes.

Refer to the illustration below:



SECTION 6

MacroFunction Tutorial

Altera's MacroFunctions are high-level building blocks that, when used together with Altera primitives, greatly increase design productivity. Most MacroFunctions are commonly used TTL SSI and MSI functions; a few are Altera-specific ones that are particularly suited to optimize a logic design with Altera EPLD architecture.

This section describes the use of Altera's MacroFunctions in a schematic design created with LogiCaps schematic capture. The section contains the following information:

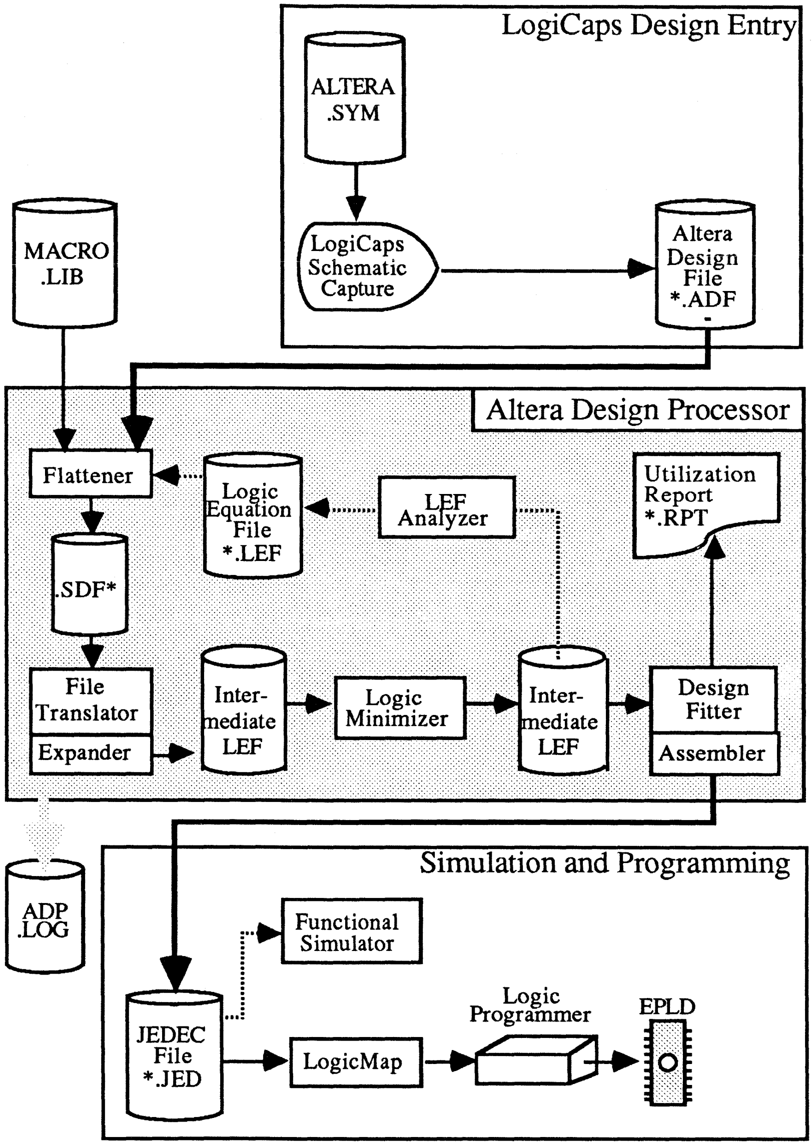
- A functional description
- A tutorial showing how to incorporate MacroFunctions into a logic design.
- A sample circuit containing MacroFunctions

Functional Description

The ALTERA.SYM file on one of your LogiCaps distribution diskettes contains all presently available Altera primitives and standard MacroFunctions. With these symbols you create your logic design with LogiCaps. LogiCaps automatically generates an Altera Design File (ADF) including primitive and MacroFunction statements in the ADF syntax.

The ADF is submitted to the ADP. The Flattener, which is the first ADP module, checks the ADF for the presence of MacroFunction statements. If it finds one, it opens the MACRO.LIB file, gets the logical definition of that MacroFunction, and substitutes the MacroFunction statement with this flattened, expanded statement. The Flattener goes through this process with every MacroFunction statement it finds, producing a secondary design file with the extension .SDF. This flattened and expanded file is then passed on to the Translator and to the rest of the ADP modules. Refer to Figure 6-1.

When the ADP analyzes your design, it removes any unused gates and flipflops from the MacroFunctions present. This feature gives you the freedom to use logic blocks without having to be concerned about obtaining the most efficient use of them. All MacroFunction inputs are defined with default input signal levels that allow you to leave unused inputs unconnected.



*The .SDF file contains flattened MacroFunction statements.
The .ADF file contains unflattened MacroFunction statements.

Figure 6-1. MacroFunctions/A+PLUS Block Diagram

Tutorial

This tutorial describes the MacroFunction symbol, shows how to use MacroFunctions in your LogiCaps design, and gives an example of a circuit containing MacroFunctions.

MacroFunction Symbol Representation

Before designing circuits for Altera parts you should become familiar with the design guidelines listed in Section 4 (*Design Guidelines*). In addition, you should note the format conventions for MacroFunction symbol entry shown in Figure 6-2:

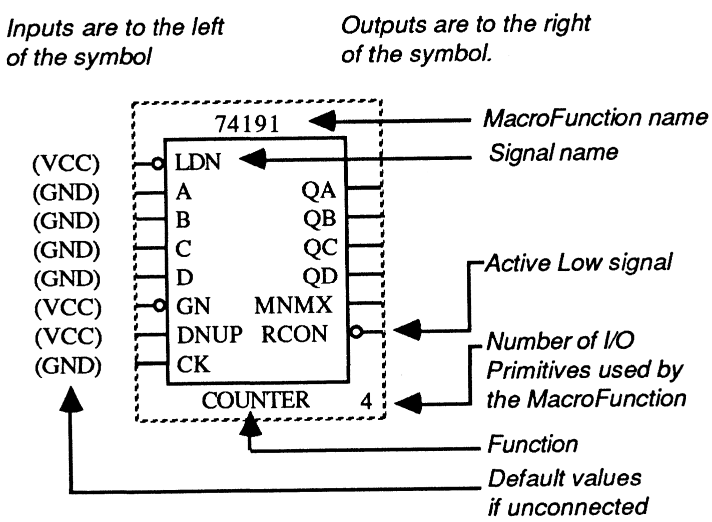


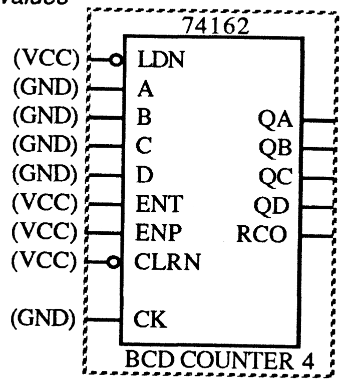
Figure 6-2. MacroFunction Symbol

☞ A few MacroFunction symbols have additional input signals on the right.

Changing the Input Signal Default Value

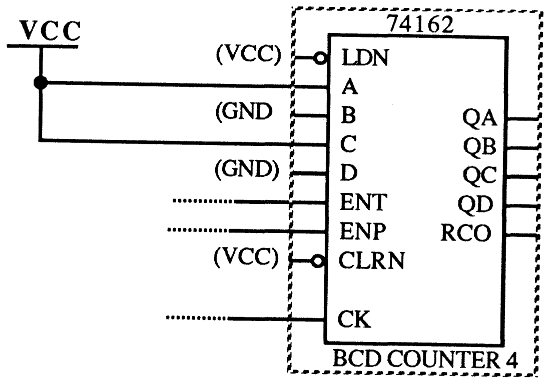
All MacroFunction input signals are assigned default values GND or VCC, in case an input signal is left unconnected. You may override these default values to accommodate your design. For example, if you have a 74162 MacroFunction for which the A and C input signal default values are GND, yet you need to make them VCC, you simply connect these two inputs to a VCC symbol. See the illustration below:

Altera-assigned input signal default values



Altera Primitive

Input signal default value changed by the user



Connecting MacroFunctions

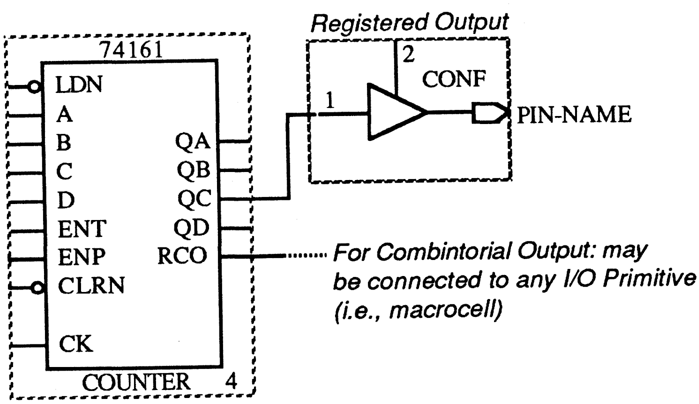
MacroFunctions may be connected to or from other MacroFunctions as well as to or from input, logic, and I/O primitives.

Connection to an External Pin

If a MacroFunction output is to be connected to an external pin of an EPLD, it should be connected directly to a CONF primitive. Use of the CONF primitive enables a process called I/O compression, which ensures that the smallest possible number of macrocells is used in the EPLD. Any other I/O primitive, however, may be used as required by your design.

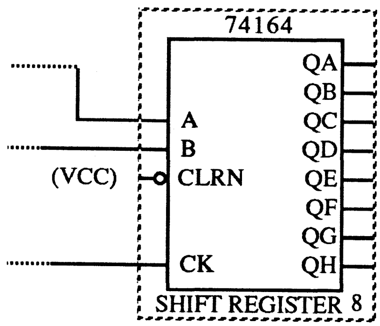
For MacroFunction outputs that are strictly combinatorial output—i.e., they don't have an NORF, NOTF, or NOCF primitive associated with them—no I/O compression is performed, and any I/O primitive may be used without penalty. See the illustration below:

☞ For additional information, refer to I/O Architecture Compression.



Change to Input Signal Default Value

All MacroFunctions are assigned either GND or VCC default input values. If an input is unconnected, the signal simply assumes its default state. (MacroFunction symbols documented in the library of *Standard MacroFunctions* and in the ***TTL MacroFunctions*** manual indicate the defaults in parentheses.) For example, in the illustration below, the CLRN defaults to VCC. This signal is active low, i.e., GND connected to the CLRN input signal will clear the internal flipflops, while VCC connected to the signal disables the clearing function.



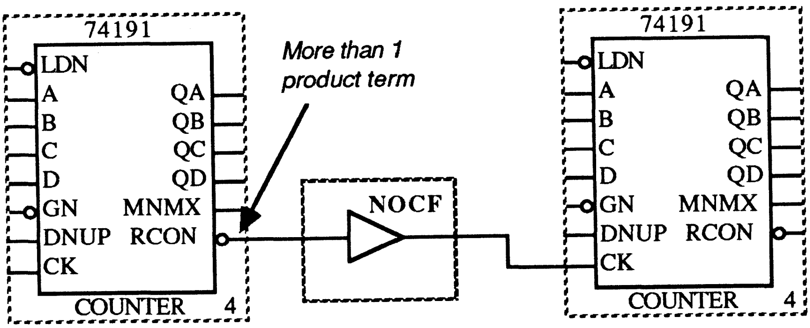
Connection of MacroFunction Outputs

Outputs of MacroFunctions that drive Output Enable, Clock, Clear, or Preset inputs of other MacroFunctions should be first connected to an NOCF primitive, unless the logic driving the MacroFunction input is one product term. The following two examples show:

1. a complex design that requires insertion of an NOCF
2. a simple design that does not require insertion of an NOCF

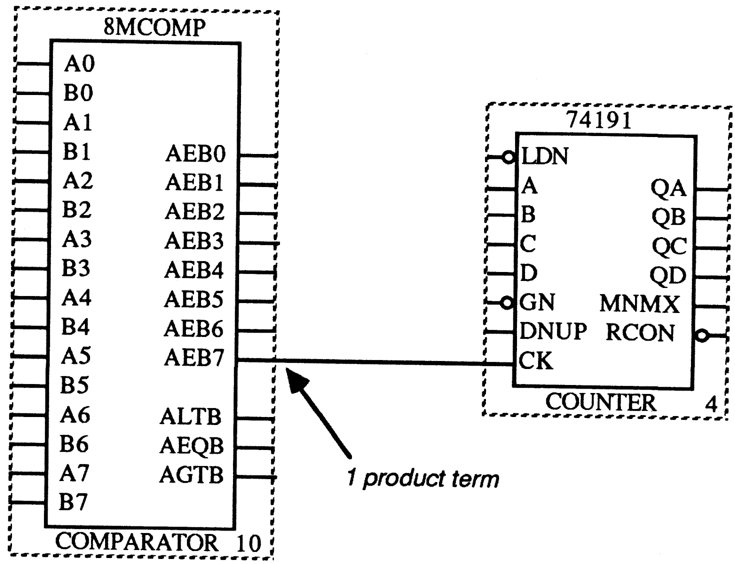
Complex Design

In this design, the RCON output of a 74191 MacroFunction is connected to the CK input of a second 74191 MacroFunction. If you look at the logic schematic of the 74191 MacroFunction—shown in *Standard MacroFunctions* of the ***Reference Guide*** and in the ***TTL MacroFunctions*** manual—you can see that the RCON output is a complex AND/OR function of several signals. Therefore, you need to insert an NOCF primitive.



Simple Design

On the other hand, in the design shown below, where the AEB7 output of an 8MCOMP MacroFunction comes directly from an NOCF primitive—as shown in the 8MCOMP logic schematic in *Standard MacroFunctions* of the ***Reference Guide*** and in the ***TTL MacroFunctions*** manual—no NOCF primitive is required between the two MacroFunctions.

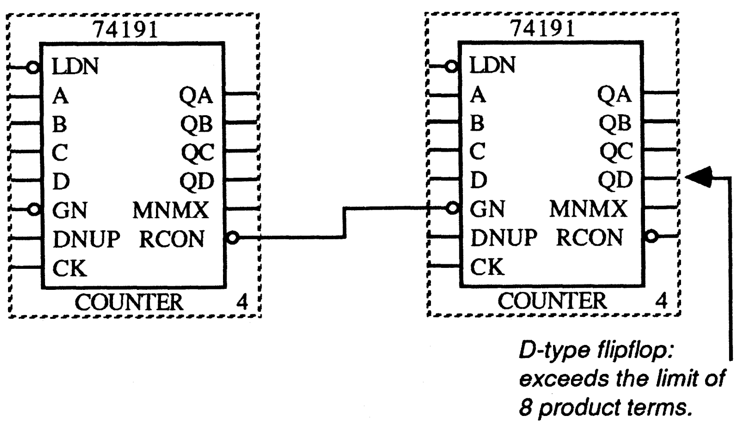


Choosing the Right MacroFunction

Altera-provided MacroFunctions are specifically designed to optimize the architecture of advanced EPLDs. Therefore, if the logic requirements match your design requirements, you should use these special Altera MacroFunctions. To verify the logic requirements of a particular MacroFunction, refer to the individual function table and logic schematic shown in *Standard MacroFunctions* in the ***Reference Guide*** and in the ***TTL MacroFunctions*** manual.

The following example illustrates the advantage of using an Altera-provided MacroFunction:

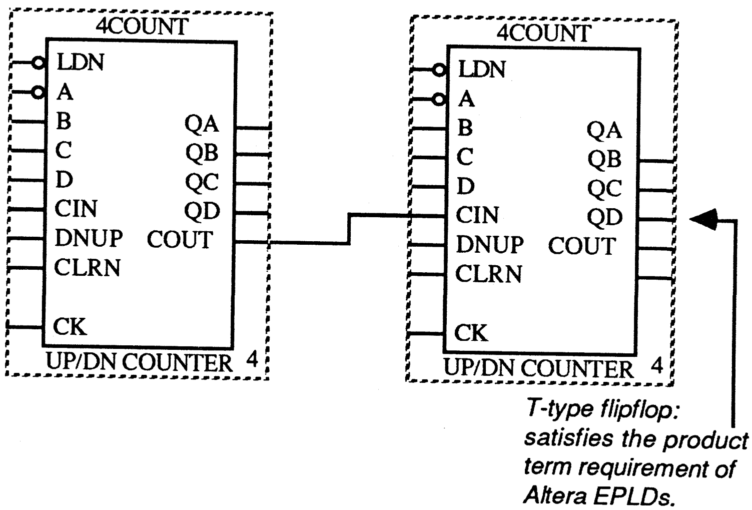
A design requires an 8-bit up/down binary counter with carry-out. You have the choice of using either two 74191 MacroFunctions or two 4COUNT MacroFunctions that must be cascaded together. As the logic schematic shows, a 74191 is implemented with D flipflops; the most significant bit of the counter (8th bit) will exceed the limit of 8 product terms allowed by most Altera EPLDs when compiled by the A+PLUS software. See the illustration below:



On the other hand, as the 4COUNT logic schematic shows, 4COUNT MacroFunctions are implemented with T flipflops; if you use two 4COUNTS, you do not exceed the EPLD requirement of eight product terms.

In fact, the 4COUNT MacroFunctions may be cascaded without ever reaching the product term limit. This feature allows you to quickly design binary counters of arbitrary size.

Refer to the illustration shown here:



I/O Architecture Compression

A+PLUS software automatically performs architecture compression of I/O primitives. Independent of the internal architecture of a MacroFunction, for registered output you always use a CONF primitive to connect a MacroFunction output to an EPLD external pin. For combinatorial output you may use any I/O primitive.

Altera EPLD I/O architecture compression is based on the following equation:

NOxF + CONF = xOxF

where

— **NOxF** (NOCF, NORF, or NOTF) is internal to the MacroFunction architecture and used for registered logic.

— **CONF** is the user-defined output assignment.

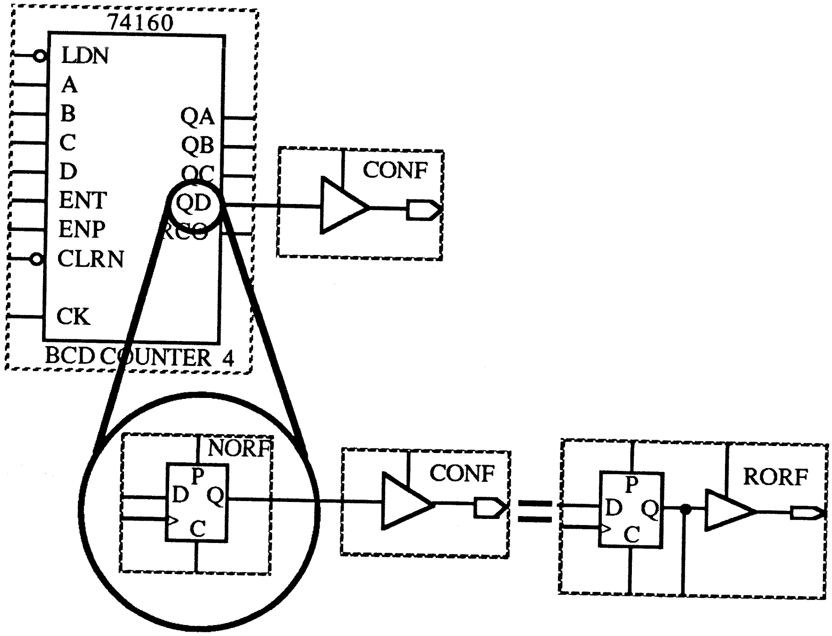
☞ I/O compression does not occur with COIF or COCF outputs.)

— **xOxF** is the resulting I/O architecture.

The following four examples show various implementationsof I/O architecture compression.

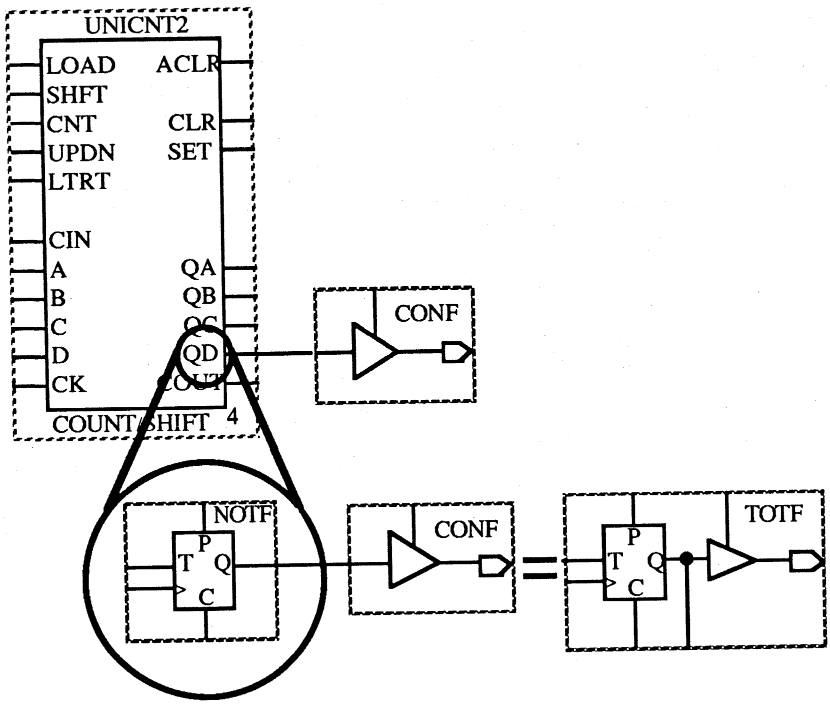
Example 1: NORF + CONF = RORF

Refer to the illustration below. Note that according to the logic schematic of the 74160 MacroFunction, an NORF primitive is connected to the QD output.



Example 2: NOTF + CONF = TOTF

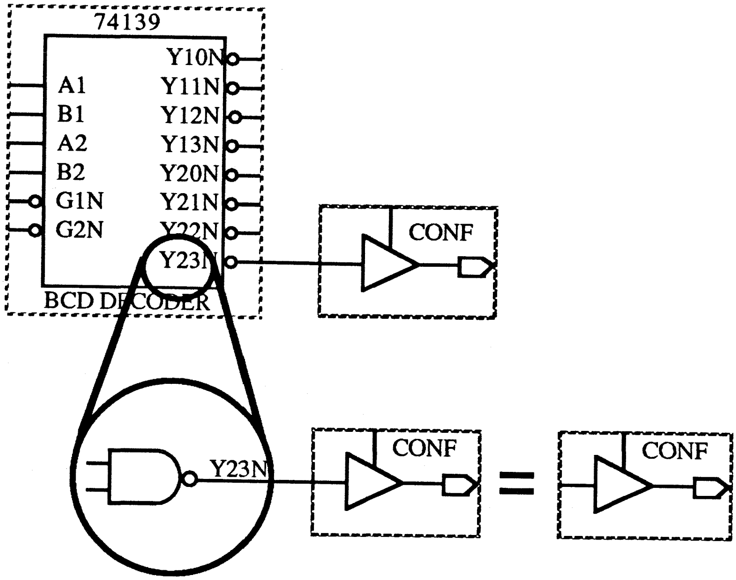
Refer to the illustration below. Note that according to the logic schematic of the UNICNT2 MacroFunction, an NOTF primitive is connected to the QD output.



Internal primitive of MacroFunction *Output primitive* *Resulting compressed output primitive*

Example 3: 0 + CONF = CONF

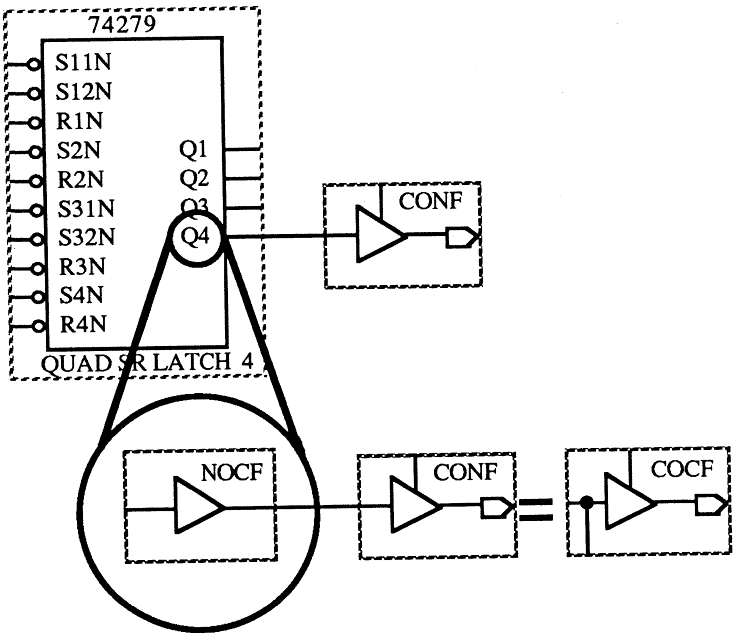
as shown in the illustration below (note that according to the logic schematic of the 74139 MacroFunction, no I/O primitive is connected to the Y23N output):



Internal primitive of MacroFunction *Output primitive* *Resulting compressed output primitive*

Example 4: NOCF + CONF = COCF

as shown in the illustration below (note that according to the logic schematic of the 74279 MacroFunction, an NOCF primitive is connected to the Q1 output):



Internal primitive of MacroFunction *Output primitive* *Resulting compressed output primitive*

MacroMuncher

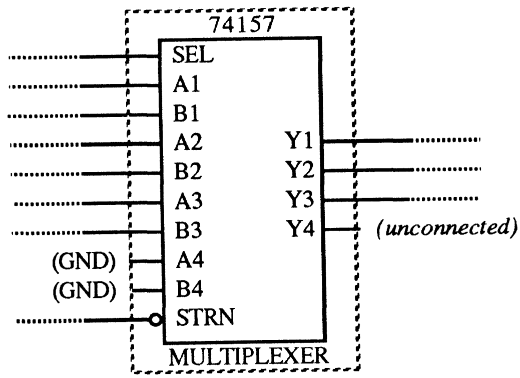
When the Altera Design Processor analyzes a logic circuit, it automatically removes unused gates and flipflops from any MacroFunction in the design. Specifically, the ADP searches the logic circuit for nodes that go nowhere. If it finds such a node, it removes the primitive associated with the node. Then it goes to the next node that is missing a destination and repeats this process. This "MacroMuncher" feature lets you design with MacroFunctions without having to be concerned about whether or not you're making optimal use of the EPLD.

The following two examples demonstrate the MacroMuncher process. The first example has combinatorial output, the second has registered output.

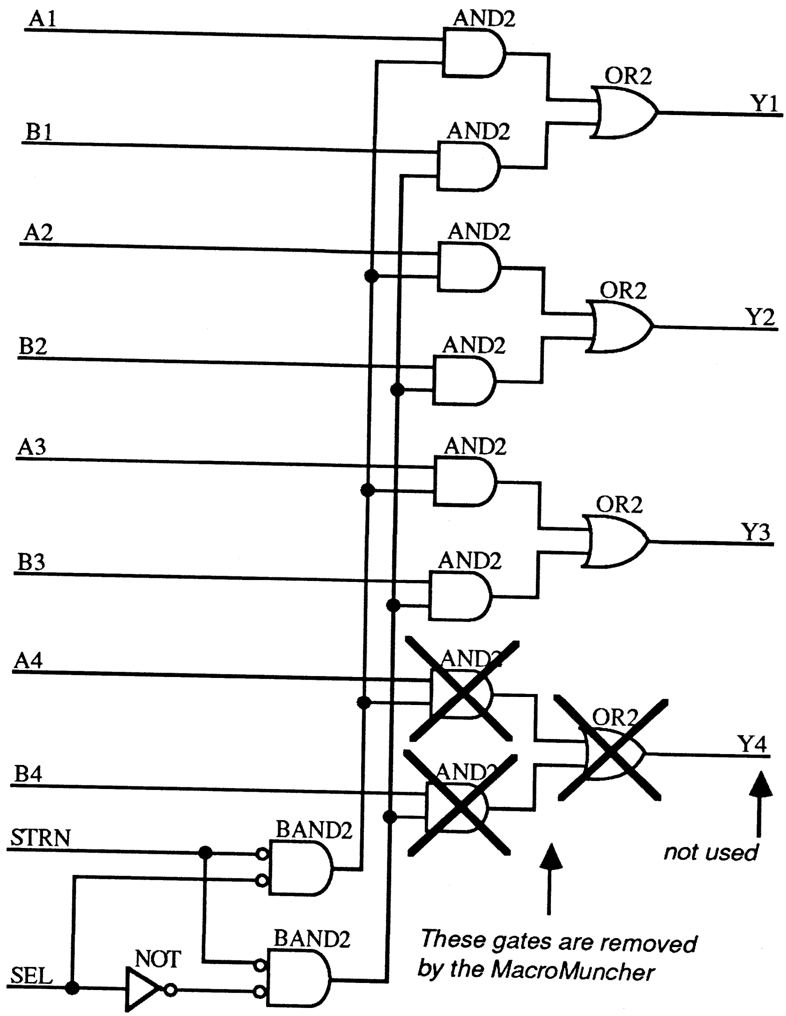
Example 1: Combinatorial Output

In the following illustration, the Y4 output of the 74157 MacroFunction is left unconnected. Consequently, the two AND gates and the OR gate, which represent the internal logic of the Y4 output, are removed from the network. However, the two BAND gates that are also connected to the Y4 output are not removed because they are connected to other primitives in the circuit. (Refer also to *Standard MacroFunctions* in the ***Reference Guide*** and to the ***TTL MacroFunctions*** manual.)

The first illustration below shows the MacroFunction symbol with unconnected output:



The following illustration shows the macro-munching process represented in the internal logic of the 74157:



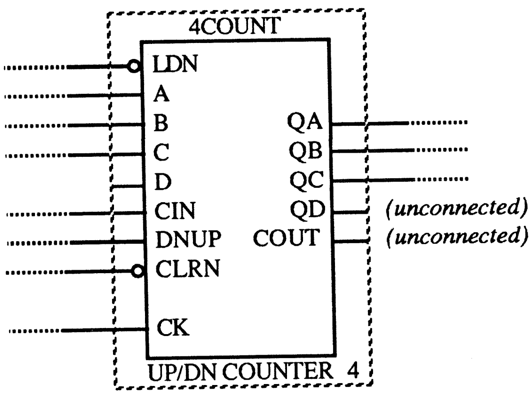
Example 2: Registered Output

This example has two variations.

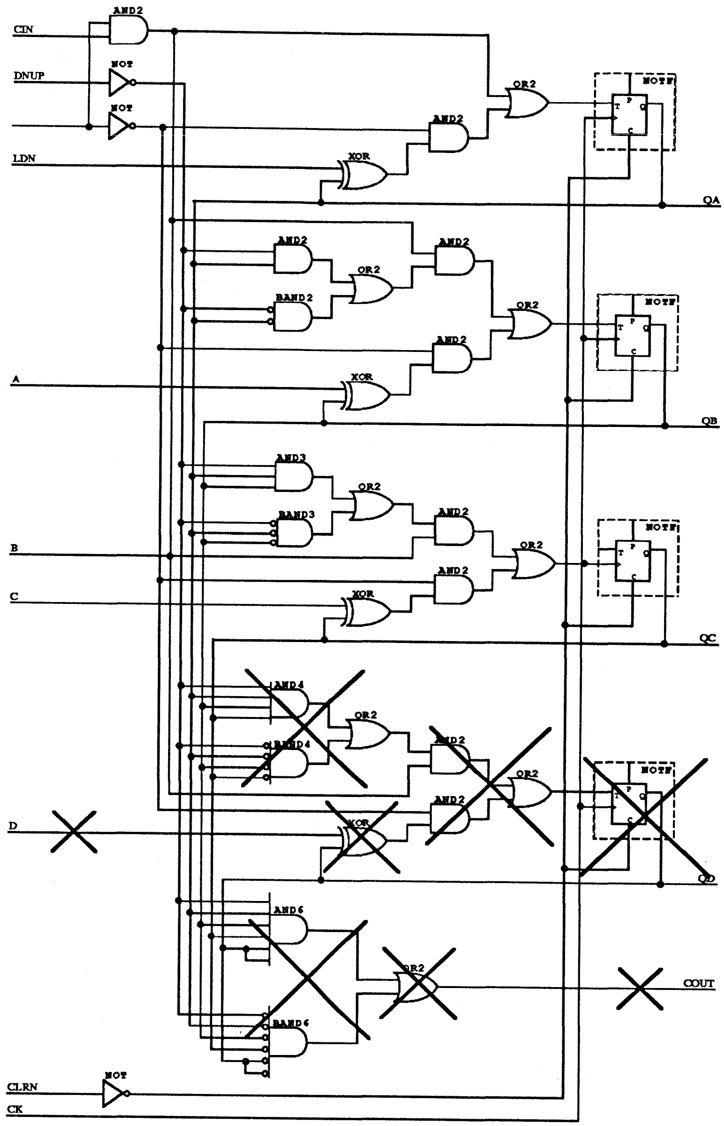
Variation 1:

The MacroFunction used is 4COUNT. However, only a 3-bit counter is actually needed for the design. To satisfy that requirement, the logic associated with the QD and COUT outputs is removed, eliminating the macrocell that would be consumed by QD. As a result, only three macrocells are used.

The illustration below shows the 4COUNT symbol with unconnected outputs:



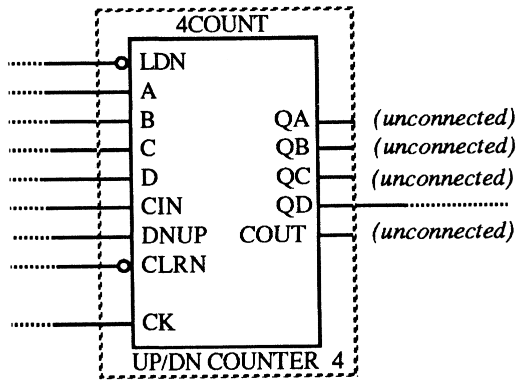
This next illustration shows the internal logic representation of the 4COUNT after the logic associated with QD and COUT has been removed, i.e., "MacroMunched":



Variation 2:

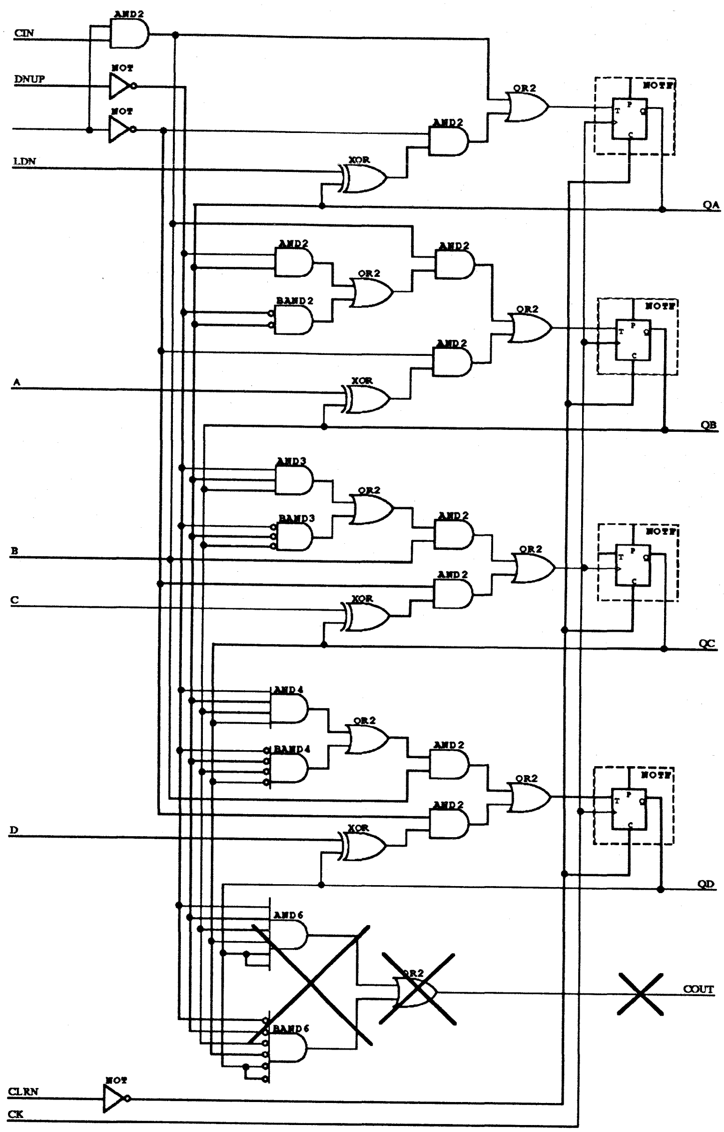
In the second variation, the 4COUNT MacroFunction is used again. However, only the QD output is used, leaving COUT, QA, QB, and QC unconnected.

The first illustration shows the 4COUNT symbol with unconnected outputs:



When you look at the internal logic of 4COUNT (see *Standard MacroFunctions* in the ***Reference Guide*** and the ***TTL MacroFunctions*** manual), notice that COUT remains unconnected and therefore is removed, while QA, QB, and QC remain in the circuit because they are internally connected to QD. As a result, four macrocells are consumed. See the illustration below.

MacroFunction logic schematic after it has been "Macro-Munched":



Example of a Circuit Containing MacroFunctions

The following example shows:

- A functional overview of a design using a Decoder and a Register MacroFunction
- A schematic of the design
- The ADF generated from the design
- The SDF which was generated after the Flattener expanded and flattened the MacroFunction statements in the ADF

Functional Overview

Figure 6-3 shows a functional block diagram of the design sample:

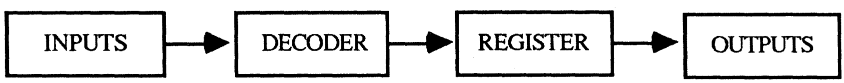


Figure 6-3. Design Sample Block Diagram

Schematic of the Sample Design

Figure 6-4 shows the schematic of the design sample.

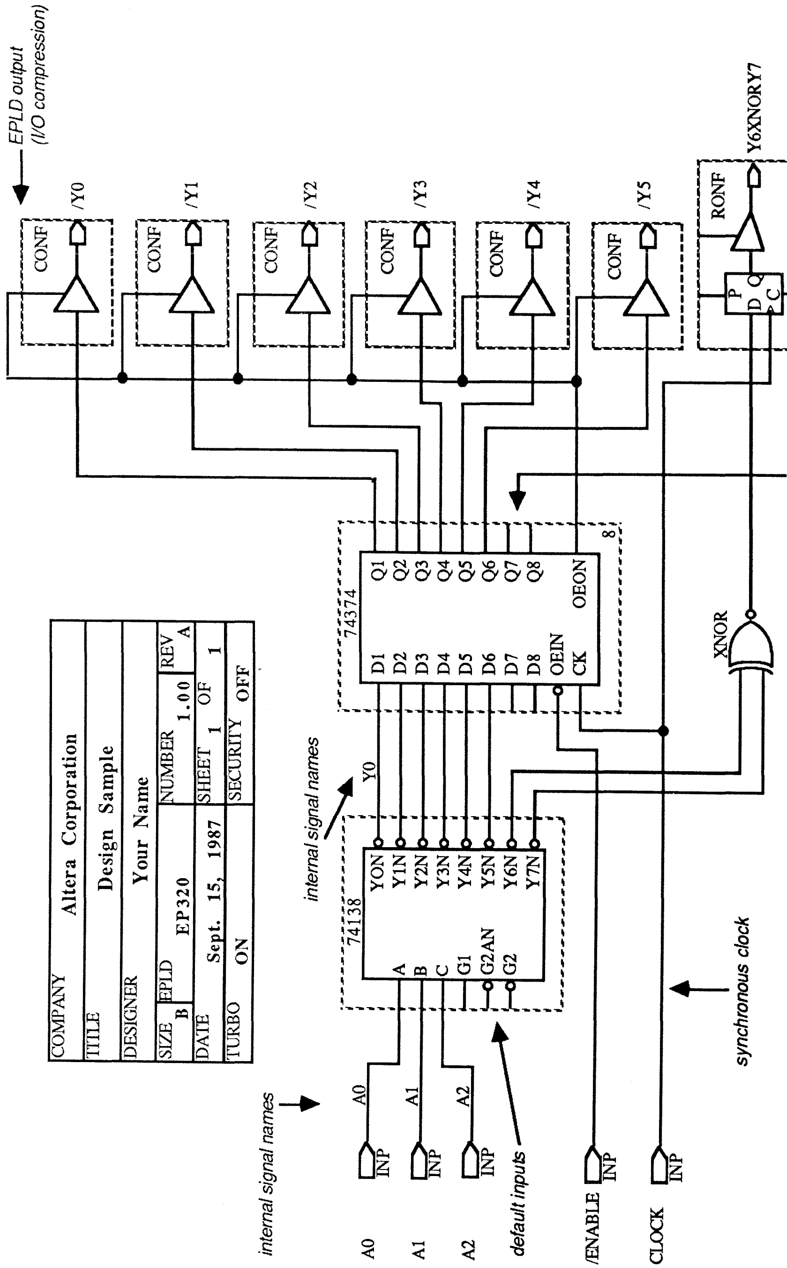


Figure 6-4. Design Sample Schematic

ADF Generated From the Sample Design

Figure 6-5 shows the Altera Design File generated:

```
YOUR NAME
ALTERA CORPORATION
SEPTEMBER 15, 1987
1.00
A
EP320
DESIGN SAMPLE
LogiCaps version 1.6

OPTIONS: TURBO = ON  SECURITY = OFF
PART: EP320
INPUTS:
    A0,A1,A2,/ENABLE,CLOCK

OUTPUTS:
    /Y0,/Y1,/Y2,/Y3,/Y4,/Y5,Y6XNORY7

NETWORK:
74138(,,,A0,A1,A2,..047048,..047046,..047044,
    ..046042,..046040,..046038,..046036,Y0) % SYM 1%
74374(Y0,..046036,..046038,..046040,..046042,
..047044,,,,..063050,..063052,..078052,,,,..078044,
..078042,..078040,..078038,..078036,..078034,)
  % SYM 2 %
/Y0 = CONF(..078034,..078052) % SYM 3 %
/Y1 = CONF(..078036,..078052) % SYM 4 %
/Y2 = CONF(..078038,..078052) % SYM 5 %
/Y3 = CONF(..078040,..078052) % SYM 6 %
/Y4 = CONF(..078042,..078052) % SYM 7 %
/Y5 = CONF(..078044,..078052) % SYM 8 %
Y6XNORY7 = RONF(..095068,..063052,,,) % SYM 9
%
..095068 = XNOR(..047046,..047048) % SYM 10 %
A0 = INP(A0) % SYM 11 %
A1 = INP(A1) % SYM 12 %
A2 = INP(A2) % SYM 13 %
..063050 = INP(/ENABLE) % SYM 14 %
..063052 = INP(CLOCK) % SYM 15 %
END$
```

Header Section

Declaration section

Network section

MacroFunction definition

User-specified signal name

LogiCaps generated signal name= XY coordinates

LogiCaps Symbol number

Figure 6-5. ADF for the Sample Design

SDF Generated From the Sample Design

Figure 6-6 shows the Secondary Design File (SDF) generated from the design sample.

```
YOUR NAME
ALTERA CORPORATION
SEPT. 15, 1987
1.00
A
EP320
DESIGN SAMPLE
LogiCaps version 1.6
OPTIONS: TURBO = ON  SECURITY = OFF
PART: EP320
INPUTS:
A0,A1,A2,/ENABLE,CLOCK
OUTPUTS:
/Y0,/Y1,/Y2,/Y3,/Y4,/Y5,Y6XNORY7
NETWORK:
% 74138(,,,A0,A1,A2,..047048,..047046,
..047044,..046042,..046040,..046038,..046036,
Y0) %
% 74374(Y0,..046036,..046038,..046040,
..046042,..047044,,,,..063050,..063052,
..078052,,,,..078044,..078042,..078040,
..078038,..078036,..078034) %
/Y5, ..078044 = RORF(..047044,..063052,
  GND,GND,..078052)
/Y4, ..078042 = RORF(..046042,..063052,
  GND,GND,..078052)
/Y3, ..078040 = RORF(..046040,..063052,
  GND,GND,..078052)
/Y2, ..078038 = RORF(..046038,..063052,
  GND,GND,..078052)
/Y1, ..078036 = RORF(..046036,..063052,
  GND,GND,..078052)
/Y0, ..078034 = RORF(YO,..063052,GND,
  GND,..078052)
```

MacroFunction definition: now specified as user comment

I/O compression: NORF+CONF = RORF

Figure 6-6. SDF for the Sample Design (Part 1 of 2)

```
Y6XNORY7 = RONF(..095068,..063052,,,)
..095068 = XNOR(..047046,..047048)
A0 = INP(A0)
A1 = INP(A1)
A2 = INP(A2)
..063050 = INP(/ENABLE)
..063052 = INP(CLOCK)
EQUATIONS:
%^% Y0 = /(VCC*GND'*GND'*A0'*A1'*A2');
%^% ..046036 = /(VCC*GND'*GND'*A0*A1'*A2');
%^% ..046038 = /(VCC*GND'*GND'*A0'*A1*A2');
%^% ..046040 = /(VCC*GND'*GND'*A0*A1*A2');
%^% ..046042 = /(VCC*GND'*GND'*A0'*A1'*A2);
%^% ..047044 = /(VCC*GND'*GND'*A0*A1'*A2);
%^% ..047046 = /(VCC*GND'*GND'*A0'*A1*A2);
%^% ..047048 = /(VCC*GND'*GND'*A0*A1*A2);
%^% ..078052 = /..063050;
END$
```

Specifies that logic is from MacroFunction

Logic equations defining MacroFunction logic

Figure 6-6. SDF for the Sample Design (Part 2 of 2)

SECTION 7

LogiCaps Commands

This reference section provides general information about LogiCaps features and a detailed description of each command available in LogiCaps.

Invoking LogiCaps

To invoke LogiCaps, type at the DOS prompt:

LOGICAPS [filename] [@macroname] [options] <Enter>

You may use the following optional specifications at the time of invoking LogiCaps:

filename (optional) The name of a drawing file that is automatically loaded.

macroname (optional) The name of a macro recorded and written to a file with the extension .MAC. It is used in place of INIT.MAC.

options Additional configuration specifications that may be entered at the time of LogiCaps invocation. The available options are:

–1 Use COM1 for the mouse (default).

–2 Use COM2 for the mouse.

–l <symbol library> Selects a different symbol library. The default is ALTERA.SYM.

–ml Increase the memory for LINE objects. This option allows you to reallocate memory to increase the memory capacity for lines.

–ms Increase the memory for SYMBOL objects. This option allows you to reallocate memory to increase the memory capacity for symbols.

–mt Increase the memory for TEXT objects. This option allows you to reallocate memory to increase the memory capacity for text entries.

–x Set new file configuration data. Lets you ignore configuration specifications in LOGICAPS.CFG and prompts you for new configuration information.

To get a screen display of this information, type at the DOS prompt:

LOGICAPS ? <Enter>

Command Entry

When designing with LogiCaps, you may enter commands in three ways:

- You may type a command on the keyboard
- You may use the mouse
- You may combine keyboard and mouse commands

The commands are divided into the following groups:

- Mouse Commands
- Function Key Commands
- Text and Command Line Argument Editing
- Command Line Commands

Mouse Commands

The mouse provides the quickest command execution. Most drawing commands such as moving, copying, deleting can be executed with the mouse.

However, if for any reason you are unable to use a mouse, note that MKY mode puts the cursor into mouse mode, allowing it to imitate mouse movements. Refer to the description of the <+> function.

Basically, the **<Left>** mouse button is for moving objects, the **<Center>** mouse button is for copying objects, and the **<Right>** mouse button is for canceling move and copy commands. More specifically, the mouse buttons perform the following functions:

<Left> Draw, tag, and move lines.
Tag and move symbols, areas, and text fields.
The comma key (<,>) emulates the left mouse button.

<Center> Enter and remove connection dots.
Tag and copy symbols, areas, and text fields.
The period key (<.>) emulates the center mouse button.

<Right> Cancel commands.
Delete lines.
The slash key (</>) emulates the right mouse button.

To find out how to use the mouse to execute a specific function, refer to the Mouse Command description of that function under Command Line Commands.

Mouse movements differ slightly from arrow key movements because the mouse moves relative to the screen, not the drawing. Therefore, at different zoom levels, the mouse will move in different units, while the arrow keys always move the cursor in multiples of a database unit (DBU). When recording a macro, you should therefore also record the zoom level.

Function Key Commands

Some of the most frequently used functions are assigned to the function keys. These functions are generally available while the Main menu is displayed on the menu line, except for the **<Esc>** function, which always returns you to the Main menu.

Each command is described on a separate page for quick reference. The information is presented in the following format:

Syntax:	Gives the command as it is used during design entry.
Function:	States the function of the command.
Description:	Describes additional details.
Note:	When appropriate, notes are added for further information.

☞ You may assign macros to any or all of the keys **<F1>** to **<F10>**. These macro assignments override the functions originally assigned to these keys. (Refer to the description of the **MA** (Macro Assign) command.)

<F1> (Recall a Window View)

Syntax: **<F1>**

Function: Recalls a view of the schematic that has been previously identified and saved with the **WT** (Window Tag) command. This key performs the **WR <F1>** function described under **WR** (Window Recall).

Description: When you press **<F1>**, the message line displays the message:

View recalled by <F1>

To repeatedly recall a certain view of your design, you can tag a desired partial view with the **WT<F1>** command. Subsequently, whenever you press **<F1>**, the cursor returns to that view of the drawing.

The **<F1>** command saves (1) the cursor position in the drawing, (2) the window location in the drawing, and (3) the zoom level in effect at the time the **WT** command was executed.

Note: **<F2>** is assigned to perform the same function to save a second partial view of the design. With these two functions keys, you can quickly jump back and forth between two views of the schematic.

(Recall a Window View)

Syntax: **<F2>**

Function: Recalls a view of the schematic that has been previously identified and saved with the **WT** (Window Tag) command. This key performs the **WR <F2>** function described under **WR** (Window Recall).

Description: When you press **<F2>**, the message line displays the message:

View recalled by <F2>

To repeatedly recall a certain view of your design, you can tag a desired partial view with the **WT<F2>** command. Subsequently, whenever you press **<F2>**, the cursor returns to that view of the drawing.

The **<F2>** command saves (1) the cursor position in the drawing, (2) the window location in the drawing, and (3) the zoom level in effect at the time the **WT** command was executed.

Note: **<F1>** is assigned to perform the same function to save a second partial view of the design. With these two functions keys, you can quickly jump back and forth between two views of the schematic.

<F3> (Select Line Type)

Syntax: **<F3>**

Function: Selects the next line type. The current line type is displayed in the LINE: field in the status line and the message line parallels the information.

Description: The status line indicates the current line type. By pressing **<F3>** repeatedly, you can cycle through the following available line types (the solid line is the default):

Solid line:

Dotted line:

High density dotted line:

Bus line:

Dashed line:

Note: For more details, refer to the description of the **LS** (Line Select) command.

<F3> always selects the line type available next as you cycle through the five line types. In contrast, when you execute the **LS** command, the selection always starts with the solid line type, regardless of which type is currently displayed in the status line.

When you record a macro, be sure to select a line type with the **LS** command not the **<F3>** key.

<F4> (Select Character Type)

Syntax: **<F4>**

Function: Selects the next character type used for text entries. The current character type is displayed in the TXT: field in the status line and the message line parallels the information.

Description: The TXT field in the status line indicates the current character type. By repeatedly pressing **<F4>**, you can cycle through the following character types (medium size is the default):

Small , caps only:	SIG
Medium:	Sig
Large:	Sig
Reverse, medium only:	Sig

For detailed information refer to the **TS** (Text Select) command.

<F4> always selects the character type available next as you cycle through the four character types. In contrast, when you execute the **TS** (Text Select) command, the selection always starts with the medium size character type, regardless of which type is currently displayed in the status line.

When you record a macro be sure to select a character type with the **TS** command not the **<F4>** key.

<F5> (Toggle Area)

Syntax: **<F5>**

Function: Toggles the area boundary ON or OFF.

Description: If an area has been previously defined with the **AB** command, pressing **<F5>** "undefines" it and removes the area box.

Note: Refer to the description of the **AT** (Area Toggle) command for additional details.

<F6> (Define Area Boundary)

Syntax: **<F6>**

Function: Enters Area Boundary Definition mode and displays the Area cursor, which is initially a large, flashing cross and then turns into a box cursor when the cursor is moved.

Description: When you press **<F6>** you are prompted with:

Area Definition Mode - <Left> to complete area definition

Move the cursor until the area box cursor contains all the objects you want inside the defined area. Then press **<Enter>** or **<Left>** to complete the area definition and return to normal mode. This command is canceled if you press **<Right>** or **<Esc>**.

Note: Refer to the **AB** (Area Boundary) command for additional details.

<F7> (Enter Text)

Syntax: **<F7>**

Function: Initiates Text Enter mode.

Description: Place the graphics cursor at the location where you wish to enter the text. When you press **<F7>**, the graphics cursor is replaced with the text cursor and you are prompted with:

Type text then press <Esc> to quit

Enter the desired text and press <Esc> to return to the graphics cursor.

Note: Refer to the description of the **TE** (Text Enter) command for additional details.

<F8> (Enter a Symbol)

Syntax: **<F8>**

Function: Initiates Symbol Enter mode.

Description: Place the graphics cursor at the location where you wish to enter the symbol. When you press **<F8>**, you are prompted with:

Enter symbol name (? for directory), then press <Enter>

Type the name of the desired symbol on the command line. When you press **<Enter>**, the symbol is displayed with the cursor located in the upper left hand corner.

As long as the command line displays Symbol Enter mode, you may enter the same symbol at a different location by simply moving the cursor to the desired location and pressing **<Enter>**.

Note: Refer to the **SE** (Symbol Enter) command for additional details.

<F9> (Toggle Grid Display)

Syntax: **<F9>**

Function: Toggles the grid to ON and OFF.

Description: Default grid spacing is (10,10).

The **WG** (Window Grid) command sets the grid location and spacing. The smallest unit of the grid is 1 database unit (DBU); there are 10 DBUs per inch. One-inch spacing is the default.

Note: For complete details on the grid, refer to the **WG** (Window Grid) command and to the description of the grid in *Getting Started*.

<F10> (Toggle Window Split)

Syntax: **<F10>**

Function: Splits the window display. This is a toggle function.

Description: When you initially press **<F10>**, the screen display is split into two halves both showing part of the entered design. One half has a solid line boundary, indicating that this is the active, currently tracking window; the other half has a dashed line boundary, indicating that this is the inactive, non-tracking window.

When you press **<F10>** again, the full screen is redisplayed. The window that was active at the time of pressing the key, is the one that is displayed in full screen mode.

The **<Tab>** key toggles the windows from active to inactive.

Note: For complete details on the split window, refer to the **WS** (Window Split) command and to the description of the window in *Introduction*.

<Tab> (Leap Command)

Syntax: **<Tab>**

Function: Leap command.

Description: The current location of the cursor is saved and the cursor leaps to a location saved by a previous leap command. With this function you can quickly jump back and forth between two locations.

In Split Window mode, **<Tab>** switches a window from being the currently active one to being the inactive one.

Note: Refer also to the descriptions of **<F10>** and the **WS** (Window Split) command.

<Home> (Home the Cursor)

Syntax: **<Home>**

Function: Homes the cursor to the center of the screen.

Description: The window is repositioned so that the cursor is centered. However, the cursor position on the drawing is unaltered.

<+> (Enter Cursor MKY Mode)

Syntax: <+>

Function: Puts the arrow keys into mouse mode. This is a toggle function.

Description: When you press <+>, the message line shows the following message:

Arrow keys set for MOUSE

In this mode the cursor moves a constant distance on the screen regardless of the window scale. The CRSR: field on the status line shows MKY while in this mode.

Note: Refer also to the description of Mouse Commands.

If for any reason you are unable to use a mouse, MKY mode will substitute for the mouse.

<PgUp> (Zoom Out)

Syntax: **<PgUp>**

Function: Zooms out. The objects on the screen become smaller.

Description: When the LogiCaps working screen is initially displayed, entered objects appear in full scale (default). If you press **<PgUp>**, the view is reduced to a smaller, intermediate scale; if you press **<PgUp>** repeatedly, the view is reduced to maximum reduced scale, where the full drawing (e.g., A, B, CV, etc.) is displayed.

Note: Use the **<PgUp>** and **<PgDn>** keys to set a zoom level; then use the **WZ** (Window Zoomset) command to save the zoom level for the **<End>** key. Subsequently, you may use the **<End>** key to quickly change to the saved zoom level while you are editing.

☞ Alphanumeric characters are visible only in full scale, 3/4 scale, and 1/2 scale modes. In the two reduced scale modes, alphanumeric characters will not be proportionately scaled on the screen or on the printed design. However, they will be scaled proportionately on the plotted design.

<PgDn> (Zoom In)

Syntax: **<PgDn>**

Function: Zooms in. The objects on the screen become larger.

Description: When an object appears in reduced scale, pressing **<PgDn>** enlarges the view. In maximum reduced scale, pressing **<PgDn>** enlarges the view to an intermediate scale; repeatedly pressing **<PgUp>**, enlarges the view to full scale.

Note: You may use the **<End>** key to quickly change to the full screen or maximum reduced screen zoom level while you are editing.

Alphanumeric characters are visible only in full scale, 3/4 scale, and 1/2 scale modes. In the two reduced scale modes, alphanumeric characters will not be proportionately scaled on the screen or on the printed design. However, they will be scaled proportionately on the plotted design.

<End> (Go Directly to Full Scale or Maximum Reduced Scale)

Syntax:	**<End>**
Function:	Toggles between full and maximum reduced scale.
Description:	If you happen to be in an intermediate scale, then this function will go to either full scale or maximum reduced scale.
Note:	The maximum reduced scale is a function of the drawing size. (Refer also to the description of the **DS** command.)

<*> (Redisplay Previous Message)

Syntax: <*>

Function: Redisplays the message most recently displayed on the message line.

Description: A displayed message is removed from the message line if you press any key or mouse button or if you move the mouse. You may redisplay it by pressing <*>.

<Space> (Refresh the Screen)

Syntax: **<Space>**

Function: Refreshes the window. Pressing **<Space>** twice refreshes the entire display.

<Esc> (Cancel a Command)

Syntax: **<Esc>**

Function: Cancels a command.

Description: Clears any menu selection without executing the command.

Terminates object Move and Copy modes.

Terminates Area Define and Line Draw modes.

Is used to exit Text Enter and Text Edit modes.

Text and Command Line Argument Editing

Command line editing commands are used to edit text entered from the keyboard prior to pressing **<Enter>**. These commands are active when you enter filenames, symbol names, etc. on the command line and also when you enter and edit text fields in the drawing.

The following editing commands are available:

<←>	Move the text cursor left.
<→>	Move the text cursor right.
<Ins>	Toggle character insertion mode.
<Del>	Delete character at text cursor.
<Backspace>	Delete the character left of the text cursor.
<End>	Move to the end of the text.
<Home>	Move to the beginning of the text.
<Ctrl><←>	Move the text cursor to the end of the previous word.
<Ctrl><→>	Move the text cursor to the beginning of the next word.
<Ctrl><End>	Move the text cursor to the right to the nearest pin number.

Command Line Commands

Most functions available in LogiCaps can be executed from the command line. This section helps you to quickly locate a detailed description of each command.

In keeping with the LogiCaps menu structure, this section is divided into nine subsections. However, unlike the menu line on the display screen, the main menu commands and the commands listed under each main menu entry are listed in alphabetical order.

You enter a command by pressing one or two letter keys corresponding to the first letter(s) of the command name. LogiCaps echos the keypress by displaying the full command name including a space. For example, to enter the **SR** (Symbol Reflect) command, you type **S** then **R** and then press **<Enter>** while in Main menu. Some commands, such as **SE** (Symbol Enter) or **DL** (Drawing Load), prompt you for additional text before you may press **<Enter>**.

You execute a command by typing a command followed by any required parameters and then pressing **<Enter>**. The typed command is executed, and the command line cursor is moved to the beginning of the line. However, the command is not removed from the line. You can press **<Enter>** again to re-execute the same command, or you can edit it with the command line editing commands.

Main menu commands are listed in alphabetical order as follows:

1. Area Commands
2. Drawing Commands
3. Help Command
4. Line Commands
5. Macro Commands
6. Quit Command
7. Symbol Commands
8 Text Commands
9. Window Commands

Each Main menu command section is preceded by a summary of the most important features of that menu.

The submenu commands are listed in alphabetical order under each one of the Main menu commands.

Each command line command is described on a separate page for quick reference. The information is presented in the following format:

Syntax:	Gives the command as it is used during design entry.
Function:	States the function of the command.
Description:	Describes additional details.
Note:	When appropriate, notes are added for further information.

Mouse Command

The mouse command describes how a command can be entered with the mouse. Note that a few commands, e.g., **SE**, cannot be entered with the mouse. Others, e.g., **SM**, may be entered with the mouse only.

If a command cannot be executed with the mouse, this part of the command description is deleted.

and/or

Function Key Assignment

Some frequently used commands are assigned to function keys for quick and easy execution. This part of the command description states which function key is assigned to a specific command.

Remember that the function initially assigned to any one of the function keys, **<F1>** to **<F10>**, may be overridden with the **MA** command.

This part of the command description is deleted if it is not applicable to a specific command.

The Area Menu

An area is a rectangular portion of the drawing definable by you. It contains one or more objects such as symbols, lines, and text fields.

Once you have defined an area, you can move, copy, and delete it as though it were one single object. You can even save it to a file, recall it at any time, and insert it into a new drawing or at a different location in your drawing. When an area file is recalled, the area is inserted at the cursor location (provided it does not conflict with already existing objects) without disturbing the current display.

A symbol and a text field are included in an area only if they are completely inside the area boundary. A line crossing the area boundary is broken at the boundary, and only the segment inside the area boundary is considered to be a part of the area.

The area boundary is a flashing dotted box that turns into a stable high-density dotted box when you exit the Area Boundary mode.

Only one area may be defined in a drawing at any given time. This area may be toggled OFF; then it can be toggled ON again, so that it is redisplayed.

☞ Any area file may be loaded as a drawing and vice versa, provided the name is changed from <filename>.SD to <filename>.SDA.

Commands are listed in the following order:

Boundary — Copy — Delete — Files — Help — Load — Move — Toggle — Write

AB (Area Boundary)

Syntax: **AB <Enter>**

Function: Defines an area boundary.

Description: Position the graphics cursor on the screen in a location that would be one of the four corners of the area you wish to define. Type **AB**. You are prompted with:

Press <Enter> to define area

The graphics cursor takes the form of a flashing dotted cross that turns into a flashing box cursor as soon as you move the mouse or use the arrow keys. Move the cursor until the box contains all the objects you wish to define with the area. Press **<Enter>**. The box cursor is replaced by a densely dotted area boundary box, and the graphics cursor is redisplayed at the corner opposite the one you started in.

If an area is already defined and displayed and the graphics cursor is on the corner or on the edge of the area, you can edit the size of the area. Place the cursor on that corner from which you want to start the change and press **<Enter>**: the box size can be enlarged or reduced. The corner diagonally across from the corner where the cursor is positioned does not move.

Note: Symbols and their associated alphanumeric fields and any other alphanumeric fields that cross the defined area boundaries are not included in the defined area.

Mouse Command

1. Place the cursor in a location that will be one of the four corners of the area you wish to define.
2. Type **AB <Enter>** at the command line.
3. Move the mouse until it includes all the objects to be defined by the area.
3. Press **<Left>** to display the area boundary and exit Area Enter mode.

Function Key Assignment

<F6> Enters Area Boundary Definition mode and displays the Area cursor, which is initially a large, flashing cross and then turns into a box cursor when the cursor is moved.

AC (Area Copy)

Syntax: **AC <Enter>**

Function: Copies the contents of a previously defined area to a new location.

Description: This command is effective if an area has been previously defined with the **AB** command. Type **AC**. You are prompted with:

Press <Enter> to copy area

The command causes the graphics cursor to take the form of the area copy cursor, which is a flashing dotted box the same size and location as the area to be copied. This box may be moved anywhere in the drawing as long as it does not overlap with other symbols. When you press **<Enter>** again—i.e., the command line still shows the **AC** command—the command is executed and the contents of the defined area are copied to the new location. Area Copy mode is terminated and the graphics cursor is redisplayed.

Note: Symbols and their associated alphanumeric fields and any other alphanumeric fields that cross the defined area boundaries are not copied with the area.

Mouse Command

1. Place the cursor inside the area to be copied.
2. Press **<Center>** and move the mouse to the new location.
3. Press **<Center>** again to copy the area.
4. Press **<Right>** to terminate Area Copy mode.

AD (Area Delete)

Syntax: **AD <Enter>**

Function: Deletes the contents of a previously defined area.

Description: This command can be executed if an area has been previously defined with the **AB** command. Type **AD**. You are prompted with:

Press <Enter> to delete data within area

The command erases the area boundary and all objects inside of it.

Note: Symbols and their associated alphanumeric fields and any other alphanumeric fields that cross the defined area boundaries are not deleted with this command.

AF (Area Files)

Syntax: **AF <Enter>**

Function: Displays a list of drawing file names.

Description: Type **AF**. You are prompted with:

Enter path [current], then press <Enter>

A list of filenames with the extension .SDA is displayed. If more files exist than can be shown on one screen, you are prompted with:

Press <Space> for more...

Press any other key to redraw the display and return to Main menu.

Note: You may type a path or drive name before pressing **<Enter>**. If you do not specify a path or drive name, the current drive or path will be used.

AH (Area Help)

Syntax: **AH <Enter>**

Function: Displays helpful information about all area commands.

Description: Type **AH**. You are prompted with:

Press <Enter>for info about AREA commands

The Area Help file describes all the Area commands. You may call it up at any time, then press any character to return to your circuit display.

If the file contains more text than can be displayed on the screen, you are prompted with:

Press <Space> for more...

The screen scrolls to display more help information.

AL (Area Load)

Syntax: **AL [filename] <Enter>**

Function: Loads a drawing file into the currently displayed drawing. The loaded file is treated as an area and is displayed so that its upper left hand corner is at the cursor location.

Description: You may use this command if you have previously saved an area with the **AW** command. Type **AL**. You are prompted with:

Enter filename[.SDA] (? for directory), then press <Enter>

You may specify an area filename. If you don't, the default name AREA.SDA is used in locating the file. The area is loaded with its upper left hand corner at the cursor location, provided there is no symbol overlap.

In addition, you may load the file of an entire drawing with the **AL** command. The drawing is then treated as an area, i.e., it will be displayed with its upper left hand corner at the cursor position (provided there is no symbol overlap) without erasing whatever is currently displayed on the screen. This function allows you to use entire drawing files as portions of new files, provided you rename the file from <filename>.SD to <filename>.SDA.

Note: The filename may include drive and path information, but any extension other than .SDA will be replaced with .SDA before loading.

AM (Area Move)

Syntax: **AM <Enter>**

Function: Moves a previously defined area to a new location.

Description: This command is effective if the area has been previously defined with the **AB** command. Type **AM**. You are prompted with:

Press <Enter> to move area

When you press **<Enter>** and the cursor is outside of the area, it jumps to the closest corner of the area and is redisplayed just inside that corner. The command causes the graphics cursor to take the form of the area move cursor, which is a flashing dotted box the same size and location as the area to be copied. This box may be moved anywhere in the drawing. However, symbols may not overlap. When you press **<Enter>** again, the command is executed and the contents of the defined area are moved to the new location. Area Move mode is terminated and the graphics cursor is redisplayed.

When rubberbanding is enabled, line connections for lines passing through area boundaries are retained. Refer to the **LR** (Line Rubberband) command.

Note: Symbols and their associated alphanumeric fields and any other alphanumeric fields that cross the defined area boundaries are not moved with this command.

Mouse Command

1. Place the cursor inside the area to be moved.
2. Press **<Left>** twice and move the mouse to the new location.
3. Press **<Left>** to execute the move and redisplay the graphics cursor or **<Right>** to abort the command.

AT (Area Toggle)

Syntax: **AT <Enter>**

Function: Toggles the area boundary. If you press **<Enter>** again, the boundary line reappears.

Description: If an area has been previously defined with the **AB** command, the **AT** command "undefines" it and removes the area box.

Type **AT**. You are prompted with:

Press <Enter> to remove area definition

Note: When the area boundary is invisible, the area is considered to be "not defined." The **AT** command redefines it and redisplays the boundary.

Function Key Assignment

<F5> Toggles the area boundary to ON or OFF.

AW (Area Write)

Syntax: **AW [filename] <Enter>**

Function: Writes the currently defined area as a drawing file with the extension .SDA.

Description: Type **AW**. You are prompted with:

Enter filename[.SDA], then press <Enter>

If the area has been previously defined, its contents are saved to disk. If no filename is specified, the default filename AREA.SDA is given to the file.

The filename may include drive and path information. However, any extension other than .SDA is replaced with .SDA before the file is saved.

Note: Symbols and alphanumeric fields that cross the defined area boundaries are not saved with this command.

If no area is currently defined, this command has no effect.

The Drawing Menu

The Drawing Menu functions allow you to save, load, and delete drawings. Also, with this menu you select a drawing size, call up a directory of drawing files (default extensions .SD), and generate the Altera Design File (ADF) for your drawing. The ADF is used by the Altera Design Processor (ADP) to process your drawing into a standard JEDEC file that is then used by LogicMap II to program the EPLD.

For correct processing, the ADF must meet a number of requirements listed here.

Design Guidelines To Generate an ADF

- Pin names may be up to eight characters long and may contain any character except percent symbol (%), comma (,), equal-symbol (=), at-symbol (@), or left and right parentheses (()).

- A pin number is assigned to a pin name or buried register output by entering an at-symbol (@) and the specific number after the signal name. Name, @, and number together may not exceed 11 characters.

- You may only edit text fields in the title block symbol and the PIN_NAME text field inside a symbol boundary. Use the **TE** command.

- Refer to Table 5-1 in the ***User Guide*** for a list of legal characters for pin names and node names.

- For information on properly assigning a signal name to a wire refer to the description of the Text Menu.

☞ When positioning or entering text fields, you should be in full scale mode.

- Lines connected to VCC and GND symbols are assigned the VCC and GND signal name. Any other names assigned to such lines are ignored.

- Bus lines are traced in ADF generation only if they are connected to symbols with bus line stubs.

- Wire segments that are assigned the same signal names may not actually be connected in the drawing. However, the ADF treats them as one wire.

- Symbol reference numbers (displayed with the **SN** command) are shown in the ADF as comments enclosed in % symbols. These numbers are used for reference in error messages. Deleting a symbol causes a gap in the numbering sequence. This gap is closed, i.e., the database is repacked, when a drawing is saved or just before the ADF is generated.

☞ Any drawing file may be loaded as an area and vice versa.

Commands are listed in the following order:

ADF — Delete — Files — Help — Load— Size — Write

DA (Drawing ADF)

Syntax: **DA <Enter>**

Function: Generates an Altera Design File from the drawing in memory.

Description: Type **DA**. You are prompted with:

Press <Enter> to generate Altera Design File

When you press **<Enter>**, the message line displays:

Tracing nodes

There will be a delay while LogiCaps is tracing the wires and pin names. Then the ADF is printed to the screen while it is output to a file with the drawing file name and a .ADF extension.

The following features are included in the ADF generated:

(1) The header of the ADF is generated from the title block entered into the drawing with the TITLE primitive.

(2) The last entry generated in the ADF is END$.

(3) Symbol reference numbers are included in the ADF as comments, i.e., they are enclosed in % symbols.

(4) Unconnected lines and text fields have no effect on the ADF.

(5) The fact that two lines meet at a T intersection or cross over each other does not constitute an electrical or logical connection: a connection dot must be entered to create a connection. Refer to the **LJ** Command.

(6) Text fields within an EQN1 and EQN8 will be put into the EQUATIONS: section of the ADF.

To generate a valid ADF, you must adhere to the LogiCaps and A+PLUS design regulations listed under *Design Guidelines*.

This command may be executed at any time. LogiCaps will show the ADF for the currently displayed drawing. Press any key to return to the normal screen display.

Note: Depending on the size and complexity of your drawing, there may be a long delay while the nodes are traced for the ADF.

DD (Drawing Delete)

Syntax: **DD <Enter>**

Function: Deletes the drawing and clears the screen.

Description: Type **DD**. You are prompted with:

Press <Enter> to delete data from memory

Deletes all items from the database. If data have been edited since the last time the database has been loaded, you are prompted with:

OK to delete current drawing Y[N]?

Enter **Y** (Yes) if you want to clear the screen without saving the current database. If not, enter **N** (No) and the command is cancelled.

DF (Drawing Files)

Syntax: **DF <Enter>**

Function: Displays a list of drawing file names.

Description: Type **DF**. You are prompted with:

Enter path [current], then press <Enter>

A list of filenames with the extension .SD is displayed. If more files exist than can be shown on one screen, you are prompted with:

Press <Space> for more...

Press any other key to redraw the display and return to Main menu.

Note: You may type a path or drive name before pressing **<Enter>**. If you don't specify a path or drive name, the current drive or path will be used.

DH (Drawing Help)

Syntax: **DH <Enter>**

Function: Displays helpful information about all drawing commands.

Description: Type **DH**. You are prompted with:

Press <Enter> for info about DRAWING commands

The Drawing Help file describes all the Drawing commands. You may call it up at any time, then press any character to return to your circuit display.

If the file contains more text than can be displayed on the screen, you are prompted with:

Press <Space> for more...

More help information will be displayed.

DL (Drawing Load)

Syntax: **DL [filename] <Enter>**

Function: Loads the specified drawing file to the database.

Description: Type **DL**. You are prompted with:

Enter filename[.SD] (? for directory), then press <Enter>

The filename is optional. If no filename is specified, the file with the filename currently shown in the status line is loaded.

The default extension .SD is appended to the name. All extensions are ignored and overridden by .SD.

The filename may be preceded by a valid drive and/or path specification. If none is specified, the current path/drive is used.

Note: Any drawing currently in the database is deleted before the new file is loaded. If the data have been edited since the file has been loaded or saved, you are prompted with:

OK to delete current drawing Y[N]?

Enter **Y** (Yes) if you want to clear the screen without saving the current database. If not, enter **N** (No) and the command is cancelled.

DS (Drawing Size)

Syntax: **DS <drawing size> <Enter>**

Function: Selects the size of the drawing. Available sizes are A, B, C, D, E, AV, BV, CV, DV, EV, and F. ("V" stands for vertical orientation.)

Description: Type **DS**. You are prompted with:

Enter size A,B,C,D,E,AV,BV,CV, DV,EV or F, then press <Enter>

This command sets the limits for the drawing size. The cursor is not able to move beyond these limits. The selected size is displayed in the status line. (The default size is B.)

The drawing size is used to determine the scaling used for maximum zoom levels. The following dimensions of drawing sizes in inches and database units (DBUs) are available:

Size A	8.5 x 11 in.	85 x 110 DBU
Size AV	11 x 8.5 in.	110 x 85 DBU
Size B	17 x 11 in.	170 x 110 DBU
Size BV	11 x 17 in.	110 x 170 DBU
Size C	22 x 17 in.	220 x 170 DBU
Size CV	17 x 22 in.	170 x 220 DBU
Size D	34 x 22 in.	340 x 220 DBU
Size DV	22 x 34 in.	220 x 340 DBU
Size E	44 x 34 in.	440 x 340 DBU
Size EV	34 x 44 in.	340 x 440 DBU
Size F	90 x 90 in.	900 x 900 DBU

A drawing size X may appear in the status line if you load a drawing that had been saved with the **AW** (Area Write) command.

DW (Drawing Write)

Syntax: **DW [filename] <Enter>**

Function: Saves the database as a drawing file.

Description: Type **DW**. You are prompted with:

Enter filename[.SD], then press <Enter>

The contents of the database are stored in a drawing file specified with the filename. If you do not provide a filename, the filename currently displayed in the status line is used.

The default extension .SD is appended to the name. All extensions are ignored and overridden by .SD.

The filename may be preceded by a valid drive and/or path specification. If none is specified, the current path/drive is used.

The Main Help Command

The main Help command describes all the functions assigned to the function keys. It also explains cursor movement and arrow key movements as well as the Main Menu commands.

H (Help)

Syntax: **H <Enter>**

Function: Displays information about the functions assigned to all function keys.

Description: Type **H**. You are prompted with:

Press **<Enter>** for some useful information

You may call up the Help file at any time, then press any character to return to your circuit display.

Press **<Space>** to scroll the screen and display additional information

The Line Menu

The Line menu commands enable you to enter, move, copy, delete, bend, rubberband, and connect lines. You also may select dotted, dashed, and solid lines as well as bus lines.

You should adhere to the following rules so that LogiCaps can generate a valid Altera Design File (ADF) from your schematic.

- When connecting a wire line to a symbol pin stub, draw the line so that it touches (or crosses) the symbol boundary. Only then does the ADF consider the line to be connected to the symbol.

- To assign a signal name to a wire, you must place the graphics cursor right on the wire line and go into **TE** (Text Enter) mode; the graphics cursor is replaced by the text cursor, positioned at the lower left hand corner of the text field and just slightly above the line; you then type the desired text. (See also the description of the Text Menu.)

  To verify the correct text position, execute the **TB** (Text Borders) command to display the text field boundary. The touching boundary line must be located right on top of the wire line.

☞ When positioning text fields, you should be in full scale mode.

- Bus lines are traced in ADF generation only if they are connected to symbols with bus line stubs.

- Wire segments that are assigned the same signal names may not actually be connected in the drawing. However, the ADF treats them as one wire.

Commands are listed in the following order:

Bend — Copy — Delete — Enter — Help — Join — Move — Rubberband — Select

LB (Line Bend)

Syntax: **LB <Enter>**

Function: Toggles the routing for entering new lines from horizontal/vertical to vertical/horizontal and back.

Description: Enter the **LE** command.

With the arrow keys or mouse move horizontally and then vertically (or vertically and then horizontally). The lines are dashed indicating that the **LE** command has been activated but not yet executed.

Type **LB**. You are prompted with:

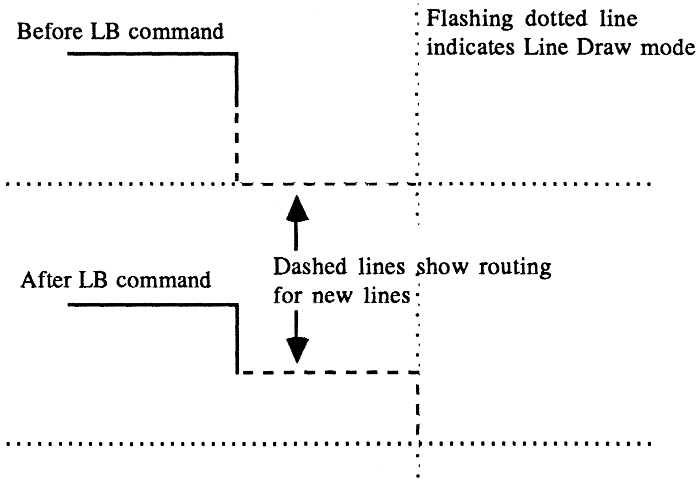
Press <Enter> to change line drawing route

The routing of the dashed lines changes to the opposite direction. This is a toggle function.

Note: If you are not in Line Draw mode, you are prompted with:

Must be drawing lines to use this command

The following figure illustrates the effect of the **LB** command.



Mouse Command

1. Press <**Left**> to go into Line Draw mode.
2. Move the mouse diagonally in any direction.
3. Press <**Center**> to switch the routing direction.
4. Press <**Left**> three times to execute the **LE** command and redisplay the graphics cursor.

LC (Line Copy)

Syntax: **LC <Enter>**

Function: Copies a line to a new location.

Description: Place the cursor on the line to be copied. Type **LC**. You are prompted with:

Press <Enter> to copy line at cursor

Line Copy mode is indicated by a dashed flashing line. Move the cursor to the new location and press **<Enter>** again to copy the line to the new location.

Note: As long as Line Copy is displayed on the command line, you can make multiple copies of the line by moving the cursor and pressing **<Enter>** repeatedly.

Mouse Command

1. Place the cursor on the line to be copied.
2. Enter the **LC** command on the command line. Press **<Enter>** and move the mouse to the new location.
3. Press **<Center>** and **<Right>** to execute the Line Copy command and redisplay the graphics cursor.

LD (Line Delete)

Syntax:	**LD <Enter>**
Function:	Deletes a specified line.
Description:	Place the cursor on the line to be deleted. Type **LD**. You are prompted with: **Press <Enter> to delete line at cursor**
Note:	As long as Line Delete is displayed on the command line, you can delete multiple lines by moving the cursor to additional lines and pressing **<Enter>**. If the cursor is not positioned on a line, the message line reads: **No line found at cursor**

Mouse Command

1. Place the cursor on the line to be deleted.
2. Press **<Right>** to delete the line.

LE (Line Enter)

Syntax: **LE <Enter>**

Function: Draws a line in the direction of the cursor movement.

Description: Type **LE**. You are prompted with:

Press <Enter> to begin drawing lines

The **LE** command activates Line Draw mode. Then you move the cursor horizontally or vertically and press **<Enter>** twice to draw the line and exit Line Draw mode.

Mouse Command

1. Press **<Left>** to enter Line Draw mode.
2. Move the mouse horizontally, vertically, or diagonally (for an orthogonal line).
3. Press **<Left>** twice (three times for an orthogonal line) to execute the **LE** command and redisplay the graphics cursor.

LH (Line Help)

Syntax: **LH <Enter>**

Function: Displays helpful information regarding the Line commands.

Description: Type **LH**. You are prompted with:

Press <Enter> for info about LINE commands

The Line Help file describes all the Line commands. You may call it up at any time, then press any character to return to your circuit display.

If the file contains more text than can be displayed on the screen, you are prompted with:

Press <Space> for more...

The screen scrolls to display more help information.

LJ (Line Join)

Syntax: **LJ <Enter>**

Function: Creates a connection dot at a specified line location. This is a toggle function.

Description: Place the cursor on the line at the location where you wish to enter a connection dot. Type **LJ**. You are prompted with:

Press <Enter> to toggle an interconnection dot

The connection dot is displayed. Press **<Enter>** again to remove it from the database.

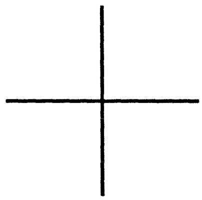
If the connection dot is entered on the line rather than at either end of it, the line is broken into two segments meeting at the new connection dot location. The two segments are then treated as two independent lines that can be deleted, moved, or copied separately.

When a connection dot is removed where two vertical or two horizontal line segments meet at the connection dot, they are merged into one horizontal or one vertical line segment.

Note: The command has no effect if the cursor is placed on a temporary line or on a location that is not a line.

The fact that two lines meet at a T intersection or cross over each other does not constitute an electrical or logical connection: a connection dot must be entered to create a connection. See the following illustration:

Lines not joined:



Lines joined:

Mouse Command

1. Place the cursor at the location where you wish to enter a connection dot.
2. Press <**Center**> to enter the connection dot.
3. Press <**Center**> again to delete the connection dot.

LM (Line Move)

Syntax: **LM <Enter>**

Function: Moves a line from one location to another.

Description: Place the cursor on the line to be moved and type **LM**. You are prompted with:

Press <Enter> to move line at cursor

The cursor takes the form of a flashing dotted image of the line being moved. Press the appropriate arrow key to move the line to the desired location. Press **<Enter>** to execute the move and return to the graphics cursor.

Any lines connected to the end points of the moving line are rubberbanded.

Mouse Command

1. Place the cursor on the line to be moved (not on one of the ends).
2. Press **<Left>** twice to display a dashed line cursor and move the mouse to the new location.
3. Press **<Left>** to execute the move and redisplay the graphics cursor.

LR (Line Rubberband)

Syntax: **LR <Enter>**

Function: Enables and disables the rubberbanding function used when you move symbols, lines, and areas. (Enable is the default.) This is a toggle function.

Description: Type **LR**. You are prompted with:

Press **<Enter>** to disable rubberbanding

When you press **<Enter>**, the message line reads:

Rubberbanding Disabled

When rubberbanding is enabled, line connections for lines ending on symbol boundaries or passing through area boundaries are retained.

Note: Temporary, high-density dotted lines may be automatically inserted to show connectivity where the movement of a symbol, line, or area has resulted in conflicts. You should remove these temporary lines with the **LD** (Line Delete) command and replace them with solid lines before you save the drawing or generate an ADF.

LS (Line Select)

Syntax: **LS <Enter>**

Function: Selects the line type for subsequent line drawing. The current line type is displayed on the status line in the LINE: field.

Description: Type **LS**. You are prompted with:

Press <Enter> until desired line type is displayed

The message line parallels the information shown in the LINE: field in the status line. For example, if the LINE: field shows a dashed line, the message line reads:

Broken lines

The following line types are available:

Solid line:

Dotted line:

High density dotted line:

Bus line:

Dashed line:

<F3> always selects the line type available next as you cycle through the line types. In contrast, when you execute the **LS** command, the selection always starts with the solid line type, regardless of which type is currently displayed in the status line.

When you record a macro, be sure to select a line type with the **LS** command not the **<F3>** key.

<F3> Cycles through the available line types. The current type is displayed in the LINE: field in the status line.

The Macro Menu

A macro is a recorded file (with the default extension .MAC) containing a keystroke and command sequence. This file is stored in a directory of macro files and may be played back at any time.

You may start recording a macro at any time during design entry. Then every keystroke and command entered is recorded, including the **MS** (Macro Stop) command .

Macros are especially useful for repetitive tasks such as creating the reference index for a specific drawing size, filling in title block information, and creating memory arrays.

When a macro is recorded, the sequence of changes entered into the status line is also saved; however, the settings themselves are not saved. Therefore, before you play back a macro, be sure to set the parameters in the status line so that they match those you used while recording the macro.

It is recommended that you use command line commands rather than function key commands while you record a macro to avoid any conflicts with function keys that have been overridden.

Commands are listed in the following order:

Assign — Clear — Delete — Help — Play — Record — Stop

MA (Macro Assign)

Syntax: **MA <filename>[.MAC] <Enter>**

Function: Assigns a function key, **<F1>** to **<F10>**, to a recorded macro so that the macro can be executed every time the function key is pressed. A maximum of 10 macro files can be assigned to the function keys.

Description: Type **MA**. You are prompted with:

Enter filename[.MAC] (? for directory), then press <Enter>

Type the name of the recorded macro file (you don't need to type the extension) and press **<Enter>**; you are prompted with:

Press <F> key to be assigned...

Press one of the function keys **<F1>** to **<F10>**. The macro is assigned to that function key—overriding the original function assigned to that key—so that everytime you press that function key, the macro is played back

Note: The **MC** (Macro Clear) command is used to return a function key to its originally assigned function.

Function Key Assignment

Any one of the function keys **<F1>** to **<F10>** may be assigned to a previously recorded macro file.

MC (Macro Clear)

Syntax: **MC <Enter>**

Function: Clears a function key from the assigned macro function and returns it to its originally assigned function.

Description: Type **MC**. You are prompted with:

Press <F> key to clear assignment

Press the key to which you had assigned a macro and from which you now want to remove that assignment.

Note: This command only deletes the assignment to a function key, not the macro file.

Function Key Assignment

Any one of the function keys, **<F1>** to **<F10>**, may be cleared from a previously assigned macro and returned to its originally assigned function.

MF (Macro Files)

Syntax: **MF <Enter>**

Function: Displays a list of macro file names.

Description: Type **MF**. You are prompted with:

Enter path [current], then press <Enter>

A list of filenames with the extension .MAC is displayed. If more files exist than can be shown on one screen, you are prompted with:

Press <Space> for more ...

Press any other key to redraw the display and return to Main menu.

Note: You may type a path or drive name before pressing **<Enter>**. If you do not specify a path or drive name, the current drive or path will be used.

MH (Macro Help)

Syntax: **MH <Enter>**

Function: Displays helpful information about the commands available from the Macro menu.

Description: Type **MH**. You are prompted with:

Press <Enter> for info about MACRO commands

The Macro Help file describes all the Macro commands. You may call it up at any time, then press any character to return to your circuit display.

If the file contains more text than can be displayed on the screen, you are prompted with:

Press <Space> for more...

The screen scrolls to display more helpful information.

MP (Macro Play)

Syntax: **MP <filename>[.MAC] <Enter>**

Function: Plays back a previously recorded macro.

Description: Type **MP**. You are prompted with:

Enter filename[.MAC] (? for directory), then press <Enter>

The macro with the specified name is played back step by step exactly the way it was entered.

Function Key Assignment

If you have a function key assigned to the macro, you simply press it to play back the macro.

MR (Macro Record)

Syntax: **MR <filename>[.MAC] <Enter>**

Function: Begins recording a macro with the specified filename and continues recording until you enter the **MS** (Macro Stop) command or press **<Shift><F10>**.

Description: Type **MR**. You are prompted with:

Enter filename[.MAC], then press <Enter>

The FILE: field in the status line changes to RECORDING MACRO. Every keystroke and mouse movement is now recorded to the specified macro file until you enter the **MS** command.

Note:

(1) Do not use any of the function keys while recording a macro.

(2) Macros cannot be nested. Although one macro can call another macro, control cannot return to the original macro.

(3) Remember that mouse movements depend on the zoom level. Unlike keystrokes, they are not consistent from zoom level to zoom level. Therefore, when you change zoom levels during macro recording, it is best to use only keystroke commands.

MS (Macro Stop)

Syntax: **MS <Enter>**

Function: Stops the recording of a macro.

Description: Type **MS**. You are prompted with:

Press **<Enter>** to terminate macro recording

Note: If you enter the **MS** command while not recording a macro, you are prompted with:

Command only valid for recording macros

Function Key Assignment

<Shift><F10> Stops the recording of a macro. This function key assignment cannot be overridden by a macro assignment.

Note that if you use **<Shift><F10>** to stop a macro, the macro you create does not disrupt the command currently on the command line.

The Quit Command

If you execute this command, you exit LogiCaps. If you have made any changes to the loaded drawing, you are prompted to indicate whether you wish to save these changes.

Q (Quit)

Syntax: **Q <Enter>**

Function: Exits LogiCaps.

Description: Type **Q**. You are prompted with:

Press <Enter> to quit LogiCaps

(1) If you have made no changes to the loaded drawing, LogiCaps exits.

(2) If you have made changes to the loaded drawing, you are prompted with:

Save changes Y, N, or <Esc>?

If you type **Y**, you are prompted with:

Enter filename[.SD], then press <Enter>

Then type the desired filename.

If you type **N**, LogiCaps exits.

Press **<Esc>** to returned to the drawing.

The Symbol Menu

With the Symbol menu you enter, move, copy, rotate, and delete symbols. You also search your drawing for a specific symbol and identify a symbol by a reference number.

To ensure that LogiCaps can generate a valid Altera Design File (ADF), you must adhere to the following design guidelines when entering symbols into your drawing:

- You may only edit the text fields in the title block symbol and the PIN_NAME text field inside a symbol boundary.
- A pin name text field should not be deleted.
- Pin names may be up to eight characters long and may contain any character except '%', ',', '=', '@', '(', and ')'.
- A pin number is assigned to a pin name or buried register output by entering an at-symbol (@) and the specific number after the name. Name, @, and number together may not exceed 11 characters.
- Symbol reference numbers (displayed with the **SN** command) are shown in the ADF as comments enclosed in % symbols. These numbers are also used for reference in error messages. Deleting any symbols causes a gap in the numbering sequence. This gap is closed and the database repacked, when the drawing is saved or before the ADF is generated.

Commands are listed in the following order:

Copy — Delete — Enter — Find — Help — Library — Move — Numbers — Reflect.

SC (Symbol Copy)

Syntax: **SC <Enter>**

Function: Copies a symbol to another location.

Description: Place the graphics cursor inside the symbol or on the symbol boundary. Type SC. You are prompted with:

Press <Enter> to copy symbol at cursor

After you press **<Enter>**, the graphics cursor is replaced with a flashing, dotted symbol boundary box as well as flashing dotted pin stubs, and you are prompted with:

Symbol Copy Mode - <Center> to copy, <Right> to terminate

With the arrow keys, move the box cursor to the desired location. The flashing dotted pin stubs enable you to easily line up the pin stubs with any wires, as necessary. Press **<Enter>** again. The symbol will be copied at the new location and Symbol Copy mode is terminated.

Any alphanumeric fields within the original symbol boundary are also copied to the new location.

Note: A symbol cannot be copied into a location where it overlaps another symbol.

While in Symbol Copy mode, the X Y field in the status line indicates the position of the symbol's upper left hand corner, although the cursor is located inside the symbol wherever it was before Symbol Copy mode was invoked.

Mouse Command

1. Place the cursor inside the symbol to be copied.
2. Press **<Center>**and move the mouse to the new location.
3. Press **<Center>** to copy the symbol and remain in Symbol Copy mode. Press **<Right>** and exit Symbol Copy mode.

SD (Symbol Delete)

Syntax: **SD <Enter>**

Function: Deletes a previously entered symbol from the database.

Description: Place the cursor inside the symbol boundaries or on the symbol boundary line. Type **SD**. You are prompted to:

Press <Enter> to delete symbol at cursor

Note: If the cursor is not inside or on the symbol boundary, the message line reads:

No symbol found at cursor

SE (Symbol Enter)

Syntax: **SE <symbol name> <Enter>**

Function: Enters the specified symbol with its upper left hand corner at the current cursor location.

Description: Type **SE**. You are prompted to:

Enter symbol name (? for directory), then press <Enter>

The symbols available with LogiCaps are those listed in the current symbol library. You can access this list with the **SL** command or by pressing **?**.

Note: Symbols may not be entered at locations where they would overlap other existing symbols.

Function Key Assignment

<F8> Initiates Symbol Enter mode.

SF (Symbol Find)

Syntax: **SF <reference number> <Enter>**

Function: Places the cursor on the symbol with the specified reference number.

Description: Type **SF**. You are prompted with:

Enter symbol reference number, then press <Enter>

Type the desired number and press **<Enter>**. The cursor is positioned at the upper left corner of the specified symbol.

Every symbol is assigned a reference number as it is entered into the database. This number is located in the upper left hand corner inside the symbol boundary. The display is enabled or disabled with the **SN** command. When you delete a symbol and then save the file, or when you move/copy an area, the symbols are renumbered (i.e., the database is packed), so that there are no gaps in the numbering sequence.

Before you create the Altera Design File (ADF) with the **DA** command, the database is packed; the ADF will then number the symbols according to the packed database. If you subsequently edit the drawing, you should regenerate the ADF to reflect the changes.

Note: Error messages issued by A+PLUS refer to these numbers so that you may quickly locate an error. The error message displays the line from the ADF and the reference number is included as a comment (e.g., % SYM 25 %) in the ADF. Therefore you can associate a given net in the ADF with a symbol in the drawing.

If you specify a reference number that does not exist in the database, the cursor does not move.

SH (Symbol Help)

Syntax: **SH <Enter>**

Function: Displays helpful information about the commands available in the Symbol menu.

Description: Type **SH**. You are prompted to:

Press <Enter> for info about SYMBOL commands

The Symbol Help file describes all the symbol commands. You may call it up at any time, then press any character to return to your circuit display.

If the file contains more text than can be displayed on the screen, you are prompted to:

Press <Space> for more...

The screen scrolls to display more information.

SL (Symbol Library)

Syntax: **SL <symbol library name> <Enter>**

Function: Clears the current screen and displays the list of available symbols in the specified symbol library. The specified library becomes the currently active symbol library.

Description: Type **SL**. The name of the default symbol library ALTERA.SYM is displayed.

Press **<Enter>** for a list of available symbols in that library.

Type **?** for a list of available symbol libraries. If you want to see a list of available symbols in a library other than the default symbol library, simply type the name of the desired library after the **SL** command.

Press any key to return to the normal screen.

Note: You may also type **SE ?** or **<F8> ?** to display the symbol directory for the current symbol library.

SM (Symbol Move)

Syntax: **SM <Enter>**

Function: Moves a symbol from its current location to a new specified location.

Description: Place the graphics cursor on or inside the symbol boundary line. Type **SM**. You are prompted to:

Press <Enter> to move symbol at cursor

Symbol Move mode is initiated, and the graphics cursor is replaced with a flashing, dotted symbol boundary box as well as flashing dotted pin stubs. With the arrow keys move the box cursor to the desired location. The dotted pin stubs enable you to easily line up the pin stubs with any wires, as necessary. Press **<Enter>** to execute the symbol move, exit Symbol Move mode, and return to the normal graphics cursor.

If rubberbanding is enabled during the symbol move, any lines terminating on the symbol edge become dashed rubber lines and will maintain connection until the move is completed. Then they are converted back to solid lines.

Rubber lines that result in conflicting routing are turned into temporary lines displayed as high-density dotted lines. You should delete these temporary lines before you save the schematic.

Any alphanumeric fields within the original symbol are also moved to the new location.

Note: A symbol may not be moved to a location where it overlaps another symbol.

While in Symbol Move mode, the X Y field in the status line indicates the position of the symbol's upper left hand corner, although the cursor is located inside the symbol wherever it was before Symbol Move mode was invoked.

Mouse Command

1. Place the cursor inside the symbol or on the symbol boundary.
2. Press <Left> (twice if you are on the symbol boundary) to go into Symbol Move mode and move the cursor to the new location.
3. Press <Left> to execute the move and redisplay the graphics cursor.

SN (Symbol Numbers)

Syntax: **SN <Enter>**

Function: Enables or disables the display of symbol reference numbers, located just inside the upper left hand corner of the symbol boundary, as well as of symbol pin numbers. (This is a toggle function. When LogiCaps is invoked, SN is disabled.)

Description: Each symbol is assigned a reference number according to the order in which it was entered into the database. This reference number is used in the Altera Design File (ADF) and in the error messages to enable you to locate a specific symbol.

Type SN. You are prompted to:

Press <Enter> to display symbol reference and pin numbers

After you press <Enter>, the message line reads:

Symbol ref numbers enabled

If a symbol is deleted from the database, the reference number is also deleted, causing a gap in the numbering sequence. As soon as the drawing is saved or before the ADF is generated, these gaps are eliminated and the symbols are renumbered.

Note: If you edit the database after you have generated an ADF for a schematic, be sure to generate the ADF again so that it reflects the most current numbering sequence.

Symbol pin numbers are important parts of the symbol definition and should not be moved or deleted.

SR (Symbol Reflect)

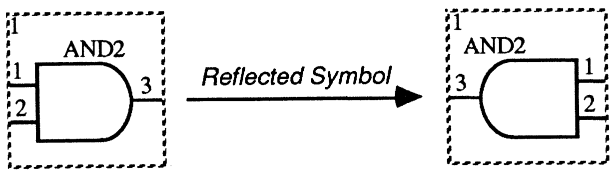
Syntax: **SR <Enter>**

Function: Flips the symbol horizontally so that it points into the opposite direction. Retains the coordinate location. This is a toggle function.

Description: Place the cursor on or inside the symbol boundary. Type **SR**. You are prompted to:

Press <Enter> to reflect symbol at cursor

After you press **<Enter>**, the symbol is flipped. All alphanumeric fields inside the symbol are moved to the appropriate locations to reflect the new direction of the symbol as shown in the illustration. (Note also the relative location of the reference and pin numbers.)



Note: If the cursor is not positioned on a symbol, the message line reads:

No symbol found at cursor

The Text Menu

Commands from the Text menu let you edit, copy, move, and delete alphanumeric characters and their associated text fields. You may also choose four different types of character sizes, including reverse video, and you may bring an entire text file from another document into your drawing and enter it at the cursor location. LogiCaps will honor the formatting features of the imported text such as columns and carriage returns. In addition, you may search for a specified text.

In command line editing or text entry mode the following keys have these functions:

<Del>	Deletes the character at the cursor and moves the text to the right of the cursor to the left.
<Ins>	Toggles between insert and exchange modes. In exchange mode it types right over the character at the cursor. In insert mode it creates a space and moves the text to the right of the cursor to the right.
<Backspace>	Deletes the character to the left of the cursor and moves the text to the right of the cursor to the left.
<←> and **<→>**	Positions the cursor without affecting the text.
<End>	Moves the cursor to the end of the text.
<Home>	Moves the cursor to the beginning of the text.
<Ctrl><←>	Moves the cursor to end of the previous word.
<Ctrl><→>	Moves the cursor to the beginning of the next word.
<Ctrl><End>	Moves the cursor to the right to the nearest pin number.

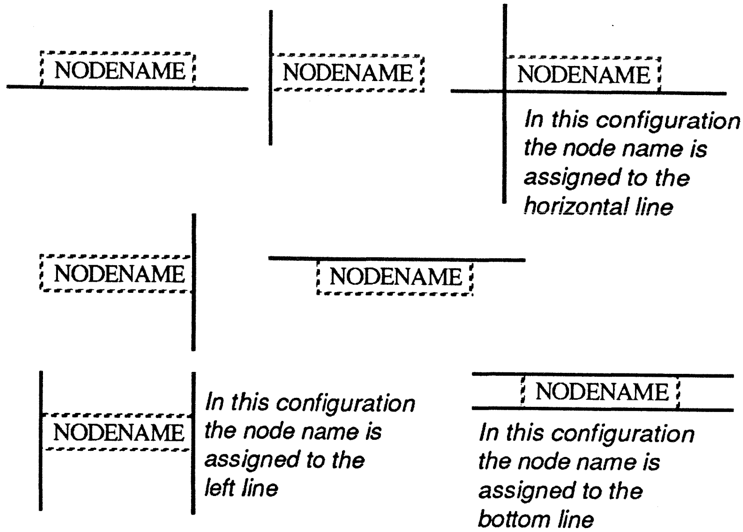
To ensure that LogiCaps can generate a valid Altera Design File (ADF), you must adhere to the following design guidelines when entering text into your drawing:

- Lines connected to VCC and GND symbols are assigned the VCC and GND signal name. Any other names assigned to such lines are ignored.
- You may only edit the text fields in the title block symbol and the PIN_NAME text field inside a symbol boundary. A pin name text field should not be moved or altered.
- A pin name text field should not be deleted.
- Pin names may be up to eight characters long and may contain any character except '%', ',', '=', '@', '(', and ')'.
- A pin number is assigned to a pin name or buried register output by entering an at-symbol (@) and the specific number after the name. Name, @, and number together may not exceed 11 characters.
- To assign a signal name to a wire, you must place the graphics cursor right on the wire line and go into **TE** (Text Enter) mode; the graphics cursor is replaced by the text cursor positioned at the lower left hand corner of the text field and just slightly above the line. You then type the desired text. If you wish to move the text field, you must move it to one of the legal text field positions shown in the following illustration.

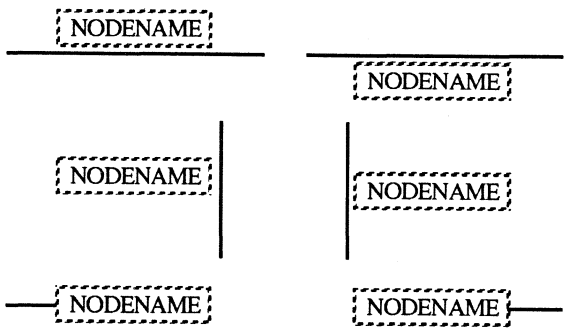
☞ When positioning and entering text fields, you should be in full scale mode.

To verify the correct position of text, execute the **TB** (Text Borders) command to display the text field boundary. The touching boundary line must be located right on top of the wire line. See the following illustrations:

Correct position of text field:



Incorrect position of text field:



Commands are listed in the following order:

Borders — Copy — Delete — Enter — Find — Help — Import — Move — Select

TB (Text Borders)

Syntax: **TB <Enter>**

Function: Toggles the display of the text field borders.

Description: Type **TB**. You are prompted to:

Press <Enter> to enable display of text borders

When LogiCaps is initially invoked, text field borders are disabled. After you enter the **TB** command, each alphanumeric text field in the displayed drawing shows a dotted box around it delineating the text field.

Whenever a command requires that the cursor be in a text field, it has to be within this box of the specified text.

Text field borders are important when you name wires. The border helps to precisely locate a signal field adjacent to a line and to ensure that the text is associated with the line when the ADF is generated.

TC (Text Copy)

Syntax: **TC <Enter>**

Function: Copies alphanumeric text contained within a text field to another location.

Description: Place the graphics cursor inside or on the edge of a text field and type TC. You are prompted to:

Press <Enter> to copy text at cursor

The graphics cursor is replaced with a flashing dotted box cursor indicating the boundaries of the text field. Move this cursor to the desired location and press **<Enter>** again to execute Text Copy.

Note: Alphanumeric fields may overlap other alphanumeric fields and objects in the drawing (i.e., no error message is issued).

If the cursor is not on a text field, the message line reads:

No text found at cursor

Mouse Command

1. Place the cursor inside the text field to be copied.
2. Press **<Center>** and move the mouse to the new location. (If you had placed the cursor on the text field border, you must press **<Center>** twice.)
3. Press **<Center>** to display the copied text at the new location. The flashing text boundary is still displayed, indicating that you can make multiple copies just by moving the mouse and pressing **<Center>** repeatedly.
4. Press **<Right>** to return to the graphics cursor.

TD (Text Delete)

Syntax: **TD <Enter>**

Function: Deletes the targeted text from the database.

Description: Place the cursor on the text field to be deleted. Type **TD**. You are prompted to:

Press <Enter> to delete text at cursor

The text and field boundary are deleted.

Note: If the cursor is not positioned on a text field, the message line reads:

No text found at cursor

☞ Do not delete text fields inside a symbol. They are needed generating the ADF.

TE (Text Enter)

Syntax: **TE <text> <Esc>**

Function: Creates a new text field or edits an existing one.

Description: Check the TXT field in the status line. It indicates the current text type. If you wish to change the type, press **<F4>** or use the **TS** (Text Select) command until the desired type is displayed.

Place the cursor at the desired location. Type **TE**. The graphics cursor is replaced with the flashing text cursor and you are prompted to:

Type text, then press <Esc> to quit

Type the desired text; then press **<Esc>** to return to the graphics cursor and exit Text Enter mode. Refer to the general description of the Text Menu for correct use of the **<Del>**, **<Ins>**, **<Backspace>**, **<←>**, and **<→>** keys.

If you want to enter multiple lines of text without exiting Text Enter mode, type the desired text then press **<Esc>**. Move the cursor to the new location and press **<Enter>**. The text cursor is redisplayed at the beginning of the new text field.

When the Text Borders function is enabled, the text field borders are displayed and you can see the dimensions of a specific text field. Refer to the general description of the Text Menu for correct positioning of text fields.

If the graphics cursor is not within or on the edge of an existing text field, the text cursor is positioned slightly above the original graphics cursor location and a new text field is created.

If the graphics cursor is within or on the edge of an existing text field, the text cursor is positioned under the first character of that field. You may then edit that field. However, you may not change the text type in an existing text field.

Note:

1. Text fields exist only if there is one or more characters in a field. If you delete all characters within a field, the field is erased from the database.
2. Character types available are: SMALL (caps only), MEDIUM, LARGE, and REVERSE.

Mouse Command

Text Enter mode may be terminated by pressing **<Right>**.

Function Key Assignment

<F7> This function key initiates Text Enter mode and replaces the graphics cursor with the text cursor.

TF (Text Find)

Syntax: **TF <text> <Enter>**

Function: Positions the graphics cursor at the lower left hand corner of the text field for the specified text.

Description: Type **TF**. You are prompted to:

Enter text sought (# matches any character), then press <Enter> ...

Type the text you wish to locate in the drawing and press **<Enter>**. If the specified text appears multiple times in the drawing, press **<Enter>** repeatedly until you locate the desired text field.

The # symbol is a wildcard character; it matches any alphanumeric character. For example, if you enter **#O#F** as your text to be searched, LogiCaps will look for all the I/O Primitives.

Note: When you enter the text to be searched, upper- and lowercase are insignificant.

TH (Text Help)

Syntax: **TH <Enter>**

Function: Displays helpful information about the commands available in the Text menu.

Description: Type **TH**. You are prompted with:

Press <Enter> for info about TEXT commands

The Text Help file describes all the Text commands. You may call it up at any time, then press any character to return to your circuit display.

If the file contains more text than can be displayed on the screen, you are prompted with:

Press <Space> for more...

The screen scrolls to display more help information.

TI (Text Import)

Syntax: **TI <filename> <Enter>**

Function: Transfers text from a text file to the current database starting at the current cursor location.

Description: Type **TI**. You are prompted to:

Enter filename (? for directory), then press <Enter>

The text being transferred is entered at the current cursor location and all formatting from the original text is maintained.

This function is useful when you want to enter documentation into a drawing. Text is entered as if it was typed with the **TE** (Text Enter) command.

Only printable characters are legal. Some control characters (e.g., **<Esc>**) will terminate this function.

TM (Text Move)

Syntax: **TM <Enter>**

Function: Moves a specified text field to a new location.

Description: Place the cursor on the border of or inside the text field to be moved. Type **TM**. You are prompted to:

Press <Enter> to move text at cursor

The text field border flashes. With the arrow keys move the field to the new location and press **<Enter>** to execute the command, exit Text Move mode, and return to the graphics cursor.

Note: LogiCaps does not check for text fields overlapping other text fields or objects in the drawing.

Mouse Command

1. Place the cursor inside the text field to be moved.
   Note: If the targeted text field is inside a symbol boundary, the mouse command will initiate Move mode for the entire symbol, so you must type **TM <Enter>** in order to be able to pick up the text field only. However, you should never move text fields inside a symbol, because they are needed to generate the ADF.
2. Press **<Left>** (if you placed the cursor on the border of the field press **<Left>** twice) and move the mouse to the new location.
3. Press **<Left>** to execute the move and redisplay the graphics cursor.

TS (Text Select)

Syntax: **TS <Enter>**

Function: Selects one of four available character types as displayed in the TXT: field in the status line. This is a toggle function.

Description: Type **TS**. You are prompted to:

Press <Enter> until desired text type is displayed

The message line parallels the information shown in the TXT: field in the status line. For example, if the TXT: field shows medium character size in reverse video, the message line reads Reverse Medium Text.

The following character types are available:

Small , caps only:	SIG
Medium:	Sig
Large:	Sig
Reverse, medium only:	Sig

Note: Already existing text fields may be edited in the same character type only. If you wish to change the type for an already existing field, you must delete the current field and re-enter the text with the new character type.

Function Key Assignment

<F4> Always selects the character type available next as you cycle through the four character types. In contrast, when you execute the **TS** (Text Select) command, the selection always starts with the medium size character type, regardless of which type is currently displayed in the status line.

When you record a macro, be sure to select a character type with the **TS** command not the **<F4>** key.

The Window Menu

The commands available from the Window menu enable you to move to a specified location in the window, save certain views and recall them, set a desired grid pattern, and split the window into halves.

For a general description of the windowing facility, zoom levels, window splitting, and grid specifications refer also to Introduction to LogiCaps.

Commands are listed in the following order:

Auto — Color — Grid — Help — Move— Pan — Recall — Split —Tag

If you are using the Hercules Graphics Card, you do do not have the Window Color (WC) option.

WA (Window Auto)

Syntax: **WA <Enter>**

Function: Enables and disables the automatic panning function. This is a toggle function.

Description: Type **WA**. You are prompted to:

Press <Enter> to disable Autopanning

When autopanning is enabled and you move the cursor past the edge of the current display window, the window is repositioned in the drawing and the cursor is recentered.

Autopanning is enabled as the default condition. Once you press **<Enter>**, the message line reads:

Autopanning Disabled

Press **<Enter>** to enable it again.

Mouse Command

You cannot enable or disable autopanning with the mouse. However, if it is enabled and you move the mouse over the edge of the window display, the display is homed and the cursor is repositioned in the center of the display. If it is disabled, the display is not rehomed.

WC (Window Color)

Syntax: **WC <Enter>**

Function: Select the color for the foreground and the background of the window display.

Description: LogiCaps supports three types of graphics cards:

1. IBM Color/Graphics Monitor Adapter (CGA)
2. IBM Enhanced Graphics Adapter (EGA)
3. Hercules Graphics Card

A description of each follows.

Color/Graphics Monitor Adapter

If you have the Color/Graphics Monitor Adapter (CGA), you are prompted to

Press <Enter> to change display color

After pressing **<Enter>**, the message line displays the message:

Current color value is n

where n is a number from 1 to 31. Now press **<Enter>** repeatedly until you get the color of your choice.

☞ One of the color choices may be black on black. If you press **<Enter>** and your screen goes dark, just keep pressing **<Enter>** to select a different color.

Type **S** to recreate the configuration file and save all color information.

☞ The Save (**S**) command will also save the mouse speed which you may have specified with the **<Shift><F1>** or **<Shift><F2>** keys.

To restore the default configuration after you have entered an **S** command, use the **-x** option when you invoke LogiCaps the next time. Refer to Invoking LogiCaps at the beginning of this *LogiCaps Commands* section for information on the **-x** option.

Enhanced Graphics Adapter (EGA)

If you have the Enhanced Graphics Adapter (EGA), the command entry line prompts you for:

Window Color Palette: <n> <Enter>

You now type one of the palette numbers 0 to15 listed below . Note that these numbers correspond to IBM palette number standards for the EGA with extended memory. If you don't have the extended memory, some of the features will share the same color.

0	Background
1	Frame, Status Labels, Menu, Command Echo
2	Not used
3	Status Data, Index Numbers, Grid, Directories, Help
4	Messages, Temporary Lines
5	Symbols, Symbol Names (Text), Documentation (Text), Solid Lines, Bus Lines, Dashed Lines, Connection Dots
6	Graphics Cursor, Text Cursor, Line-Draw Cursor, Targeted Symbol, Targeted Text, Targeted Area
7	Dotted Lines, High-Density Dotted Lines
8	Not used
9	Symbol Boundaries, Symbol Numbers
10	Not used
11	Pin Numbers, Area Boundaries
12	Rubber-Banded Lines, Line-Draw Routing
13	Signal Names (Text)
14	Area Boundary Definitions
15	I/O Pin Names (Text)

Next, you type the color number of your choice (64 colors are available), or you may press **<Space>** to scroll through the available colors or **<Esc>** to cancel the command.

Once you have entered a color number, the color for the selected palette will change.

Type **S** to recreate the configuration file and save all color information.

The Save (**S**) command will also save the mouse speed which you may have specified with the **<Shift><F1>** or **<Shift><F2>** keys.

To restore the default configuration after you have entered an **S** command, use the **-x** option when you invoke LogiCaps the next time. Refer to **Invoking LogiCaps** at the beginning of this section for information on the **-x** option.

Hercules Graphics Card

If you have the Hercules Graphics Card, you cannot invoke the **WC** command.

WG (Window Grid)

Syntax: **WG <X Y> <Enter>**

Function: Sets the grid spacing according to the specified X and Y numbers. (10,10) is the default.

Description: Type **WG**. You are prompted to:

Enter X,Y spacing or TOG, then press <Enter> ...

If you wish to set the grid lines 10 database units (DBUs) apart horizontally and vertically, then you only need to press **<Enter>** and the grid lines are displayed with the specified spacing.

If you wish to space the grid lines differently, you type in the desired numbers and press **<Enter>**. The first number entered determines spacing for the horizontal lines, the second number determines spacing for the vertical lines. The numbers may be separated by either a space or a comma (,).

The grid start point is partially determined by the cursor location: it is set so that the current cursor position will be at a grid line intersection when the grid is displayed to facilitate alignment with drawing features.

If you type **TOG**—or any other non-numeric character—the grid spacing and positioning remains as it is, but the display of the grid is toggled to OFF or ON.

Note: When LogiCaps is invoked initially, grid spacing is set to 10,10 with no offset.

Function Key Assignment

<F9> Displays or hides the current grid.

WH (Window Help)

Syntax: **WH <Enter>**

Function: Displays helpful information about the commands available in the Window menu.

Description: Type **WH**. You are prompted to:

Press <Enter> for info about WINDOW commands

The Window Help file describes all the Window commands. You may call it up at any time, then press any character to return to your circuit display.

If the file contains more text than can be displayed on the screen, you are prompted to:

Press <Space> for more...

The screen scrolls to display more help information.

WM (Window Move)

Syntax: **WM <X Y> <Enter>**

Function: Moves the cursor to the specified coordinate location.

Description: Type **WM**. You are prompted to:

Enter X,Y coordinates, then press <Enter>

Type the desired coordinates—separated by a space or a comma—and press **<Enter>**. The cursor is moved to the specified database location, the display is refreshed with the cursor approximately in the center of the current display window.

WP (Window Pan)

Syntax: **WP <Enter>**

Function: Puts the cursor keys in PAN mode. Each time you press a cursor key, the distance moved is equal to the width (**<←>** and **<→>**) or height (**<↑>** and **<↓>**) of the current display window, unless it is near the edge of the drawing. This is a toggle function.

Description: Type **WP**. You are prompted to:

Press <Enter> to set arrow keys for PAN

Cursor movement is relative to the drawing, not to the screen.

To exit PAN mode press **1** (default) or type a desired number to set the CRSR: field in the status line.

WR (Window Recall)

Syntax: **WR <F1> [<F2>] <Enter>**

Function: Recalls a view previously saved with the **WT** command.

Description: The **WR** command is used together with the **WT** command which saves a specified window view. Type **WR.** You are prompted to:

Press <F1> or <F2> to Recall view saved for that key

Press one of the two function keys. The message line reads:

View recalled by <F1> [<F2>]

The window view that was saved with the **WT** command may be recalled by pressing the assigned function key.

WR and **WT**, used together, allow you to save two views at one time, so that you may be able to jump back and forth between these two views.

Note: A macro assigned to one of these function keys overrides the Window Tag or Window Recall function.

Function Key Assignment

<F1>/<F2> **<F1>** saves (1) the cursor position in the drawing, (2) the window location in the drawing, and (3) the zoom level in effect at the time the **WT** command was executed.

<F2> is assigned to perform the same function to save a second partial view of the design. With these two functions keys, you can quickly jump back and forth between two views of the schematic.

WS (Window Split)

Syntax: **WS <Enter>**

Function: Splits the display window into two separate view windows in the database. This is a toggle function.

Description: Type **WS**. You are prompted to:

Press <Enter> to split display

Each window has its own reference index and is independent of the other window.

The window that happens to contain the cursor when the command is entered, is the "active" window, indicated by the fact that its boundary is a solid line. The other window has a dashed boundary, indicating that it is currently inactive. Toggling **<Tab>** "activates" the other window and moves the cursor into it.

Pressing the "greater than" key, **<Shift><'<'>**, enables you to move the vertical split line to the left; pressing the "smaller than " key, **<Shift><'>'>**, enables you to move the vertical split line to the right; thus, you may adjust the size of both the active and the inactive windows.

Changes may be entered into the active window as though it were a full window display. You may go into different zoom levels; enter, delete, copy, or move lines, symbols, and text. All changes to the drawing are reflected in the inactive window.

The cursor is always in the same drawing location in both halves; therefore, you could make changes in one half that's at full screen zoom level, and verify these changes by looking at the other half that's zoomed out to full drawing size.

<F10> Toggles between full and split window display.

WT (Window Tag)

Syntax: **WT <F1> [<F2>] <Enter>**

Function: Saves a specified window view for recall by either **<F1>** or **<F2>**. This view is then assigned to one of these two function keys so that it may be recalled every time you press the assigned key.

Description: The **WT** command is used together with the **WR** command or the **<F1>** or **<F2>** keys. Type **WT**. You are prompted to:

Press <F1> or <F2> to save view for that key

Then you press one of the two function keys. The message line reads:

View saved for <F1> [<F2>]

The window view is assigned to the function key you pressed.

<F1> and **<F2>** allow you to save two views at one time, so you are able to jump back and forth between these two views.

Note: A macro assigned to one of these function keys overrides the recall function, but does not override the function when it is invoked with Window Tag or Window Recall from the command line. Even though a macro may be assigned to **<F1>** and/or **<F2>**, you may tag a view associated with one of these keys by using the **WT** command, and you may recall the view with the **WR** command.

SECTION 8

Standard MacroFunctions

This section contain a detailed description of Altera's 30 Standard MacroFunctions. For detailed information on how to use MacroFunctions in your design, refer to Section 6 (*MacroFunction Tutorial*) in the ***User Guide***.

Altera's MacroFunctions are a collection of high-level building blocks. The 30 Standard MacroFunctions are the most commonly used SSI and MSI functions. MacroFunctions may be used together with the gate and flipflop primitives available in the Altera Primitive Library .

The Altera Design Processor (ADP) ensures that use of MacroFunctions does not cause any loss of design efficiency. It analyzes the complete logic circuit and automatically removes unused gates and flipflops from a MacroFunction used in a design. This MacroMuncher feature allows you to freely employ logic blocks in your design without having to worry about optimizing their use. All inputs to MacroFunctions are defined with default input signal levels so that unused inputs can simply be left unconnected.

The 30 Standard MacroFunctions also include a few EPLD-specific MacroFunction blocks that are particularly efficient in optimizing logic functions for EPLD implementation.

☞ The logic of some MacroFunctions differs slightly from the logic of the TTL equivalents. Be sure to always check the function tables provided with each MacroFunction description to see whether a particular MacroFunction meets your requirements.

MacroFunction Description

Each MacroFunction description consists of the following parts:

- The name assigned to the MacroFunction (an Altera-specific name or the standard TTL number)
- A symbol representation
- A description including syntax, available EPLDs, and default signal levels
- A function table
- A logic schematic showing the logic contained in the MacroFunction block

Figure 8-1 shows a standard MacroFunction symbol representation:

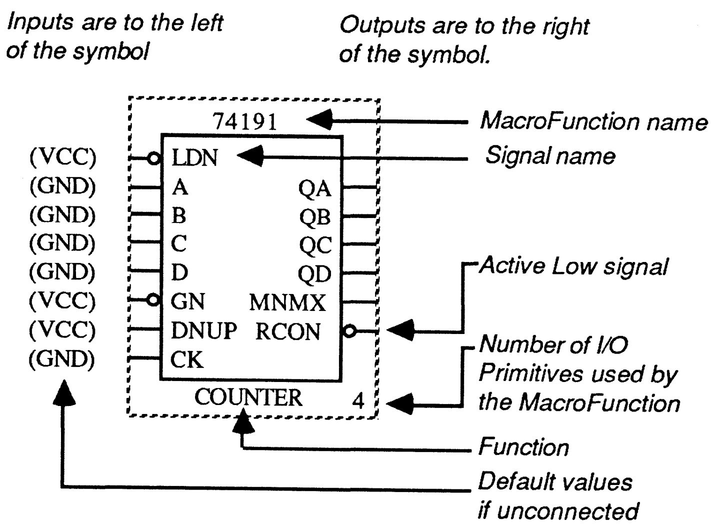


Figure 8-1. MacroFunction Symbol Representation

☞ All MacroFunction clocks are positive-edge triggered.

The symbol representation is followed by a detailed description as outlined here:

Declaration:	Specifies the syntax for the MacroFunction.
EPLDs:	Specifies for which Altera EPLDs the MacroFunction may be used.
Default Signal Levels:	Indicates the default signal level for each input pin if no connection has been made. If no default is specified, it is DON'T CARE.

Next, the function table shows a table detailing the relation between the individual inputs and outputs.

Finally, the logic schematic shows a schematic of the logic within the MacroFunction.

Categories of MacroFunctions

Altera MacroFunctions are listed in numerical sequence, with Altera-specific MacroFunctions listed first. All other designations correspond to those of the 74 Series of TTL circuits.

You can quickly locate a particular MacroFunction in the table of contents. However, if you need assistance in choosing a particular MacroFunction for a specific application, you may consult Table 8-1, which lists the 30 Standard MacroFunctions according to their functions.

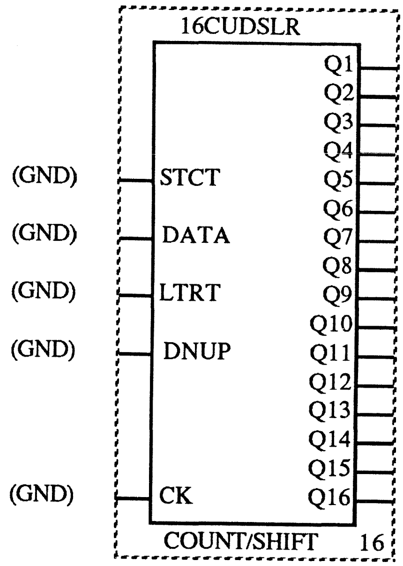
Table 8-1. MacroFunction Categories

FUNCTION	COMPONENT
Adder	7483
Comparator	8MCOMP, 7485
Counter	16CUDSLR,4COUNT, UNICNT2, 7493, 74160, 74161, 74162, 74163, 74190, 74191, 74393
Latch	74279, 74373
Multiplexer/Decoder	7442, 7449, 74138, 74139, 74151, 74153, 74157
Parity Generator	74180
Register	74273, 74374
Shift Register	7491, 74164, 74165 74194

Appendix A contains a foldout page showing the 30 Standard Altera MacroFunctions.

For a current list of all available MacroFunctions, write to Altera's Applications Department.

16CUDSLR (Counter)



Name:	**16CUDSLR** (16-Bit Binary Up/Down Counter With Left/Right Shift Register)
Declaration:	16CUDSLR(STCT,DNUP,LTRT,DATA,CK, Q16,Q15,Q14,Q13,Q12,Q11,Q10,Q9,Q8, Q7,Q6,Q5,Q4,Q3,Q2,Q1) (STCT = Shift/Count; DNUP = Up/Down; LTRT = Left/Right; DATA = Serial Data Input; CK = Clock)
EPLDs:	EP600, EP610, EP900, EP910, EP1800, EPB1400
Default Signal Levels:	GND — all input pins

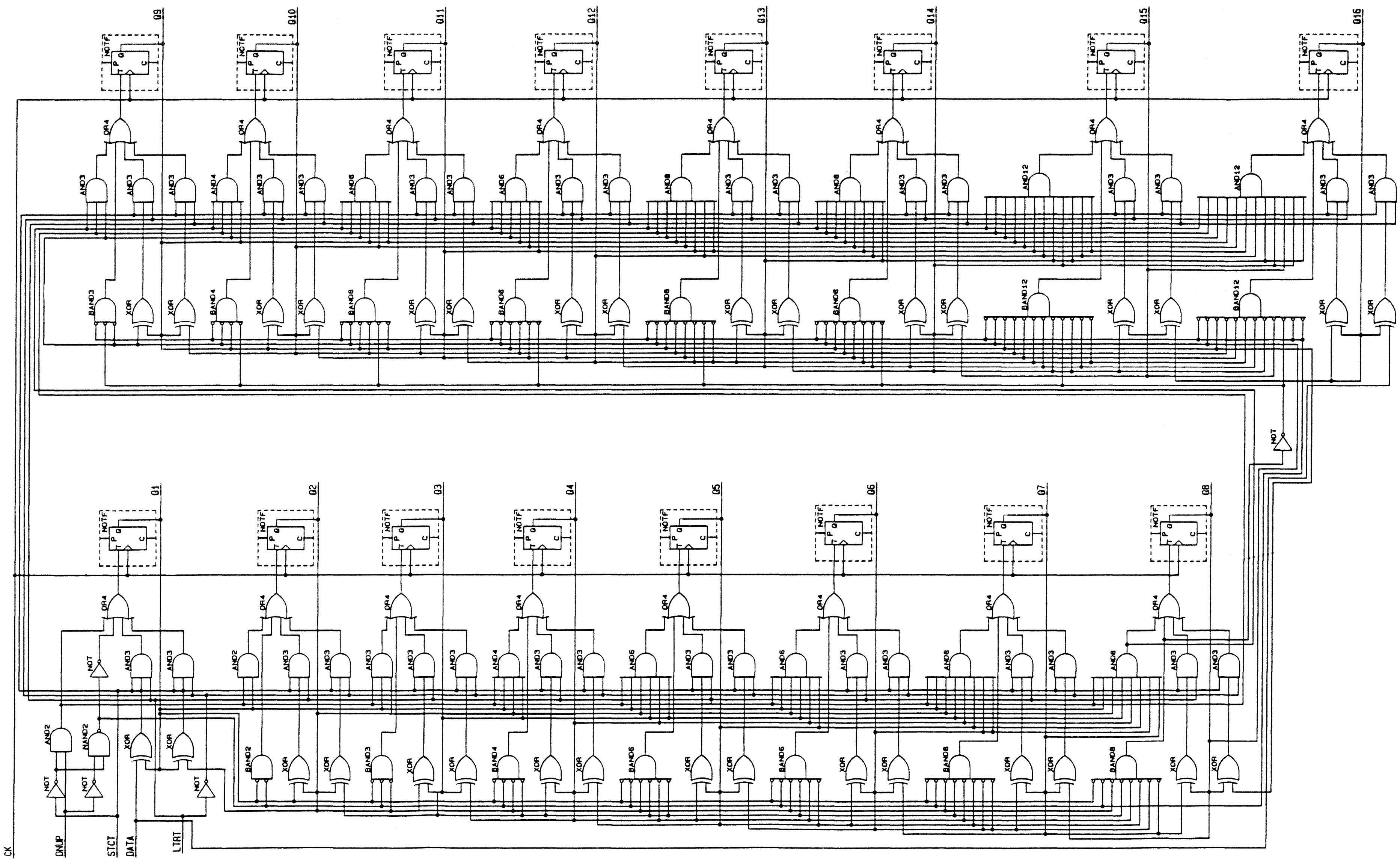
16CUDSLR Function Table:

16CUDSLR Function Table

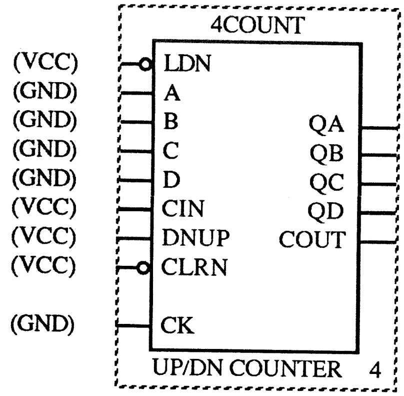
INPUTS			OPERATION
STCT	DNUP	LTRT	
H	X	H	SHIFT LEFT
H	X	L	SHIFT RIGHT
L	H	X	COUNT UP
L	L	X	COUNT DOWN

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)

16CUDSLR Logic Schematic:



4COUNT (Counter)



Name:	**4COUNT** (4-Bit Up/Down Binary Counter With Synchronous Load and Asynchronous Clear)
Declaration:	4COUNT(CLRN,LDN,DNUP,CIN,A,B,C,D, CK,QD,QC,QB,QA,COUT) (LDN = Load, Active Low; CIN = Carry In; DNUP = Down/Up; CLRN = Clear, Active Low; CK = Clock; COUT = Carry Out)
EPLDs:	EP600, EP610, EP900, EP910, EP1800, EPB1400
Default Signal Levels:	GND — CK, A, B, C, D VCC — CLRN, LDN, DNUP,CIN

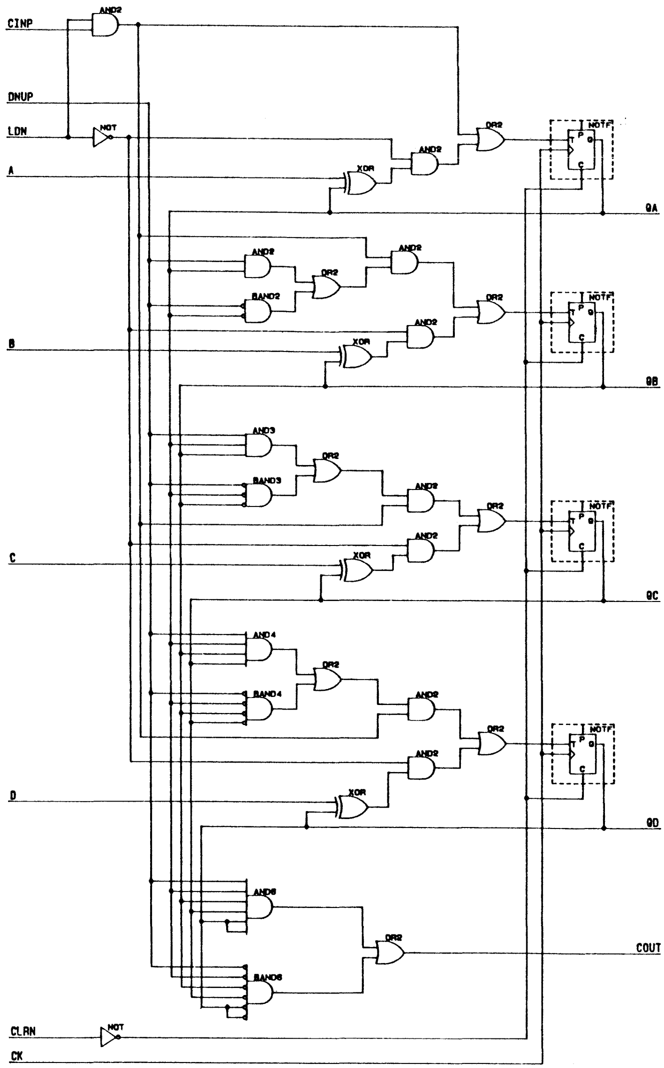
4COUNT Function Table:

4COUNT Function Table

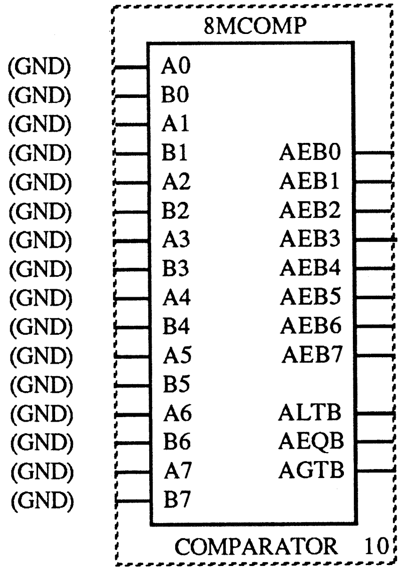
INPUTS									OUTPUTS				
CK	LDN	CLRN	DNUP	CIN	D	C	B	A	Q_D	Q_C	Q_B	Q_A	COUT
X	X	L	X	X					L	L	L	L	X
↑	L	H	X	X	d	c	b	a	d	c	b	a	X
↑	H	H	X	L					HOLD				X
↑	H	H	L	H					COUNT DOWN				L
↑	H	H	H	H					COUNT UP				L
↑	H	H	L	H					H	H	H	H	H
↑	H	H	H	H					L	L	L	L	H

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

4COUNT Logic Schematic:



8MCOMP (Comparator)



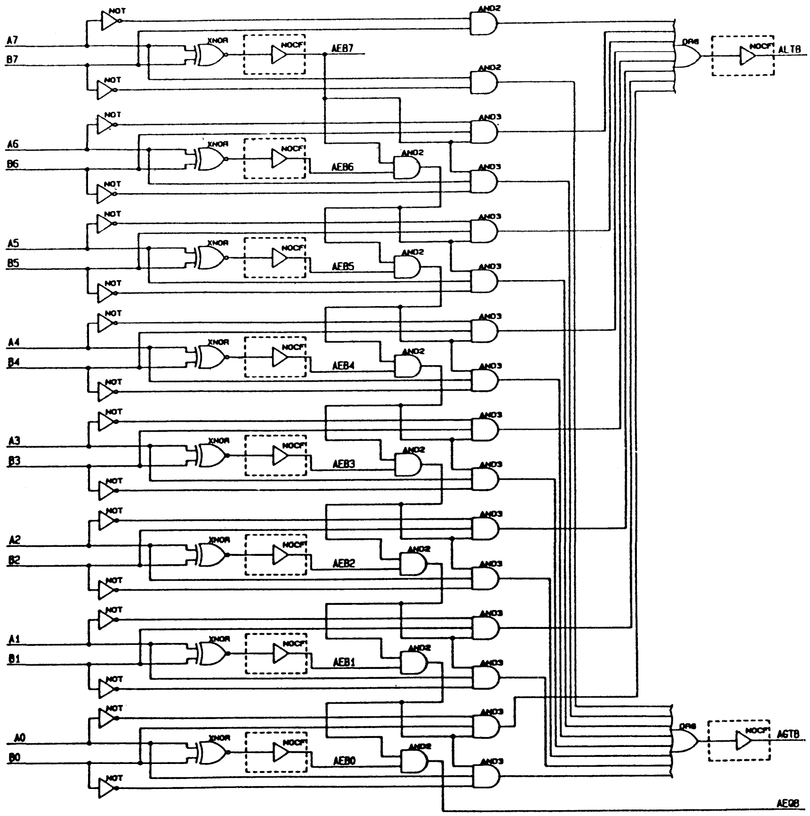
Name:	**8MCOMP** (8-Bit Magnitude Comparator)
Declaration:	8MCOMP(A0,B0,A1,B1,A2,B2,A3,B3,A4,B4, A5,B5,A6,B6,A7,B7,AGTB,AEQB,ALTB, AEB7,AEB6,AEB5,AEB4,AEB3,AEB2, AEB1,AEB0) (AGTB = A Greater Than B; AEQB = Word A Equals Word B; ALTB = A Less Than B; AEB = Bit A Equals Bit B)
EPLDs:	EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — all input pins

8MCOMP Function Table:

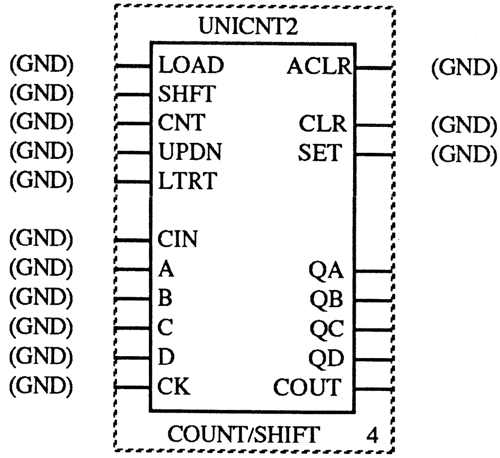
8MCOMP Function Table

INPUTS	OUTPUTS			
A B	ALTB	AEQB	AGTB	AEBx
A=B	0	1	0	
A<B	1	0	0	
A>B	0	0	1	
Ax=Bx				1
Ax≠Bx				0
x = Bit position				

8MCOMP Logic Schematic:



UNICNT2 (Counter)



Name: **UNICNT2** (Universal 4-Bit Up/Down Counter, Left/Right Shift Register with Load, Clear; Cascade)

Declaration: UNICNT2(LOAD,SHFT,CNT,UPDN,LTRT,CIN, A,B,C,D,CK,COUT,QD,QC,QB,QA,SET, CLR,ACLR)

(LOAD = Load; SHFT = Shift; CNT = Count; UPDN = Up/Down; LTRT = Left/Right; CIN = Count In; CK = Clock; ACLR = Asynchronous Clear; CLR = Clear; SET = Set; COUT = Carry Out)

EPLDs: EP600, EP610, EP900, EP910, EP1800, EPB1400

Default Signal Levels: GND — all input pins

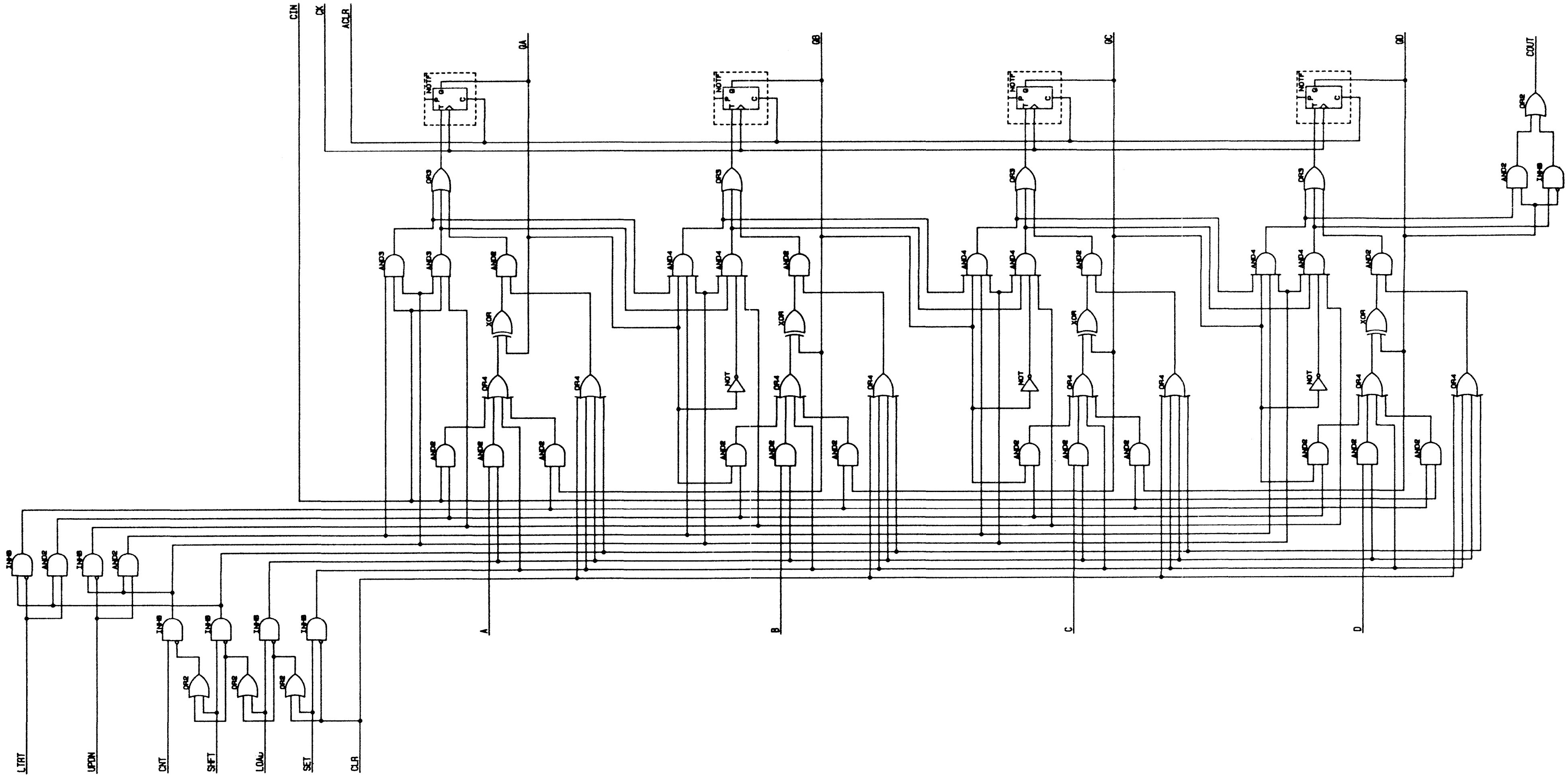
UNICNT2 Function Table:

UNICNT2 Function Table

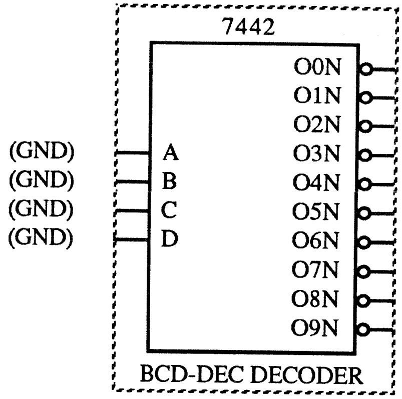
INPUTS									OUTPUTS				
CK	ACLR	CLR	SET	LOAD	SHFT	LTRT	CNT	UPDN	Q_D	Q_C	Q_B	Q_A	COUT
X	1	X	X	X	X	X	X	X	0	0	0	0	X
↑	0	1	X	X	X	X	X	X	0	0	0	0	/UPDN
↑	0	0	1	X	X	X	X	X	1	1	1	1	UPDN
↑	0	0	0	1	X	X	X	X	d	c	b	a	0
↑	0	0	0	0	1	1	X	X	CIN	QD	QC	QB	0
↑	0	0	0	0	1	0	X	X	QC	QB	QA	CIN	0
↑	0	0	0	0	0	X	1	1	4-Bit Binary Up Counter				
↑	0	0	0	0	0	X	1	0	4-Bit Binary Down Counter				

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

UNICNT2 Logic Schematic:



7442 (Decoder)



Name:	**7442** (1:10 BCD to Decimal Decoder)
Declaration:	7442(A,B,C,D,O9N,O8N,O7N,O6N,O5N, O4N,O3N,O2N,O1N,O0N)
EPLDs:	EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — all input pins

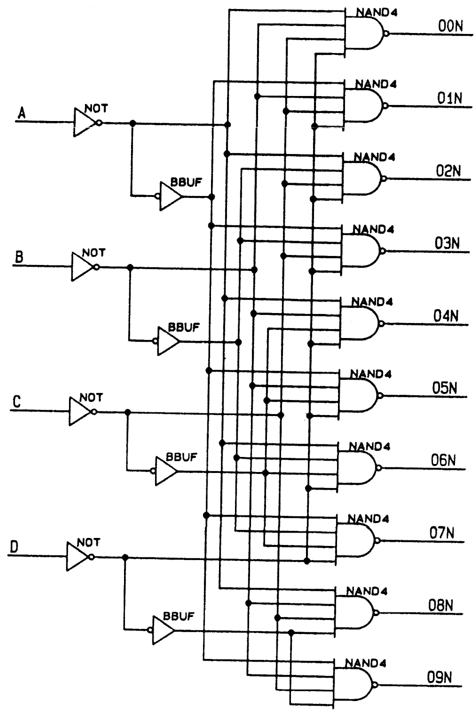
7442 Function Table:

7442 F unction Table

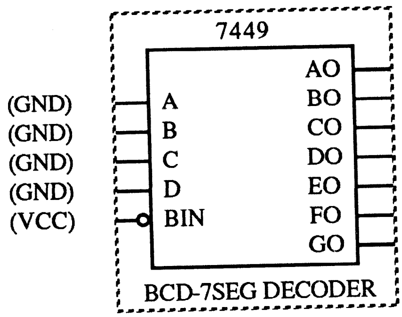
NO.	BCD INPUT				DECIMAL OUTPUT									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
I	H	L	H	L	H	H	H	H	H	H	H	H	H	H
N	H	L	H	H	H	H	H	H	H	H	H	H	H	H
V	H	H	L	L	H	H	H	H	H	H	H	H	H	H
A	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	H	H	H	L	H	H	H	H	H	H	H	H	H	H
I	H	H	H	H	H	H	H	H	H	H	H	H	H	H
D														

H = high level
L = low level

7442 Logic Schematic:



7449 (Decoder)



Name:	**7449** (BCD-To-7 Segment Decoder)
Declaration:	7449(A,B,C,D,BIN,GO,FO,EO,DO,CO, BO,AO)
EPLDs:	All
Default Signal Levels:	GND — A, B, C, D VCC — BIN

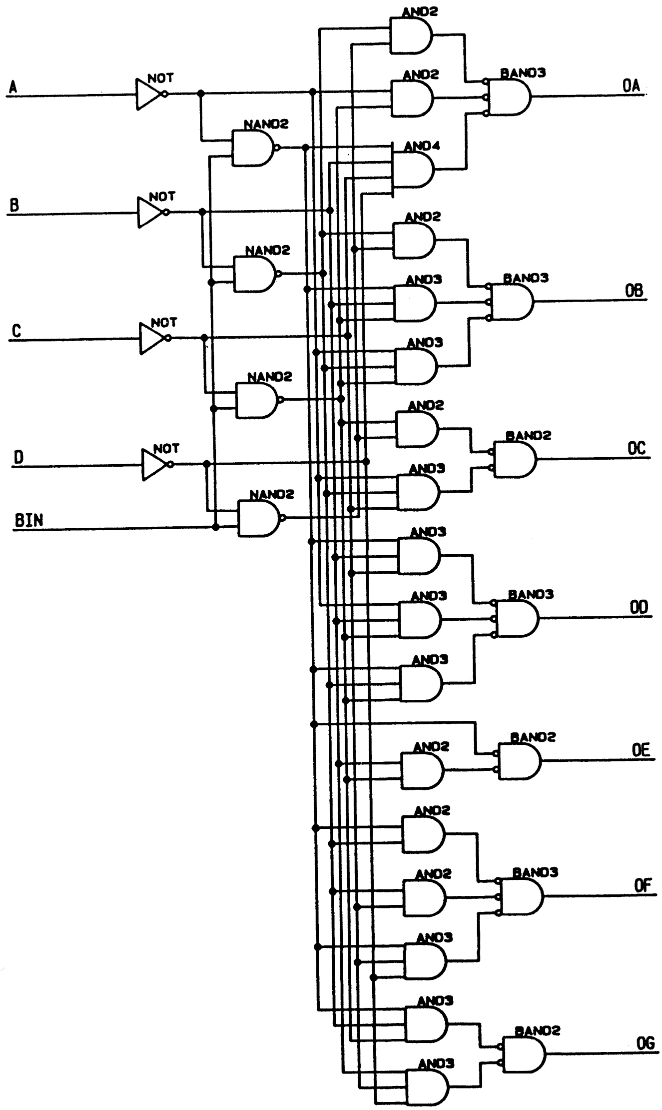
7449 Function Table:

7449 Function Table

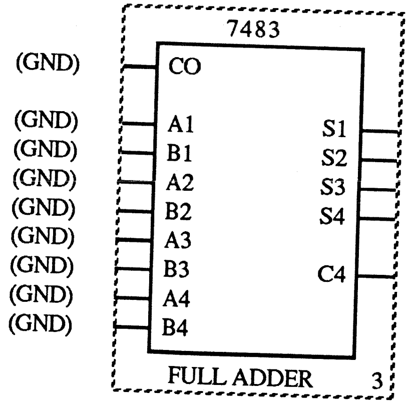
Decimal or Function	INPUTS					OUTPUTS							Note
	D	C	B	A	BIN	AO	BO	CO	DO	EO	FO	GO	
0	L	L	L	L	H	H	H	H	H	H	H	L	1
1	L	L	L	H	H	L	H	H	L	L	L	L	
2	L	L	H	L	H	H	H	L	H	H	L	H	
3	L	L	H	H	H	H	H	H	H	L	L	H	
4	L	H	L	L	H	L	H	H	L	L	H	H	
5	L	H	L	H	H	H	L	H	H	L	H	H	
6	L	H	H	L	H	L	L	H	H	H	H	H	
7	L	H	H	H	H	H	H	H	L	L	L	L	
8	H	L	L	L	H	H	H	H	H	H	H	H	
9	H	L	L	H	H	H	H	H	L	L	H	H	
10	H	L	H	L	H	L	L	L	H	H	L	H	
11	H	L	H	H	H	L	L	H	H	L	L	H	
12	H	H	L	L	H	L	H	L	L	L	H	H	
13	H	H	L	H	H	H	L	L	H	L	H	H	
14	H	H	H	L	H	L	L	L	H	H	H	H	
15	H	H	H	H	H	L	L	L	L	L	L	L	
BI	X	X	X	X	L	L	L	L	L	L	L	L	2

H = high level
L = low level
X = don't care
NOTES: 1. The blanking input (BIN) must be held at a high logic level when output functions 0 through 15 are desired.
2. When a low logic level is applied directly to the blanking input (BIN), all segment outputs are low regardless of the level of any other input.

7449 Logic Schematic:



7483 (Adder)



Name:	**7483** (4-Bit Full Adder)
Declaration:	7483(C0,A1,B1,A2,B2,A3,B3,A4,B4,C4,S4, S3,S2,S1)
EPLDs:	All
Default Signal Levels:	GND — all input pins

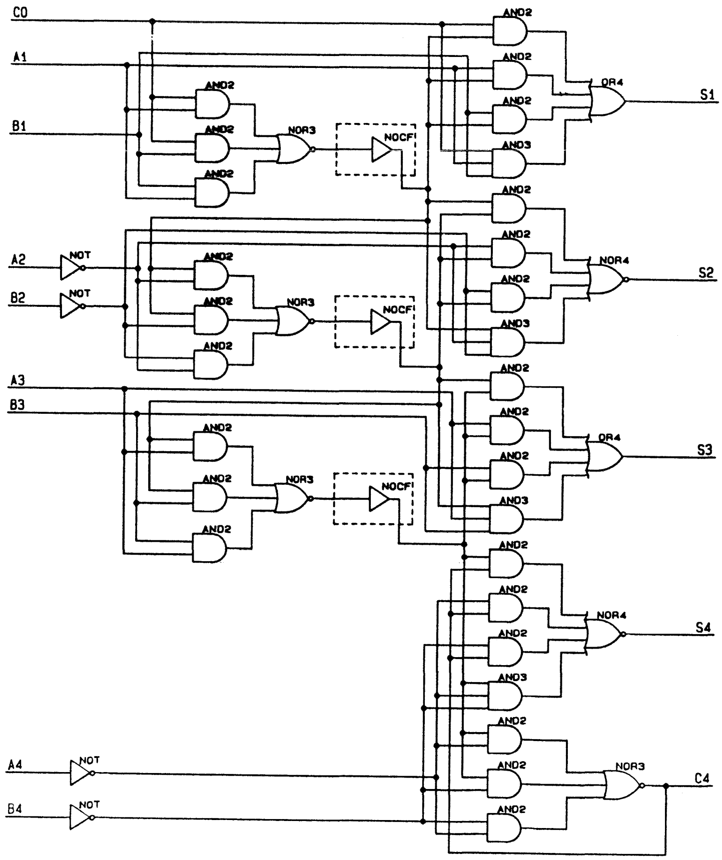
7483 Function Table:

7483 Function Table

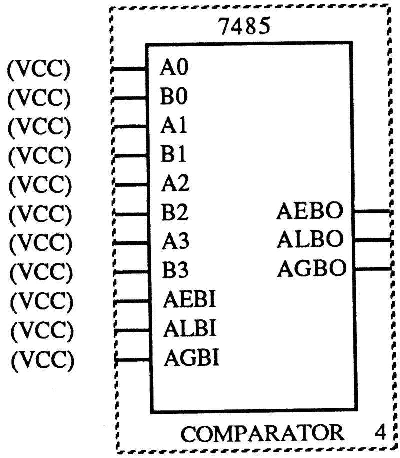
INPUT				OUTPUT					
				WHEN C0 = L / WHEN C2 = L			WHEN C0 = H / WHEN C2 = H		
A1 / A3	B1 / B3	A2 / A4	B2 / B4	S1 / S3	S2 / S4	C2 / C4	S1 / S3	S2 / S4	C2 / C4
L	L	L	L	L	L	L	H	L	L
H	L	L	L	H	L	L	L	H	L
L	H	L	L	H	L	L	L	H	L
H	H	L	L	L	H	L	H	H	L
L	L	H	L	L	H	L	H	H	L
H	L	H	L	H	H	L	L	L	H
L	H	H	L	H	H	L	L	L	H
H	H	H	L	L	L	H	H	L	H
L	L	L	H	L	H	L	H	H	L
H	L	L	H	H	H	L	L	L	H
L	H	L	H	H	H	L	L	L	H
H	H	L	H	L	L	H	H	L	H
L	L	H	H	L	L	H	H	L	H
H	L	H	H	H	L	H	L	H	H
L	H	H	H	H	L	H	L	H	H
H	H	H	H	L	H	H	H	H	H

H = high level
L = low level
Input conditions at A1, B1, A2, B2, C0 are used to determine outputs S1 and S2 and the value of the internal carry C2. The values at C2, A3, B3, A4, B4 are then used to determine outputs S3, S4, C4.

7483 Logic Schematic:



7485 (Comparator)



Name: **7485** (4-Bit Magnitude Comparator)

Declaration: 7485(A0,B0,A1,B1,A2,B2,A3,B3,AEBI,ALBI, AGBI,AGBO,ALBO,AEBO)

EPLDs: All

Default Signal Levels: VCC — all input pins

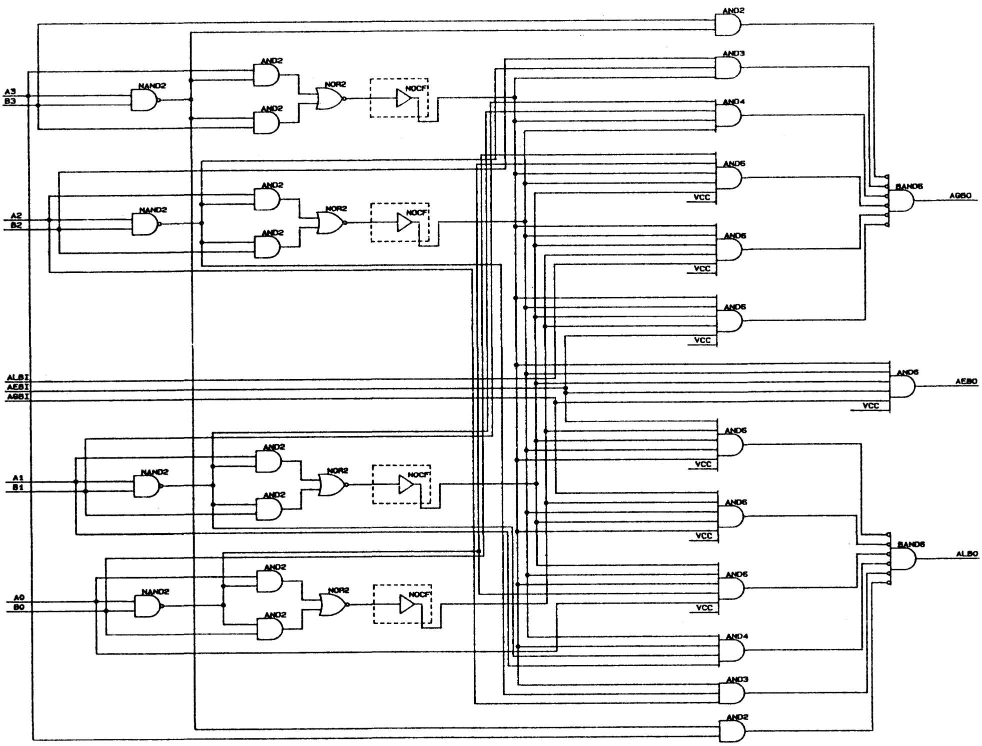
7485 Function Table:

7485 Function Table

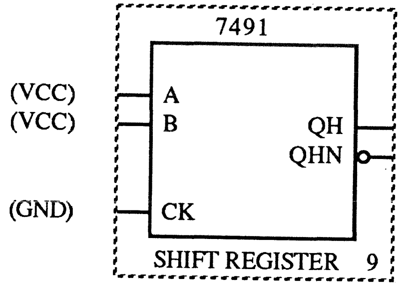
COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3,B3	A2,B2	A1,B1	A0,B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	X	X	X	X	X	X	H	L	L
A3<B3	X	X	X	X	X	X	L	H	L
A3=B3	A2>B2	X	X	X	X	X	H	L	L
A3=B3	A2<B2	X	X	X	X	X	L	H	L
A3=B2	A2=B2	A1>B1	X	X	X	X	H	L	L
A3=B3	A2=B2	A1<B1	X	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	H	L	L	H	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	H	L	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	L	L	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	L	H	H	L

H = high level L = low level X = don't care

7485 Logic Schematic:



7491 (Shift Register)



Name: **7491** (Serial-In Serial-Out Shift Register)

Declaration: 7491(A,B,CK,QHN,QH)

EPLDs: All

Default Signal Levels: GND — CK
VCC — A, B

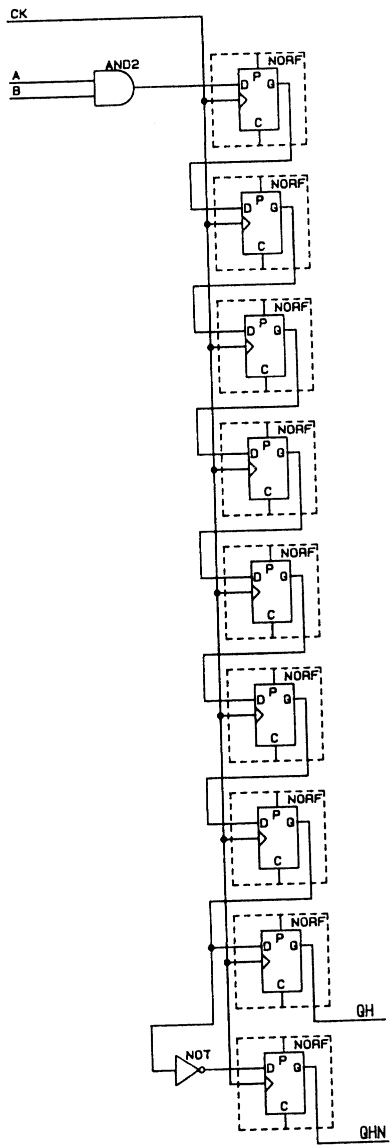
Function Table:

7491 Function Table

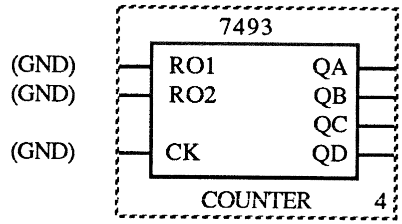
INPUTS AT t_n		OUTPUTS AT t_{n+8}	
A	B	Q_H	$\overline{Q}_H$
H	H	H	L
L	X	L	H
X	L	L	H

H = high L = low X = don't care
t_n = reference bit time, clock low
t_{n+8} = bit time after 8 low-to-high clock transitions

7491 Logic Schematic:



7493 (Counter)



Name:	**7493** (4-Bit Binary Counter)
Declaration:	7493(RO1,RO2,CK,QD,QC,QB,QA)
EPLDs:	EP310, EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — all input pins

7493 Function Table:

7493 Function Table (Count Sequence)

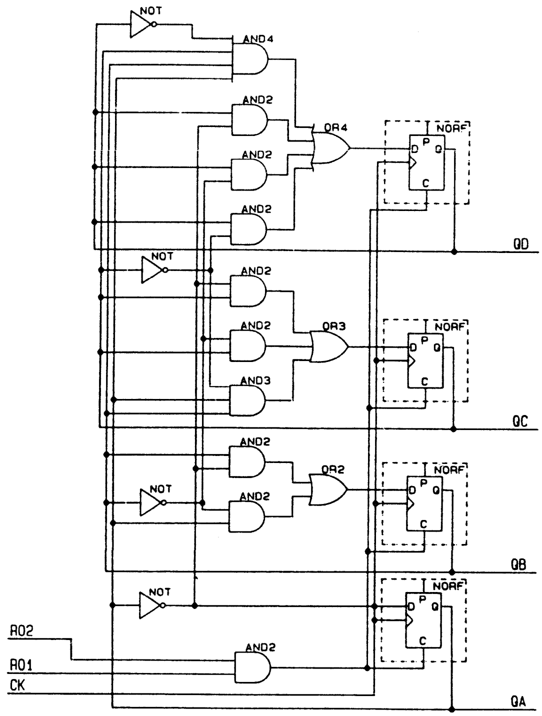
COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

H = high level
L = low level

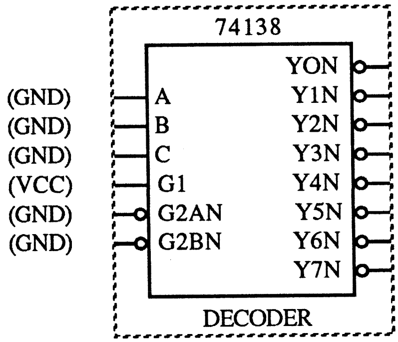
7493 Function Table (Reset/Count)

RESET INPUTS		OUTPUT			
$R_{O(1)}$	$R_{O(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

7493 Logic Schematic:



74138 (Decoder)



Name:	**74138** (3:8 Decoder)
Declaration:	74138(G1,G2AN,G2BN,A,B,C,D,Y7N,Y6N, Y5N,Y4N,Y3N,Y2N,Y1N,Y0N)
EPLDs:	All
Default Signal Levels:	GND — G2AN, G2BN, A, B, C VCC — G1

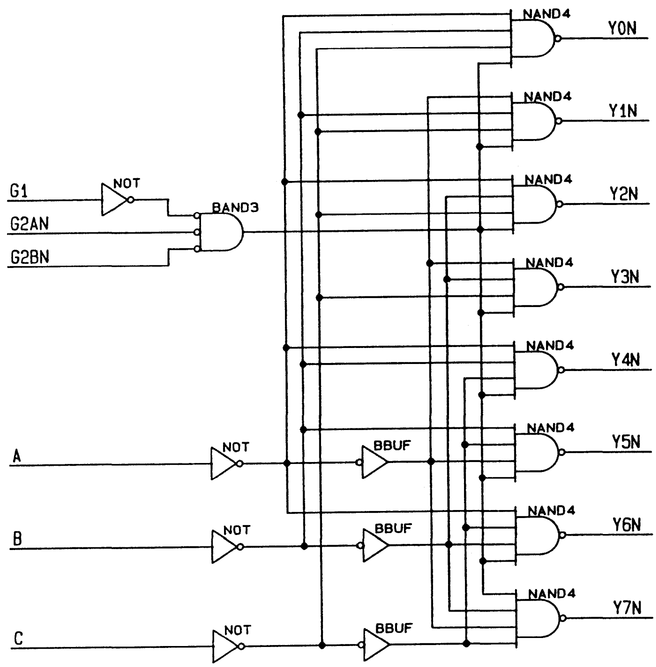
74138 Function Table:

74138 Function Table

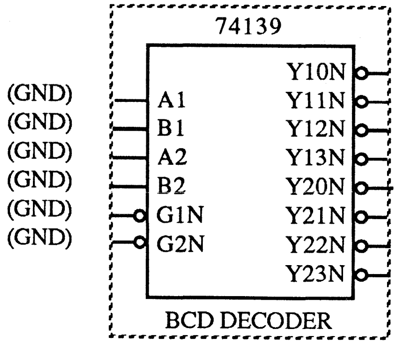
INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0N	Y1N	Y2N	Y3N	Y4N	Y5N	Y6N	Y7N
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

*G2 = G2AN + G2BN
H = high level
L = low level
X = don't care

74138 Logic Schematic:



74139 (Decoder)



Name: **74139** (1:4 Decoder)

Declaration: 74139(G1N,G2N,A1,B1,A2,B2,Y23N,Y22N, Y21N,Y20N,Y13N,Y12N,Y11N,Y10N)

EPLDs: All

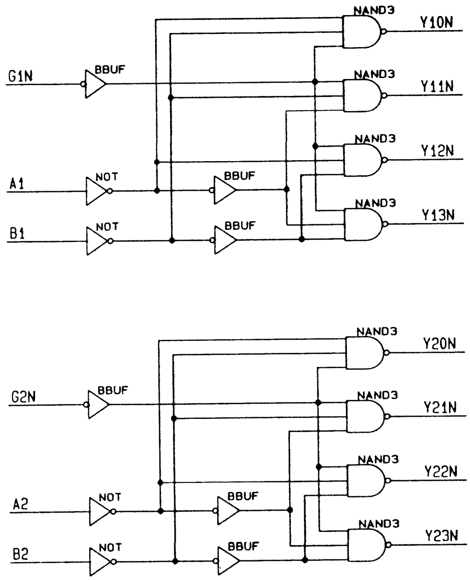
Default Signal Levels: GND — all input pins

Function Table:

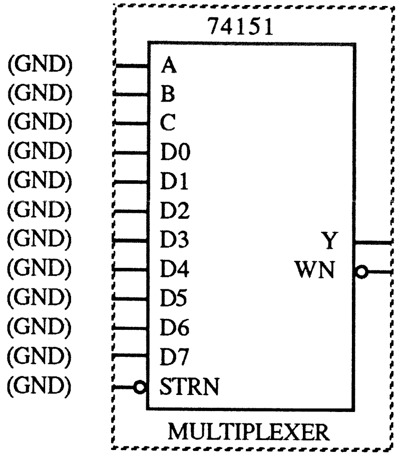
74139 Function Table

INPUTS			OUTPUTS			
ENABLE	SELECT					
GN	B	A	Y0N	Y1N	Y2N	Y3N
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L
H = high level L = low level X = don't care						

74139 Logic Schematic:



74151 (Multiplexer)



Name:	**74151** (8:1 Multiplexer)
Declaration:	74151(STRN,D0,D1,D2,D3,D4,D5,D6,D7, A,B,C,WN,Y)
EPLDs:	All
Default Signal Levels:	GND — all input pins

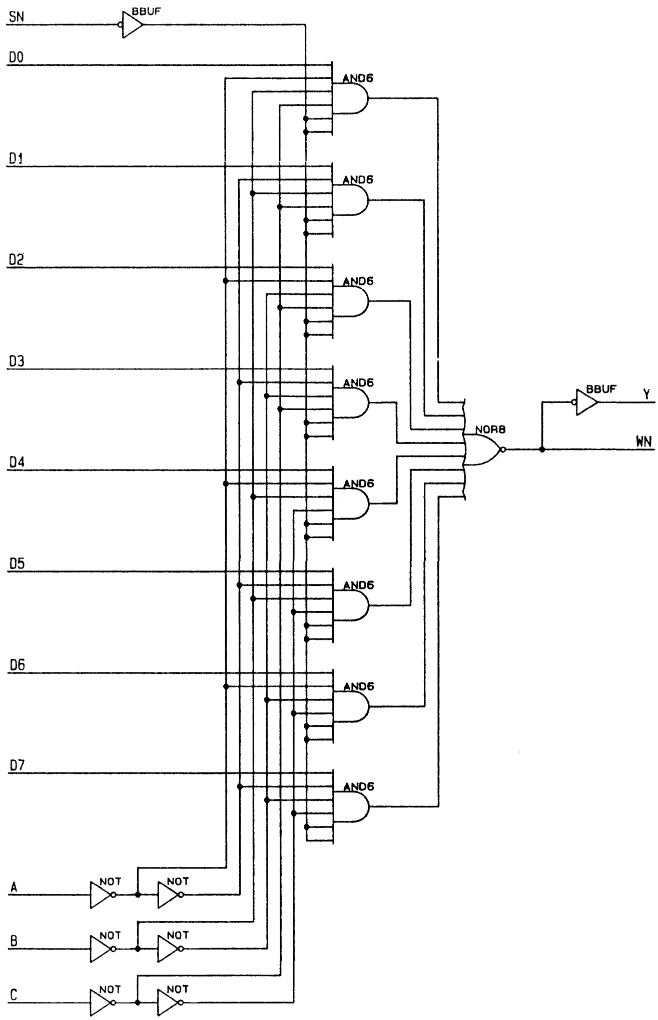
74151 Function Table:

74151 Function Table

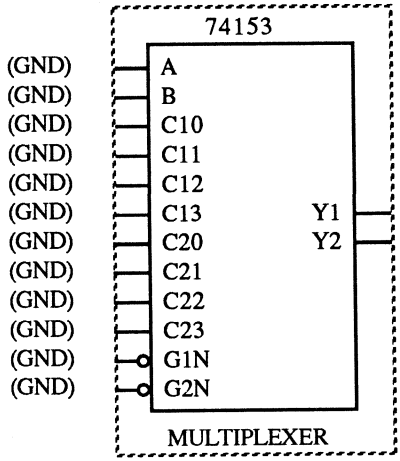
INPUTS				OUTPUTS	
SELECT			STROBE		
C	B	A	SN	Y	WN
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = high level   L = low level   X = don't care
D0, D1, ... D7 = the level of the D input

74151 Logic Schematic:



74153 (Multiplexer)



Name: **74153** (Dual 4:1 Multiplexer)

Declaration: 74153(G1N,C10,C11,C12,C13,B,A,C20,
C21,C22,C23,G2N,Y2,Y1)

EPLDs: All

Default Signal Levels: GND — all input pins

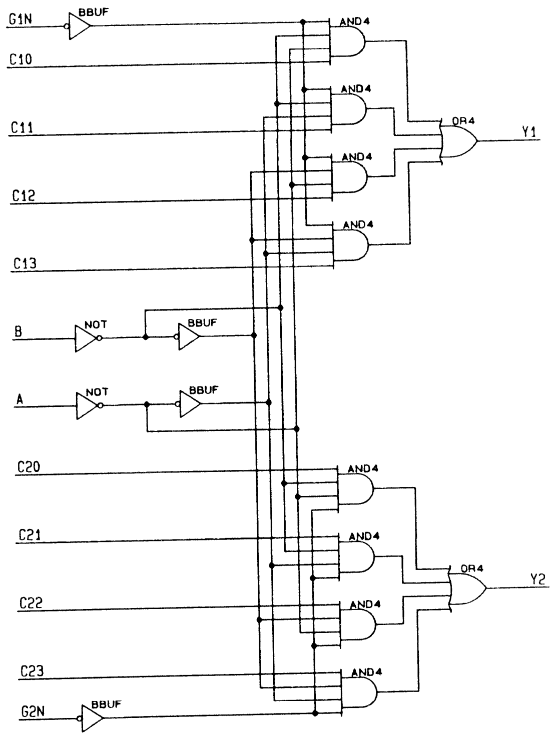
74153 Function Table:

74153 Function Table

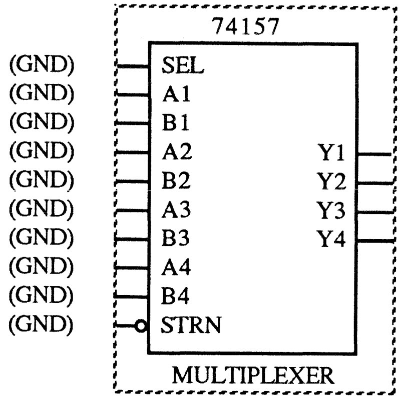
SELECT INPUTS		DATA INPUTS				STROBE	OUTPUT
B	A	C0	C1	C2	C3	GN	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

H = high level L = low level X = don't care
Select inputs A and B are common to both sections.

74153 Logic Schematic:



74157 (Multiplexer)



Name: **74157** (Quad 4:1 Multiplexer)

Declaration: 74157(A1,B1,A2,B2,A3,B3,A4,B4,STRN, SEL,Y4,Y3,Y2,Y1)

EPLDs: All

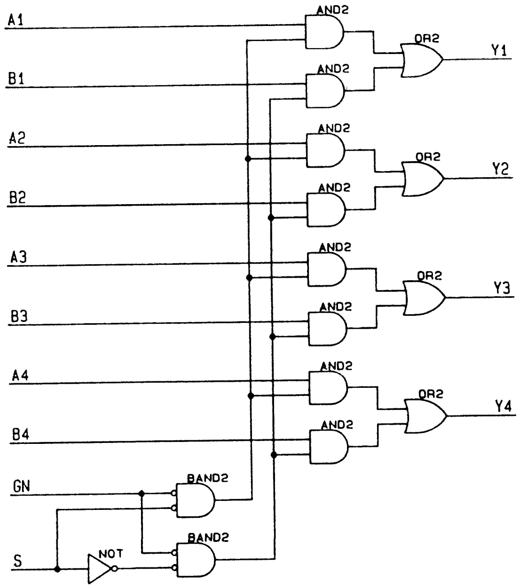
Default Signal Levels: GND — all input pins

Function Table:

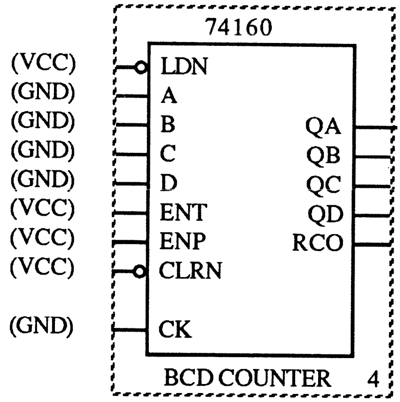
74157 Function Table

INPUTS				OUTPUT Y
STRN	SEL	A	B	
H	X	X	X	L
L	L	L	X	L
L	L	H	X	H
L	H	X	L	L
L	H	X	H	H
H = high level L = low level X = don't care				

74157 Logic Schematic:



74160 (Counter)



Name:	**74160** (4-Bit Decade Counter With Synchronous Load, Asynchronous Clear)
Declaration:	74160(CLRN,LDN,ENP,ENT,A,B,C,D,CK, QD,QC,QB,QA,RCO)
EPLDs:	EP310, EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — A, B, C, D, CK VCC — LDN, ENT, ENP, CLRN

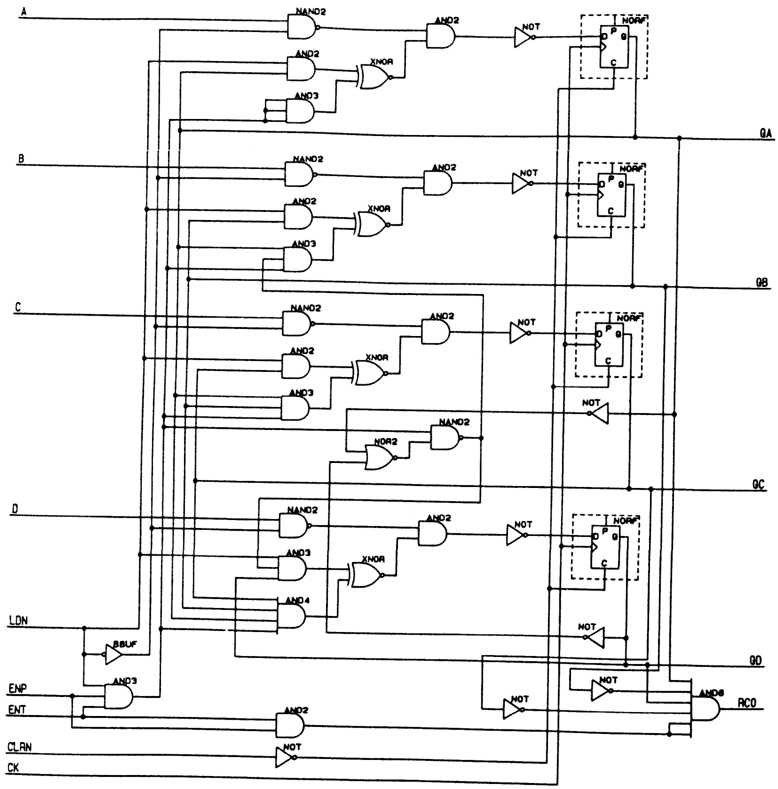
74160 Function Table:

74160 Function Table

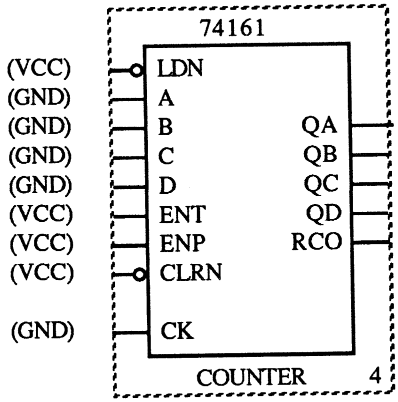
INPUTS									OUTPUTS				
CK	LDN	CLRN	ENP	ENT	D	C	B	A	Q_D	Q_C	Q_B	Q_A	RCO
X	X	L	X	X					L	L	L	L	L
↑	L	H	X	X	d	c	b	a	d	c	b	a	L
↑	H	H	X	L					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	L	X					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	H	H					L	L	L	L	L
↑	H	H	H	H					L	L	L	H	L
↑	H	H	H	H					L	L	H	L	L
↑	H	H	H	H					L	L	H	H	L
↑	H	H	H	H					L	H	L	L	L
↑	H	H	H	H					L	H	L	H	L
↑	H	H	H	H					L	H	H	L	L
↑	H	H	H	H					L	H	H	H	L
↑	H	H	H	H					H	L	L	L	L
↑	H	H	H	H					H	L	L	H	H

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

74160 Logic Schematic:



74161 (Counter)



Name:	**74161** (4-Bit Binary Up Counter With Synchronous Load, Asynchronous Clear)
Declaration:	74161(CLRN,LDN,ENP,ENT,A,B,C,D,CK, QD,QC,QB,QA,RCO)
EPLDs:	EP310, EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — A, B, C, D, CK VCC — LDN, ENT, ENP, CLRN

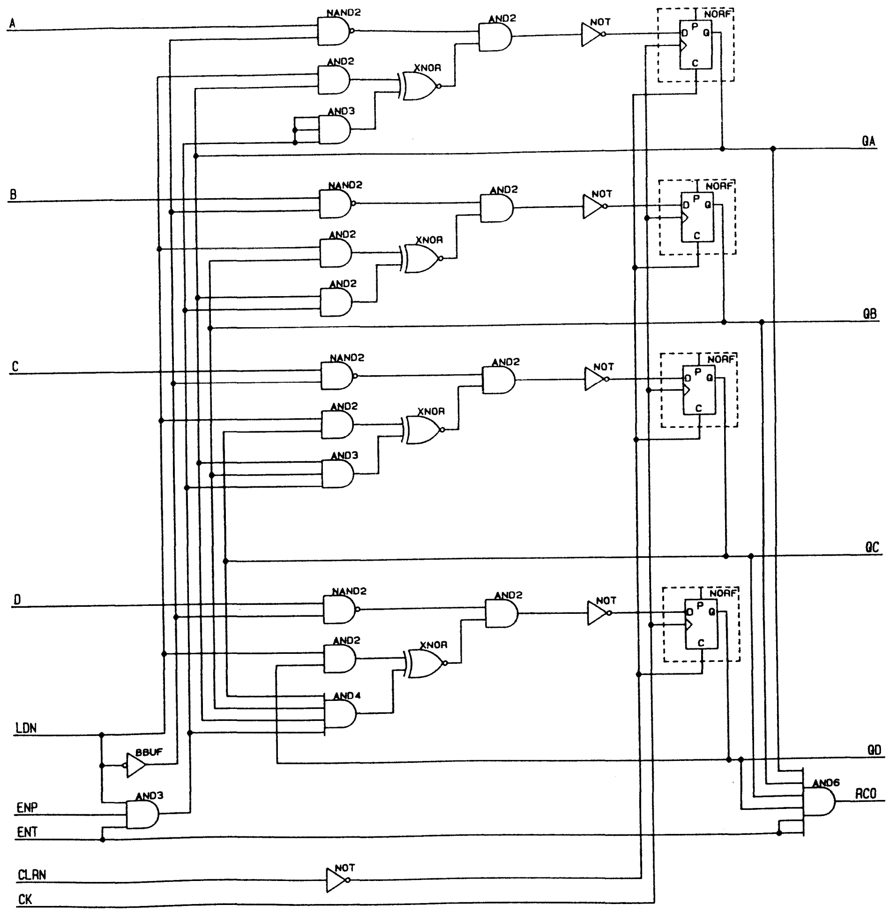
74161 Function Table:

74161 Function Table

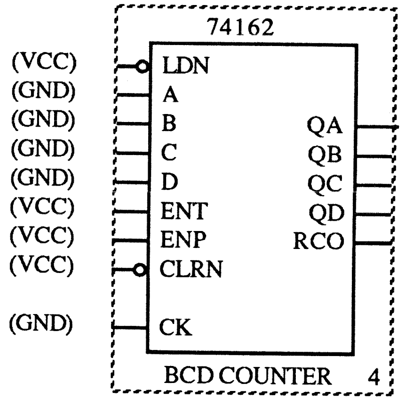
INPUTS									OUTPUTS				
CK	LDN	CLRN	ENP	ENT	D	C	B	A	Q_D	Q_C	Q_B	Q_A	RCO
X	X	L	X	X					L	L	L	L	L
↑	L	H	X	X	d	c	b	a	d	c	b	a	L
↑	H	H	X	H					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	L	L					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	H	H					L	L	L	L	L
↑	H	H	H	H					L	L	L	H	L
↑	H	H	H	H					L	L	H	L	L
↑	H	H	H	H					L	L	H	H	L
↑	H	H	H	H					L	H	L	L	L
↑	H	H	H	H					L	H	L	H	L
↑	H	H	H	H					L	H	H	L	L
↑	H	H	H	H					L	H	H	H	L
↑	H	H	H	H					H	L	L	L	L
↑	H	H	H	H					H	L	L	H	L
↑	H	H	H	H					H	L	H	L	L
↑	H	H	H	H					H	L	H	H	L
↑	H	H	H	H					H	H	L	L	L
↑	H	H	H	H					H	H	L	H	L
↑	H	H	H	H					H	H	H	L	L
X	H	H	H	H					H	H	H	H	H

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

74161 Logic Schematic:



74162 (Counter)



Name:	**74162** (4-Bit Up Decade Counter With Synchronous Load and Clear)
Declaration:	74162(CLRN,LDN,ENP,ENT,A,B,C,D,CK, QD,QC,QB,QA,RCO)
EPLDs:	All
Default Signal Levels:	GND — A, B, C, D, CK VCC — LDN, ENT, ENP, CLRN

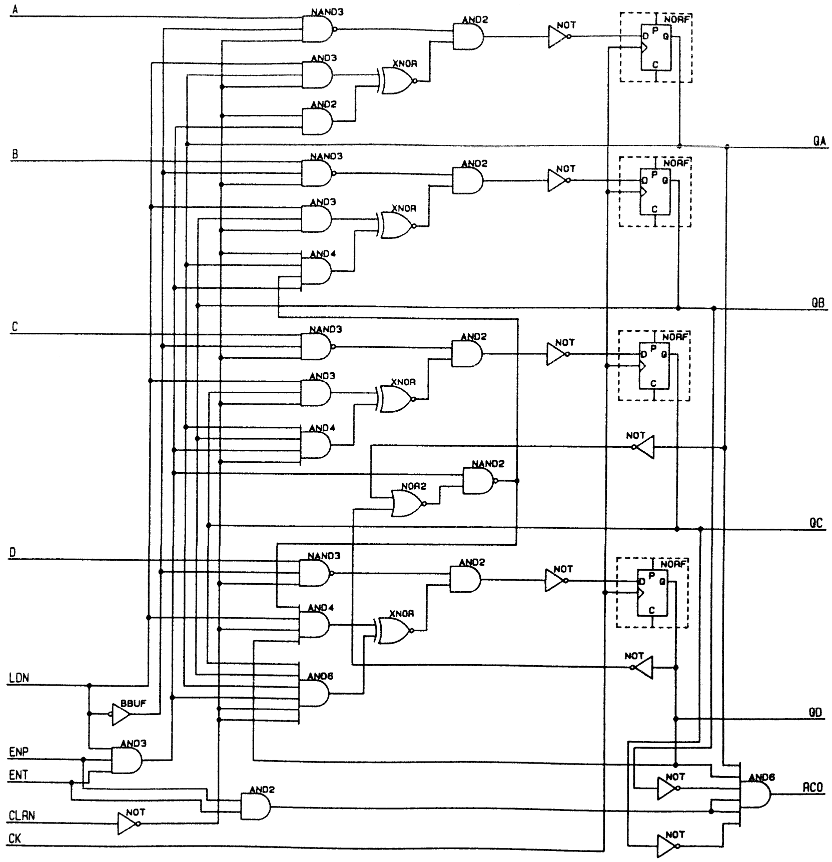
74162 Function Table:

74162 Function Table

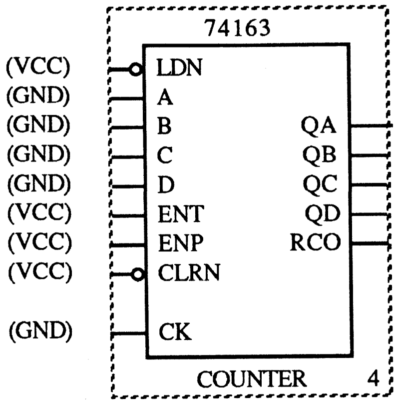
INPUTS									OUTPUTS				
CK	LDN	CLRN	ENP	ENT	D	C	B	A	Q_D	Q_C	Q_B	Q_A	RCO
↑	X	L	X	X					L	L	L	L	L
↑	L	H	X	X	d	c	b	a	d	c	b	a	L
↑	H	H	X	L					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	L	X					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	H	H					L	L	L	L	L
↑	H	H	H	H					L	L	L	H	L
↑	H	H	H	H					L	L	H	L	L
↑	H	H	H	H					L	L	H	H	L
↑	H	H	H	H					L	H	L	L	L
↑	H	H	H	H					L	H	L	H	L
↑	H	H	H	H					L	H	H	L	L
↑	H	H	H	H					L	H	H	H	L
↑	H	H	H	H					H	L	L	L	L
X	H	H	H	H					H	L	L	H	H

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

74162 Logic Schematic:



74163 (Counter)



Name:	**74163** (4-Bit Up Binary Counter With Synchronous Load and Clear)
Declaration:	74163(CLRN,LDN,ENP,ENT,A,B,C,D,CK, QD,QC,QB,QA,RCO)
EPLDs:	All
Default Signal Levels:	GND — A, B, C, D, CK VCC — LDN, ENT, ENP, CLRN

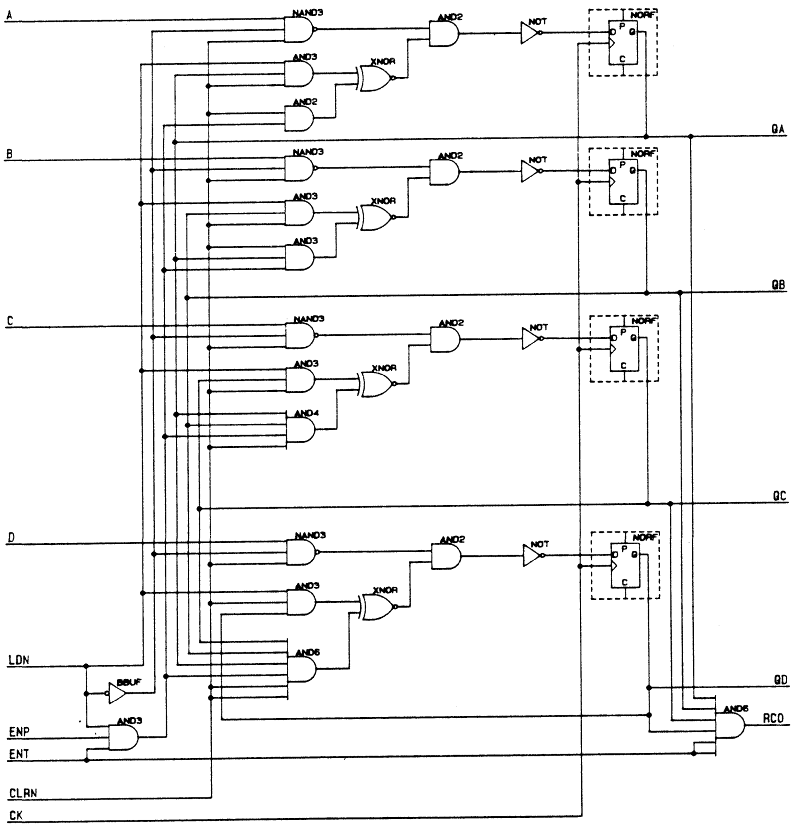
74163 Function Table:

74163 Function Table

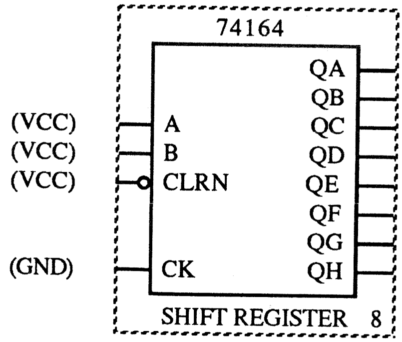
INPUTS									OUTPUTS				
CK	LDN	CLRN	ENP	ENT	D	C	B	A	Q_D	Q_C	Q_B	Q_A	RCO
↑	X	L	X	X					L	L	L	L	L
↑	L	H	X	X	d	c	b	a	d	c	b	a	L
↑	H	H	L	H					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	H	L					Q_D	Q_C	Q_B	Q_A	L
↑	H	H	H	H					L	L	L	L	L
↑	H	H	H	H					L	L	L	H	L
↑	H	H	H	H					L	L	H	L	L
↑	H	H	H	H					L	L	H	H	L
↑	H	H	H	H					L	H	L	L	L
↑	H	H	H	H					L	H	L	H	L
↑	H	H	H	H					L	H	H	L	L
↑	H	H	H	H					L	H	H	H	L
↑	H	H	H	H					H	L	L	L	L
↑	H	H	H	H					H	L	L	H	L
↑	H	H	H	H					H	L	H	L	L
↑	H	H	H	H					H	L	H	H	L
↑	H	H	H	H					H	H	L	L	L
↑	H	H	H	H					H	H	L	H	L
↑	H	H	H	H					H	H	H	L	L
X	H	H	H	H					H	H	H	H	H

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

74163 Logic Schematic:



74164 (Shift Register)



Name:	**74164** (Serial-In Parallel-Out Shift Register)
Declaration:	74164(A,B,CLRN,CK,QH,QG,QF,QE,QD, QC,QB,QA)
EPLDs:	EP310, EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — CK VCC — A, B, CLRN

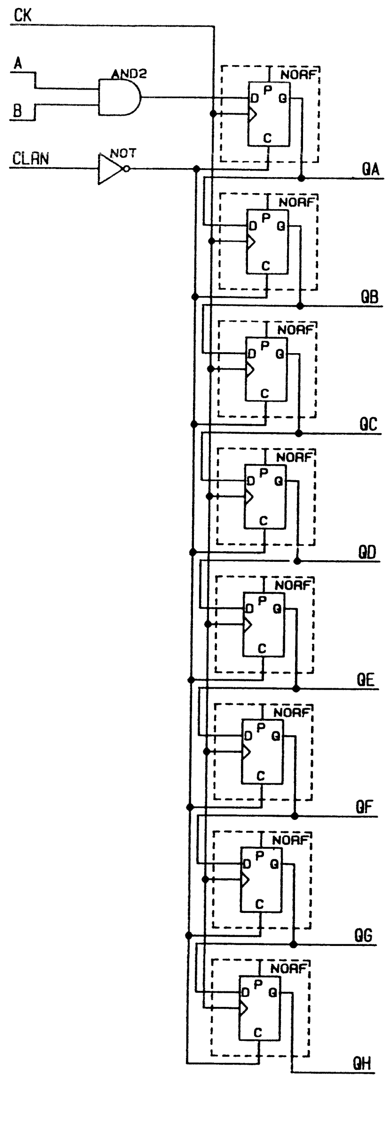
74164 Function Table:

74164 Function Table

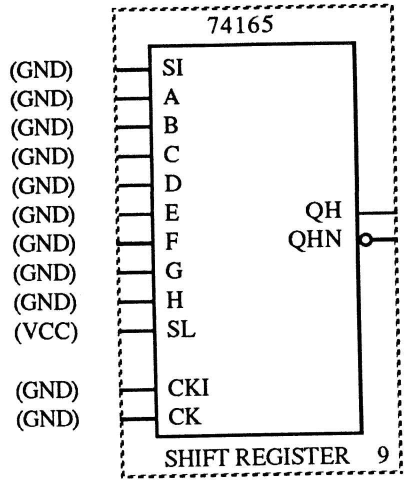
INPUTS				OUTPUTS		
CK	CLRN	A	B	Q_A	Q_B	... Q_H
X	L	X	X	L	L	L
L	H	X	X	Q_{A0}	Q_{B0}	Q_{H0}
↑	H	H	H	H	Q_{An}	Q_{Gn}
↑	H	L	X	L	Q_{An}	Q_{Gn}
↑	H	X	L	L	Q_{An}	Q_{Gn}

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
Q_{A0}, Q_{B0}, Q_{H0} = level of Q_A, Q_B, Q_H before the indicated steady-state input conditions were established
Q_{An}, Q_{Gn} = level of Q_A or Q_G before the most recent ↑ transition of the clock; indicates a one-bit shift.

74164 Logic Schematic:



74165 (Shift Register)



Name:	**74165** (Parallel Load 8-Bit Shift Register)
Declaration:	74165(SI,A,B,C,D,E,F,G,H,SL,CKI,CK, QHN,QH)
EPLDs:	EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — SI, A, B, C, D, E, F, G, H, CKI, CK VCC — SL

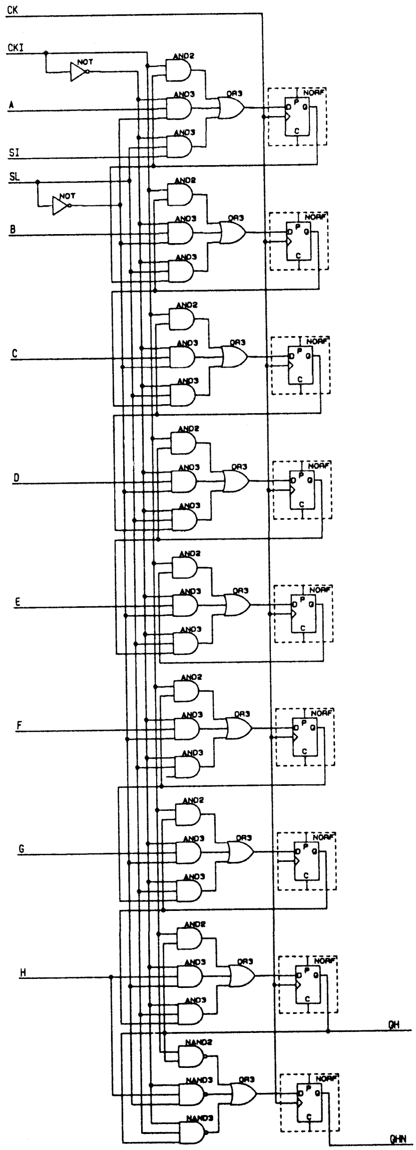
74165 Function Table:

74165 Function Table

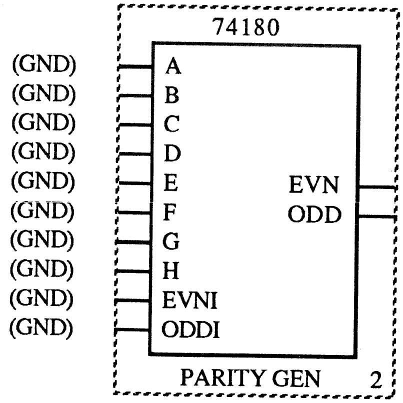
INPUTS					INTERNAL OUTPUTS		OUTPUT
CK	CKI	SL	SI	PARALLEL A ... H	Q_A	Q_B	Q_H
↑	L	L	X	a ... h	a	b	h
L	L	H	X	X	Q_{A0}	Q_{B0}	Q_{H0}
↑	L	H	H	X	H	Q_{An}	Q_{Gn}
↑	L	H	L	X	L	Q_{An}	Q_{Gn}
X	H	H	X	X	Q_{A0}	Q_{B0}	Q_{H0}

H = high level (steady state)
L = low level (steady state)
↑ = transition from low to high level
X = don't care (any input, including transitions)
a ... h = level of steady-state inputs at inputs A through H
Q_0 = level of Q before the indicated steady-state input conditions were established
Q_n = level of Q before the most recent active transition indicated by ↑

74165 Logic Schematic:



74180 (Parity Generator)



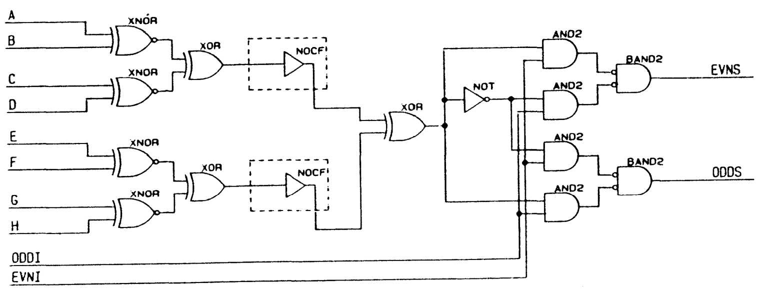
Name:	**74180** (9-Bit Odd/Even Parity Generator/ Checker)
Declaration:	74180(A,B,C,D,E,F,G,H,EVNI,ODDI,ODD, EVN)
EPLDs:	All
Default Signal Levels:	GND — all input pins

74180 Function Table:

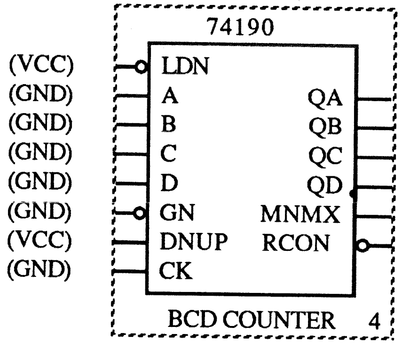
74180 Function Table

INPUTS			OUTPUTS		
Σ OF H's AT A THROUGH H	EVNI	ODDI	Σ EVNS	Σ ODDS	H = high level L = low level X = don't care
EVEN	H	L	H	L	
ODD	H	L	L	H	
EVEN	L	H	L	H	
ODD	L	H	H	L	
X	H	H	L	L	
X	L	L	H	H	

74180 Logic Schematic:



74190 (Counter)



Name:	**74190** (4-Bit Up/Down Decade Counter With Synchronous Load)
Declaration:	74190(GN,LDN,DNUP,A,B,C,D,CK,QD,QC, QB,QA,MNMX,RCON)
EPLDs:	All
Default Signal Levels:	GND — A, B, C, D, CK, GN VCC — LDN, DNUP

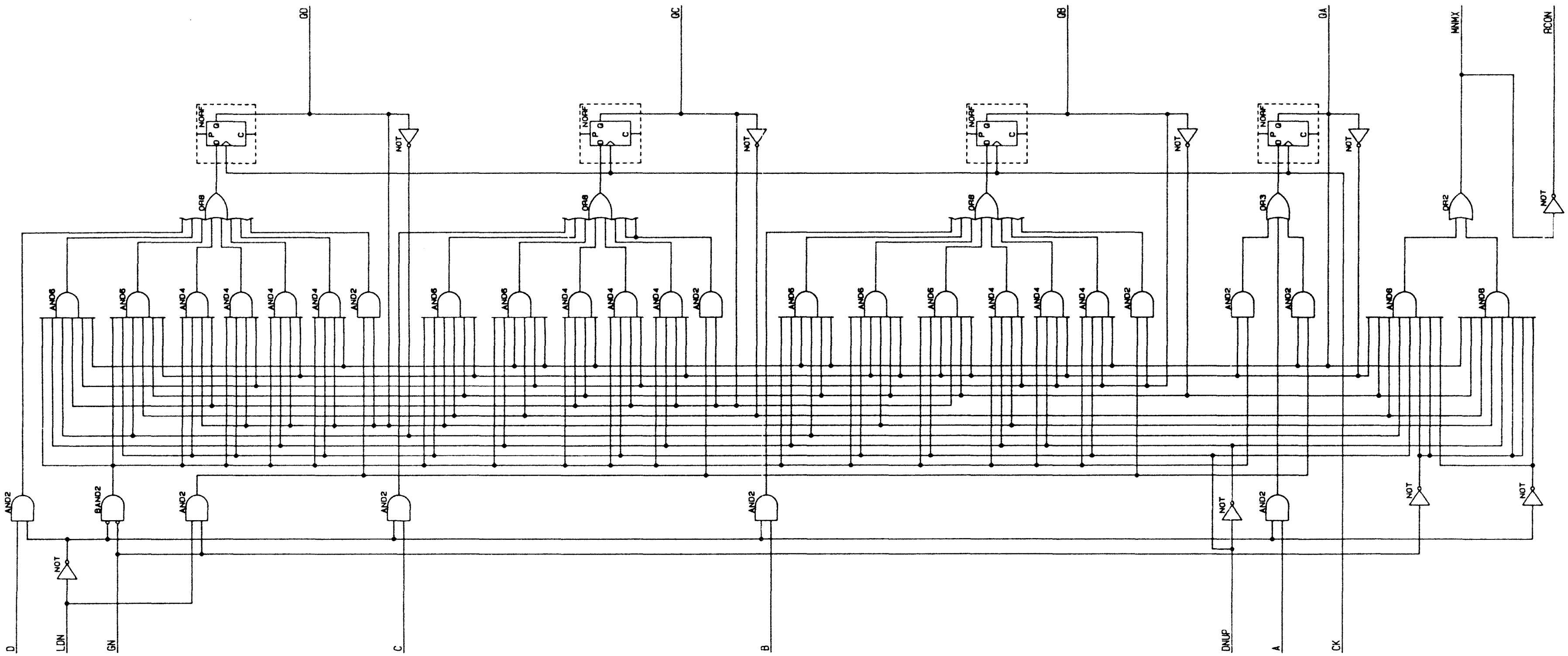
74190 Function Table:

74190 Function Table

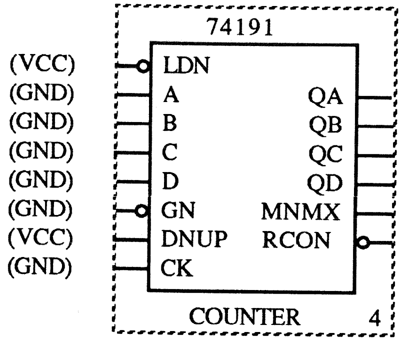
INPUTS								OUTPUTS					
CK	GN	LDN	DNUP	D	C	B	A	Q_D	Q_C	Q_B	Q_A	MNMX	RCON
↑	X	L	X	d	c	b	a	d	c	b	a	X	X
X	L	H	L					H	L	L	H	H	L
X	L	H	H					L	L	L	L	H	L
↑	L	H	L					COUNT UP				L	H
↑	L	H	H					COUNT DOWN				L	H
↑	H	H	X					HOLD COUNT				L	H

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

74190 Logic Schematic:



74191 (Counter)



Name:	**74191** (4-Bit Binary Up/Down Counter With Synchronous Load)
Declaration:	74191(GN,LDN,DNUP,A,B,C,D,CK,QD,QC, QB,QA,MNMX,RCON)
EPLDs:	All
Default Signal Levels:	GND — A, B, C, D, CK, GN VCC — LDN, DNUP

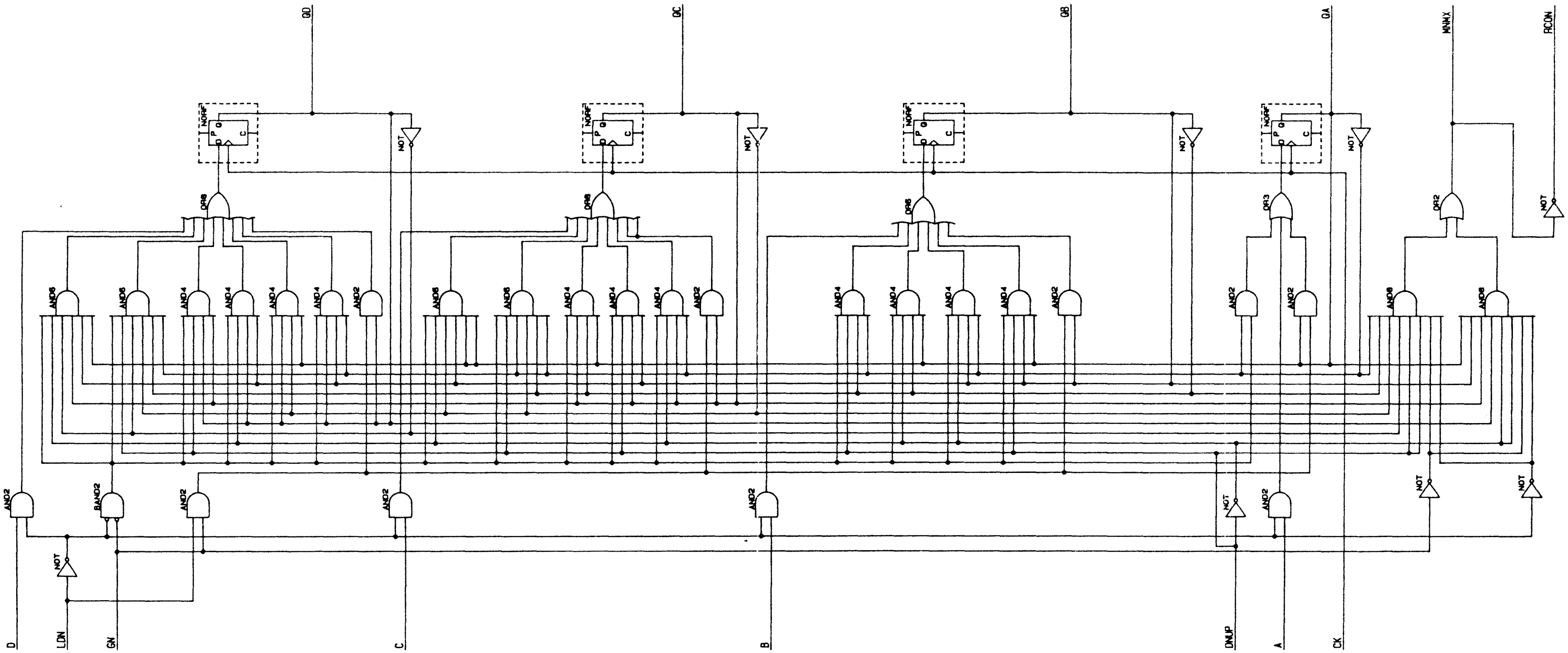
74191 Function Table:

74191 Function Table

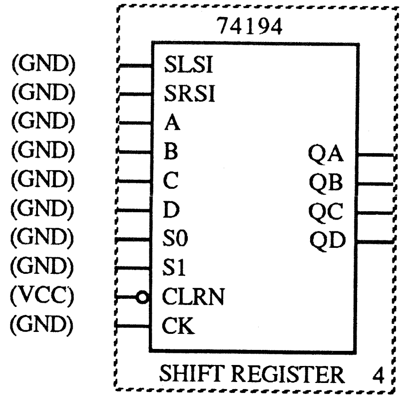
INPUTS								OUTPUTS					
CK	GN	LDN	DNUP	D	C	B	A	Q_D	Q_C	Q_B	Q_A	MNMX	RCON
↑	X	L	X	d	c	b	a	d	c	b	a	X	X
X	L	H	L					H	H	H	H	H	L
X	L	H	H					L	L	L	L	H	L
↑	L	H	L					COUNT UP				L	H
↑	L	H	H					COUNT DOWN				L	H
↑	H	H	X					HOLD COUNT				L	H

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
↑ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D

74191 Logic Schematic:



74194 (Shift Register)



Name: **74194** (4-Bit Bi-Directional Shift Register With Parallel Load)

Declaration: 74194(SLSI,SRSI,A,B,C,D,S0,S1,CLRN,CK, QD,QC,QB,QA)

EPLDs: EP310, EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400

Default Signal Levels: GND — SLSI,SRSI, A, B, C, D,S0, S1, CK
VCC — CLRN

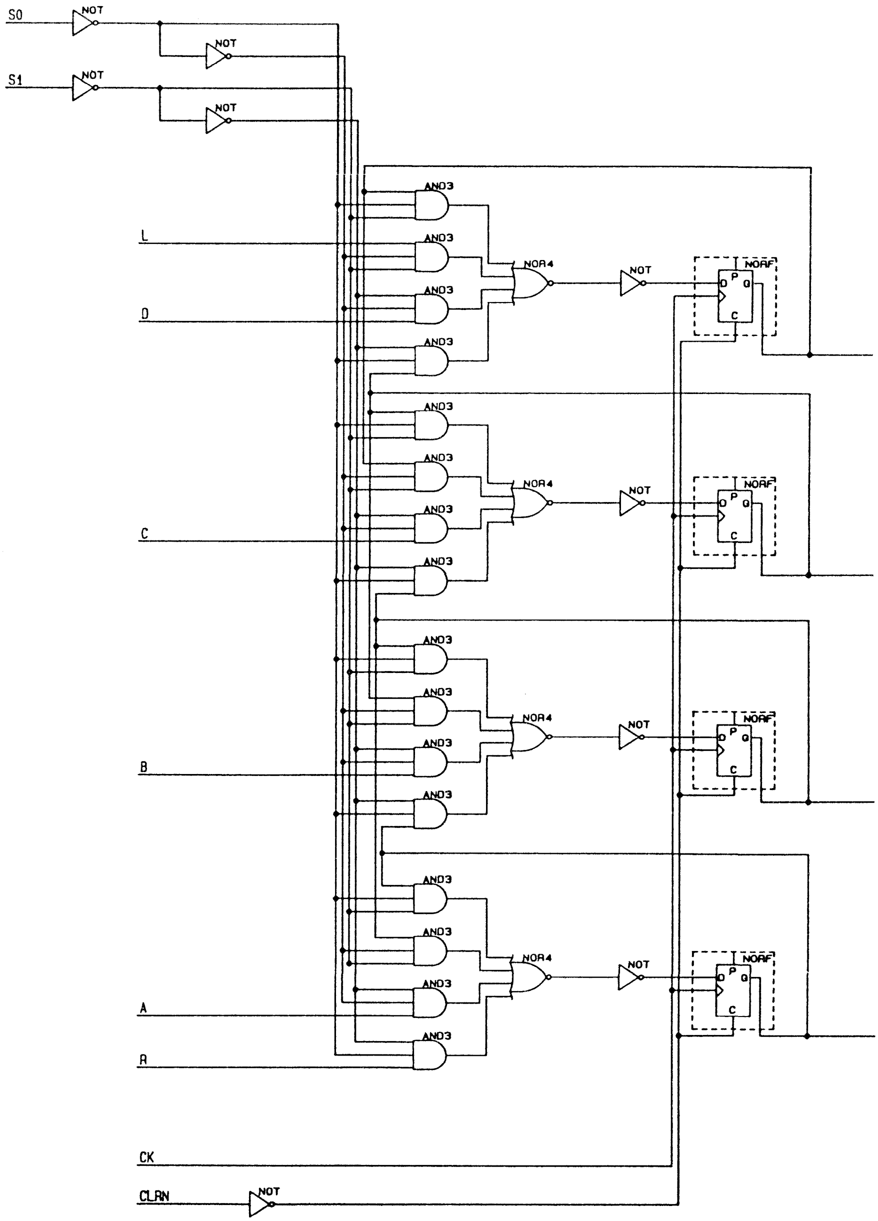
74194 Function Table:

74194 Function Table

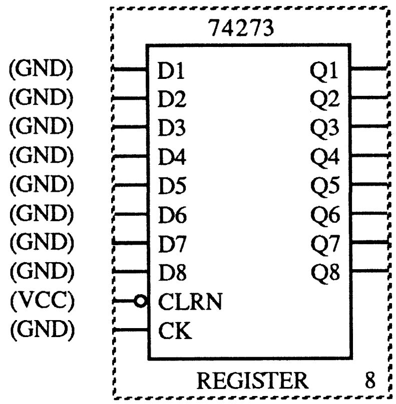
INPUTS										OUTPUTS			
CLEAR	MODE		CLOCK	SERIAL		PARALLEL				Q_A	Q_B	Q_C	Q_D
	S1	S0		L	R	A	B	C	D				
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	L	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
H	H	H	$\uparrow$	X	X	a	b	c	d	a	b	c	d
H	L	H	$\uparrow$	X	H	X	X	X	X	H	Q_{An}	Q_{Bn}	Q_{Cn}
H	L	H	$\uparrow$	X	L	X	X	X	X	L	Q_{An}	Q_{Bn}	Q_{Cn}
H	H	L	$\uparrow$	H	X	X	X	X	X	Q_{Bn}	Q_{Cn}	Q_{Dn}	H
H	H	L	$\uparrow$	L	X	X	X	X	X	Q_{Bn}	Q_{Cn}	Q_{Dn}	L
H	L	L	$\uparrow$	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}

H = high level (steady state)
L = low level (steady state)
X = don't care (any input including transitions)
$\uparrow$ = transition from low to high level
a,b,c,d, = level of steady state input at inputs A,B,C,D
Q_{A0} Q_{B0} Q_{C0} Q_{D0} = level of Q_A Q_B Q_C Q_D before the indicated steady-state input conditions were established
Q_{An} Q_{Bn} Q_{Cn} Q_{Dn} = level of Q_A Q_B Q_C Q_D before the most recent $\uparrow$ transition of the clock

74194 Logic Schematic:



74273 (Register)



Name:	**74273** (Octal D-Type Flipflop With Asynchronous Clear)
Declaration:	74273(D1,D2,D3,D4,D5,D6,D7,D8,CLRN, CK,Q8,Q7,Q6,Q5,Q4,Q3,Q2,Q1)
EPLDs:	EP310, EP600, EP610, EP900, EP910, EP1210, EP1800, EPB1400
Default Signal Levels:	GND — D1, D2, D3, D4, D5, D6,D7, D8, CK VCC — CLRN

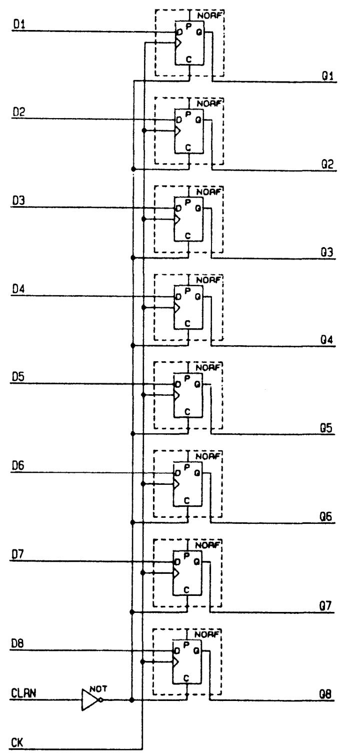
74273 Function Table:

74273 **Function Table (each flip-flop)**

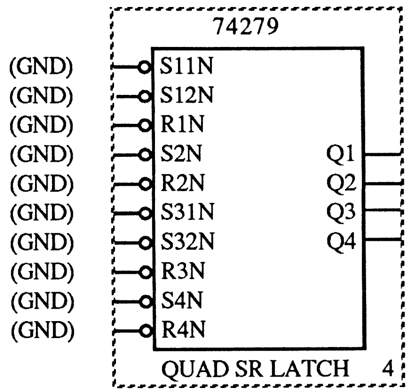
INPUTS			OUTPUT
CLEAR	CLOCK	D	Q
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	Q_0

H = high level (steady state)
L = low level (steady state)
X = don't care
Q_0 = level of Q before the indicated steady-state input conditions were established
↑ = transition from low to high

74273 Logic Schematic:



74279 (Latch)



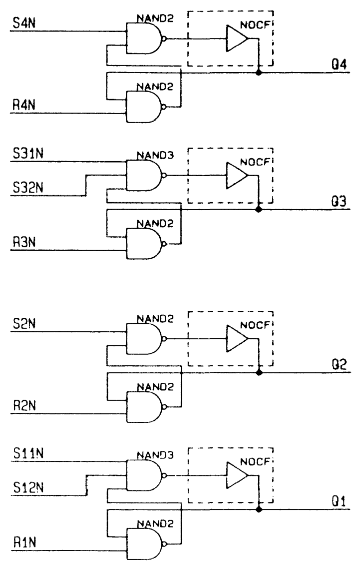
Name:	**74279** (Quad /S-/R Latch)
Declaration:	74279(S11N,S12N,R1N,S2N,R2N,S31N, S32N,R3N,S4N,R4N,Q4,Q3,Q2,Q1)
EPLDs:	All
Default Signal Levels:	GND— all input pins

74279 Function Table:

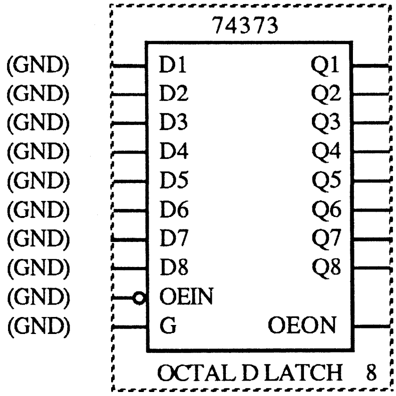
74279 Function Table

INPUTS $\overline{SN}$	$\overline{RN}$	OUTPUT Q	
H	H	Q_0*	H = high level (steady state)
L	H	H	L = low level (steady state)
H	L	L	Q_0 = level of Q before the indicated input conditions were established
L	L	H**	* This output level is pseudo stable.
			** For latches with double $\overline{S}$ inputs: H = both $\overline{SN}$ inputs high; L = one or both $\overline{SN}$ inputs low

74279 Logic Schematic:



74373 (Latch)



Name: **74373** (Transparent Octal D-Type Latch With Output Enable)

Declaration: 74373(D1,D2,D3,D4,D5,D6,D7,D8,OEIN,G, OEON,Q8,Q7,Q6,Q5,Q4,Q3,Q2,Q1)

EPLDs: All

Default Signal Levels: GND — all input pins

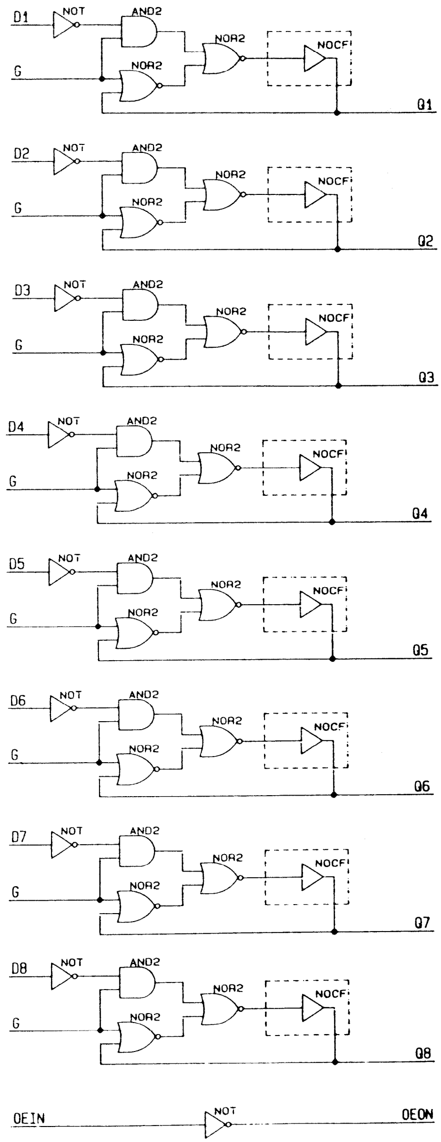
Function Table:

74373 Function Table

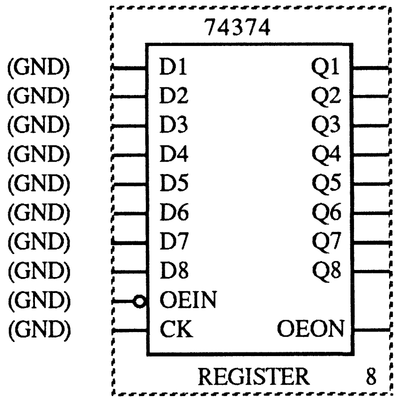
INPUTS			OUTPUTS	
OEIN	G	D	Q	OEON
L	X	X	X	H
H	X	X	X	L
X	H	L	L	X
X	H	H	H	X
X	L	X	Q_0	X

H = high level (steady state)
L = low level (steady state)
X = don't care
Q_0 = level of Q before the indicated steady-state input conditions were established

74373 Logic Schematic:



74374 (Register)



Name:	**74374** (Octal D-Type Flipflop With Output Enable)
Declaration:	74374(D1,D2,D3,D4,D5,D6,D7,D8,OEIN, CK,OEON,Q8,Q7,Q6,Q5,Q4,Q3,Q2,Q1)
EPLDs:	All
Default Signal Levels:	GND — all input pins

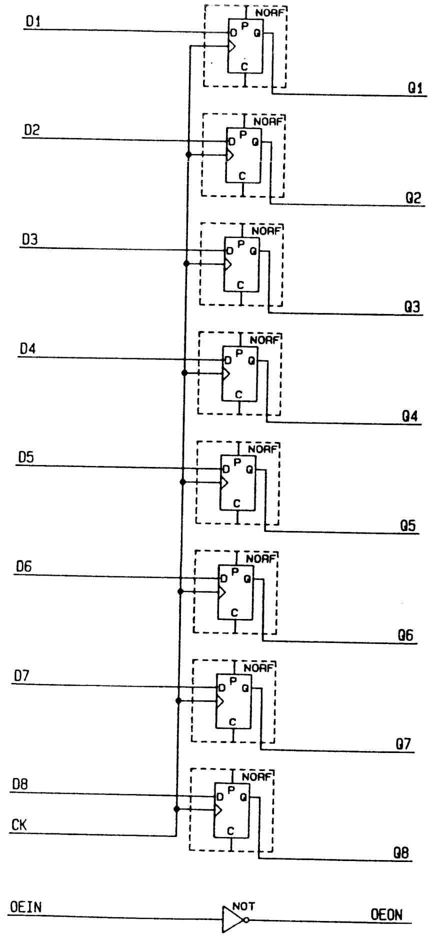
Function Table:

74374 Function Table

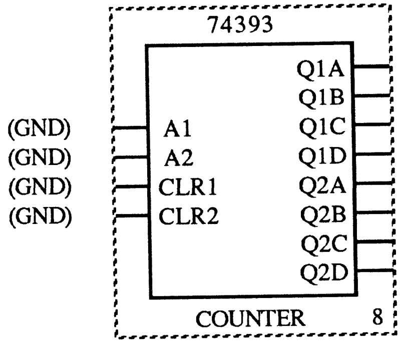
INPUTS			OUTPUTS	
CK	OEIN	D	Q	OEON
X	L	X	X	H
X	H	X	X	L
↑	X	L	L	X
↑	X	H	H	X
L	X	X	Q_0	X

H = high level (steady state)
L = low level (steady state)
X = don't care
Q_0 = level of Q before the indicated steady-state input conditions were established
↑ = transition from low to high level

74374 Logic Schematic:



74393 (Counter)



Name: **74393** (4-Bit Up Counter With Asynchronous Clear)

Declaration: 74393(CLR1,CLR2,A1,A2,Q2D,Q2C,Q2B,Q2A,Q1D,Q1C,Q1B,Q1A)

EPLDs: EP600, EP610, EP900, EP910, EP1800, EPB1400

Default Signal Level: GND — all input pins

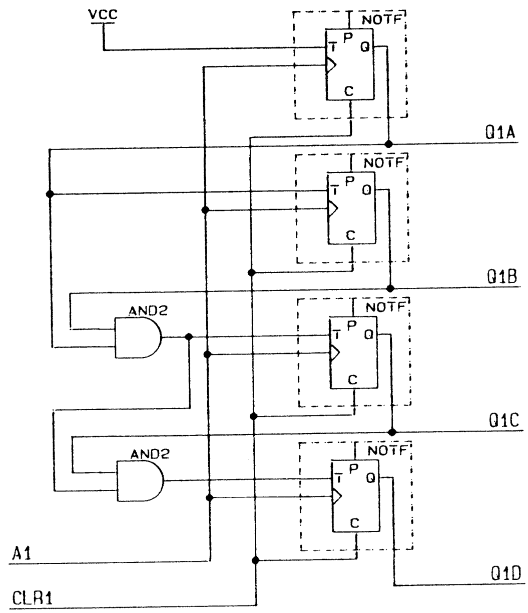
74393 Function Table:

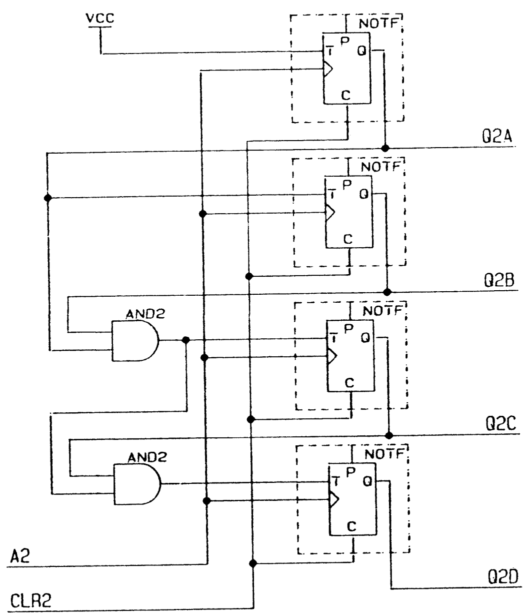
74393 Function Table

COUNT	INPUTS		OUTPUT			
	A	CLR	Q_D	Q_C	Q_B	Q_A
	X	H	L	L	L	L
0	↑	L	L	L	L	L
1	↑	L	L	L	L	H
2	↑	L	L	L	H	L
3	↑	L	L	L	H	H
4	↑	L	L	H	L	L
5	↑	L	L	H	L	H
6	↑	L	L	H	H	L
7	↑	L	L	H	H	H
8	↑	L	H	L	L	L
9	↑	L	H	L	L	H
10	↑	L	H	L	H	L
11	↑	L	H	L	H	H
12	↑	L	H	H	L	L
13	↑	L	H	H	L	H
14	↑	L	H	H	H	L
15	↑	L	H	H	H	H

H = high level
L = low level
↑ = transition from low to high level

74393 Logic Schematic:





LogiCaps Messages

All LogiCaps messages are listed alphabetically below. Each message is accompanied by an explanation and suggestions for corrective action. All error messages are prefixed with **ERROR-**.

Area conflict

CAUSE: You tried to move or copy an area to a location that already contains one or more objects (symbols, lines, and/or text). An area may not be moved or copied to a location where it overlaps with another symbol or text field.

ACTION: Position the area at a location where there is no obstruction.

Area not defined

CAUSE: An area is only defined when the area boundary is visible. You tried to move, copy, or delete an area that is undefined, i.e., the area boundary lines are invisible.

ACTION: Use the **AT** command to display the boundary lines or the **AB** command to define a new area.

Closing macro file

CAUSE: The macro file could not be closed because of a disk problem.
ACTION: Try writing the file to another disk.

Dot array full

CAUSE: The array for connection dots has reached its capacity. You cannot enter any more dots into your drawing.
ACTION: You may want to divide your drawing into two or more smaller drawings. The size of this array is a function of the available memory. Saving the drawing with the **DW** command may make additional memory available. Or you may restart LogiCaps with the **–ml** option to expand the amount of memory allocated for lines and dots.

Excessive line segmentation

CAUSE: LogiCaps is unable to complete a line because it is broken up by too many connection dots.
ACTION: You may have to remove some of the dots or draw the line in smaller segments.

File not found or error opening

CAUSE: The program couldn't find the file you tried to open with the **TI** (Text Import) command.
ACTION: Make sure you have specified the correct path name.

File read error

CAUSE: The file cannot be read. It may be corrupted or there may be a disk problem.
ACTION: Try using a backup copy of the file.

File write error

CAUSE: The file cannot be written to disk. It may be corrupted or there may be a disk problem.
ACTION: Try writing the file to another disk.

Hline array full

CAUSE: The array for horizontal lines has reached its capacity. You cannot enter any more horizontal lines into your drawing.

ACTION: You may want to divide your drawing into two or more smaller drawings. The size of this array is a function of the available memory. Saving the drawing with the **DW** command may make additional memory available. Or you may restart LogiCaps with the **–ml** option to expand the amount of memory allocated for lines and dots.

Insufficient memory

CAUSE: You don't have enough system memory. LogiCaps cannot be opened.

ACTION: Make sure that you have at least 512 K of available memory. Also, eliminate resident programs that use up memory, such as copies of A+PLUS that you may have exited via the DOS command rather than the **QUIT** command.

Invalid entry

CAUSE: You typed an unexpected entry on the command line.

ACTION: A filename or some numeric parameters are expected.

Invalid EQN symbol

CAUSE: You entered a non-standard EQN symbol.

ACTION: Use the EQN symbol provided by the Altera symbol library.

Invalid file designation

CAUSE: You used the **SL** (Symbol Library) command and entered an invalid filename.

ACTION: Type **SL** (Symbol Library) to open the default symbol library.

Invalid grid specification

CAUSE: You have entered the **WG** (Window Grid) command and have specified a value that is not supported by LogiCaps.

ACTION: Refer to the **WG** command description for valid grid values.

Invalid key for macro

CAUSE: Macros may be assigned to function keys <**F1**> to <**F10**> only. You tried to assign a macro a key other than a function key.

ACTION: Re-enter the **MA** command and assign the macro to a valid key.

Invalid size specification

CAUSE: You have entered the **DS** (Drawing Size) command and have specified a size that is not supported by LogiCaps.

ACTION: Refer to the **DS** command description for valid drawing sizes.

Memory allocation error

CAUSE: You don't have enough system memory or the system memory has been corrupted by other software. LogiCaps cannot be opened.

ACTION: Make sure that you have at least 512 K of available memory. Also, eliminate resident programs that use up memory, such as copies of A+PLUS that you may have exited via the DOS command rather than the **QUIT** command.

No files found matching

CAUSE: You entered an invalid DOS path or no files were found in the specified directory.

ACTION: Make sure that the specified path and directory exist.

Node (x) tied to node (y)

CAUSE: You have assigned multiple text fields to one wire (node). Each node may have only one text field associated with it.

ACTION: Refer to the description of the Text Menu in the Reference Section for proper text field positioning.

Node too complex to trace

CAUSE: The wire in your design has too many branches.

ACTION: Reduce the number of branches.

Non-standard file

CAUSE: You have specified a file that does not conform to LogiCaps format (i.e., it is not a valid .SD or .SDA file). Possibly your file is corrupted.
ACTION: Try to load a backup copy of the file.

Not recording macro

CAUSE: You have entered the **MS** (Macro Stop) command but have not previously used the **MR** (Macro Record) command to start recording a macro.
ACTION: You must be in recording mode to be able to stop a macro.

Opening ADF file

CAUSE: The ADF cannot be opened because of a disk problem.
ACTION: Check to be sure that the disk is not write-protected, that it has enough memory left, and/or that it is in the correct disk drive. LogiCaps writes the ADF to the path from which you load the drawing.

Opening drawing file

CAUSE: The filename you entered cannot be found.
ACTION: Be sure to provide a valid path name. Use the **DF** (Drawing Files) or **AF** (Area Files) command to get a directory of drawing files.

Reading symbol

CAUSE: The symbol library cannot be accessed. The file has been corrupted.
ACTION: Try using a copy from your backup disk.

Rubberbanding buffer full

CAUSE: The array for rubberbanding lines has reached its capacity. Some lines connected to the area you are moving will be left in their original place.
ACTION: You will have to reconnect these lines manually after completing the move.

Rubberbanding problem

CAUSE: You may have inadvertently edited a symbol and added too many pins to it. You should not edit the symbols provided in the LogiCaps symbol library.

ACTION: Delete the corrupted symbol and replace it by using the **SE** (Symbol Enter) command.

Symbol array full

CAUSE: The array for symbols has reached its capacity. You cannot enter any more symbols into your drawing.

ACTION: You may want to divide your drawing into two or more smaller drawings. The size of this array is a function of the available memory. Saving the drawing with the **DW** command may make additional memory available. Or you may restart LogiCaps with the **–ms** option to expand the amount of memory allocated for symbols.

Symbol crossing area boundary

CAUSE: You used the **AW** (Area Write) command but the area boundary crosses through a symbol.

ACTION: Move the boundary or reposition the symbol.

Symbol definition array full

CAUSE: The array for symbol definitions has reached its capacity. You cannot enter any more symbols into your drawing.

ACTION: You may want to divide your drawing into two or more smaller drawings. The size of this array is a function of the available memory. Saving the drawing with the **DW** command may make additional memory available. Or you may restart LogiCaps with the **–ms** option to expand the amount of memory allocated for symbols.

Symbol exceeds drawing size limits

CAUSE: You entered a symbol whose right or bottom side extends beyond the drawing boundaries.

ACTION: Reposition the symbol or increase the drawing size.

Symbol library not open

CAUSE: The library you tried to open was not found in the current DOS directory.

ACTION: Make sure ALTERA.SYM is available in your directory. Use the **SL** (Symbol Library) command to open a valid library.

Symbol not in library

CAUSE: You have entered the **SE** (Symbol Enter) command and have entered an invalid symbol name.

ACTION: Check your command line entry and type a valid symbol name. Use the **SL** (Symbol Library) command to see the list of available symbols. Note that the symbol library may not have been found when you started the program. The library ALTERA.SYM must be present in the default path when LogiCaps is invoked.

Symbol overlap

CAUSE: You have tried to enter or copy a symbol at a location that already contains a symbol.

ACTION: Check to be sure the symbol boundary borders do not overlap.

Temporary line buffer full

CAUSE: The array for temporary rubberbanding lines has reached its capacity. Temporary rubberbanding lines are produced as a result of conflicts when LogiCaps is re-inserting rubberbanding lines after a symbol or area move.

ACTION: You should delete temporary lines right after you execute the rubberbanding function and replace them with solid lines to empty the temporary line buffer.

Text array full

CAUSE: The array for text entries has reached its capacity. You cannot enter any more text into your drawing.

ACTION: You may want to divide your drawing into two or more smaller drawings. The size of this array is a function of the available memory. Saving the drawing with the **DW** command may make additional memory available. Or you may restart LogiCaps with the **–mt** option to expand the amount of memory allocated for text.

Text at drawing size limit

CAUSE: You used the **TE** (Text Enter) command but the text field extends beyond the drawing boundaries.

ACTION: Reposition the text field or increase the drawing size.

Text field array full

CAUSE: The array for text fields has reached its capacity. You cannot enter any more text into your drawing.

ACTION: You may want to divide your drawing into two or more smaller drawings. The size of this array is a function of the available memory. Saving the drawing with the **DW** command may make additional memory available. Or you may restart LogiCaps with the **–mt** option to expand the amount of memory allocated for text.

Text string too long

CAUSE: A single text string may be up to 250 characters long.

ACTION: Be sure that you don't exceed that limit when you use the **TI** (Text Import) command.

Unable to open file for write

CAUSE: You have entered the **DW** (Drawing Write) or **AW** (Area Write) command and the file cannot be opened.

ACTION: The disk may be write-protected, or you have too many files on the disk.

Unable to open HELP file

CAUSE: LogiCaps searches only the current directory for the help file.

ACTION: Be sure the files with the .HLP extensions are in the current directory if you wish to use the Help commands.

Unable to open macro file

CAUSE: The macro file cannot be opened because of a disk problem.

ACTION: Check to be sure that the disk is not write-protected, that it has enough memory left, and/or that it is in the correct disk drive. You may also verify that you have specified the correct path.

Unable to open symbol library

CAUSE: You used the **SL** (Symbol Library) command but the entered library name was not found.

ACTION: Be sure to use a valid pathname.

Vline array full

CAUSE: The array for vertical lines has reached its capacity. You cannot enter any more vertical lines into your drawing.

ACTION: You may want to divide your drawing into two or more smaller drawings. The size of this array is a function of the available memory. Saving the drawing with the **DW** command may make additional memory available. Or you may restart LogiCaps with the **–ml** option to expand the amount of memory allocated for lines and dots.

Writing ADF file

CAUSE: The ADF cannot be generated because of a disk problem.

ACTION: Check to make sure your disk has sufficient memory or try to write the ADF to another disk.

Writing macro file

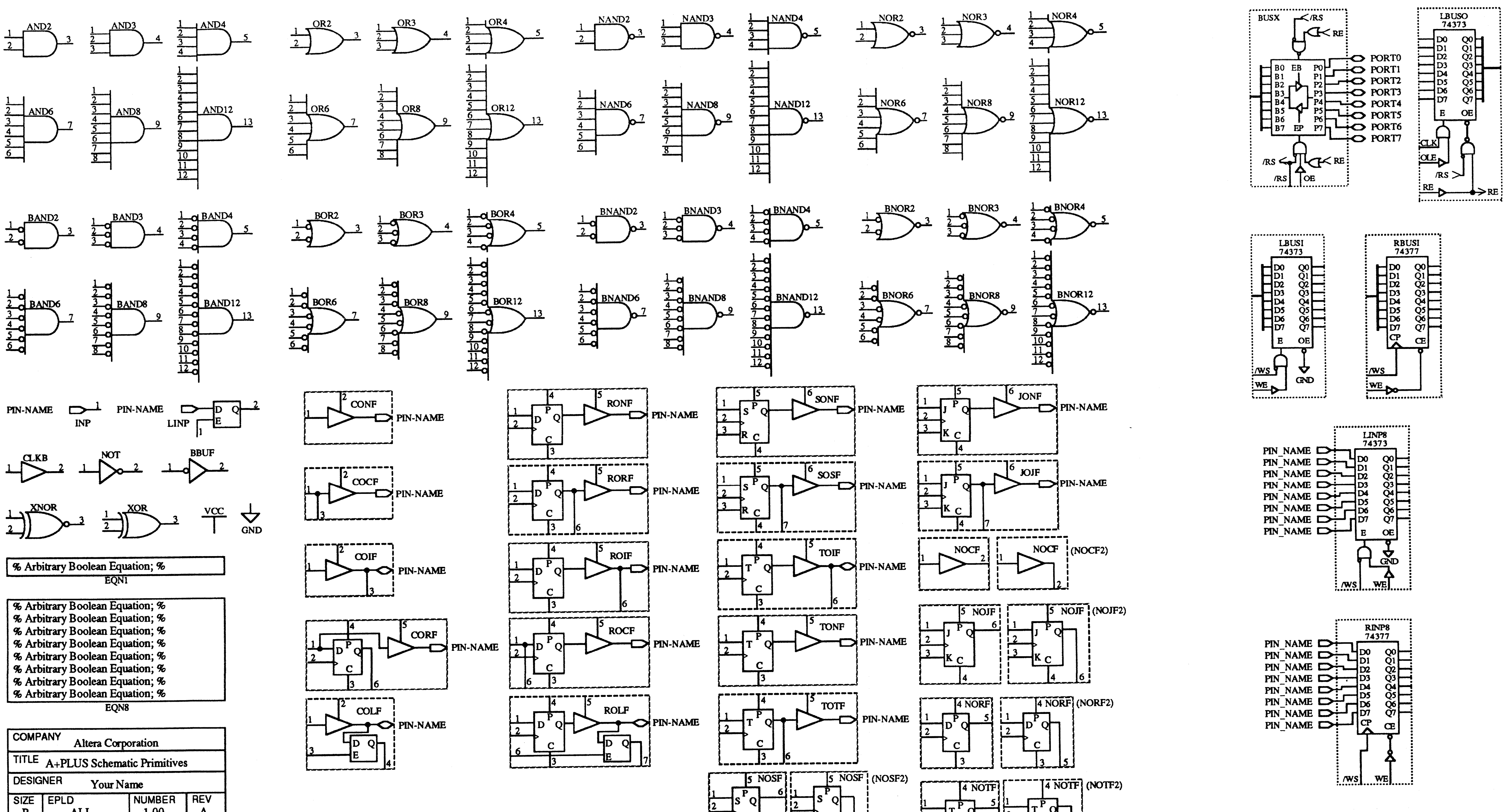
CAUSE: The macro file cannot be written to disk because of a disk problem.

ACTION: Check to be sure that the disk is not write-protected, that it has enough memory left, and/or that it is in the correct disk drive. You may also verify that you have specified the correct path.

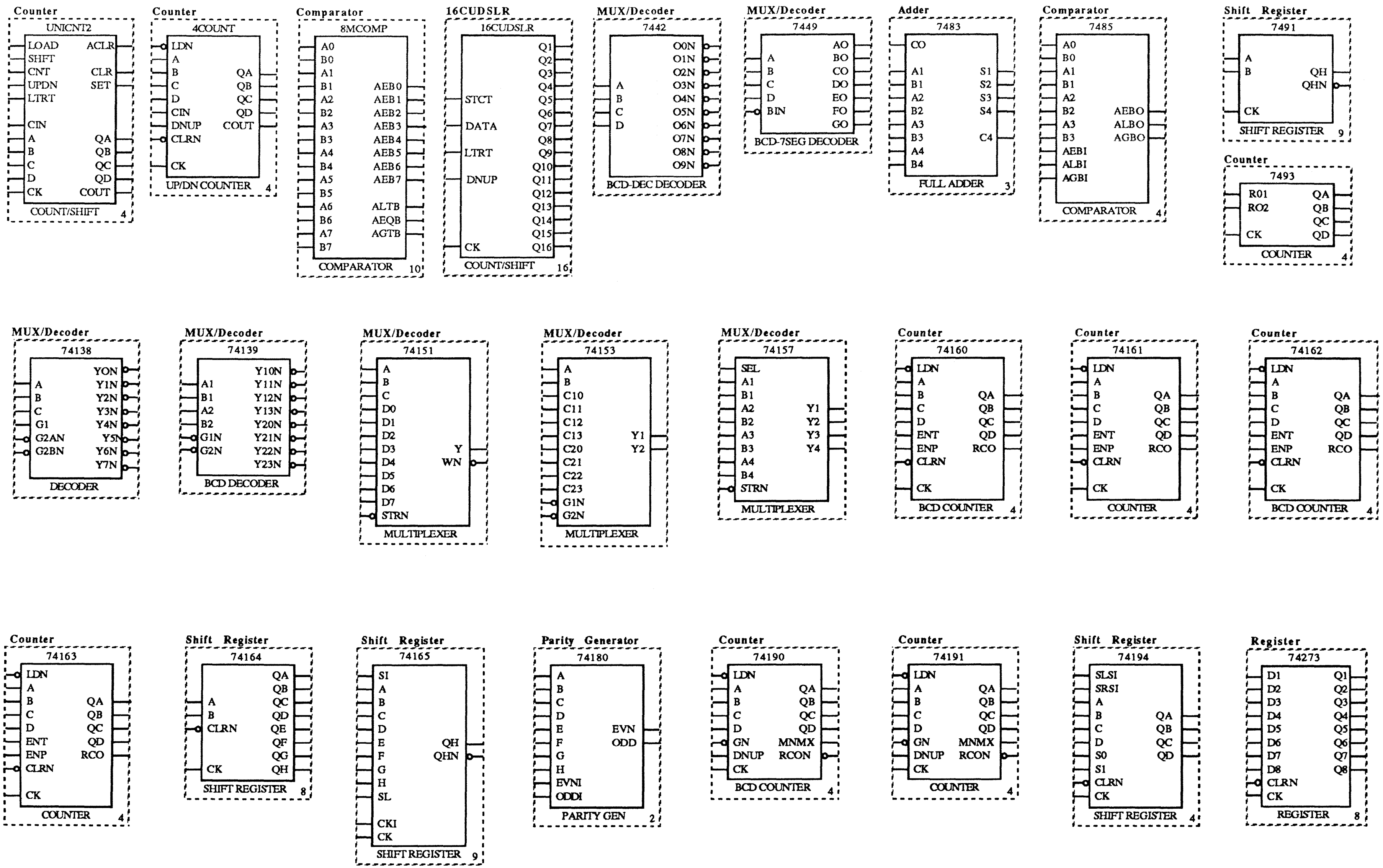
APPENDIX A

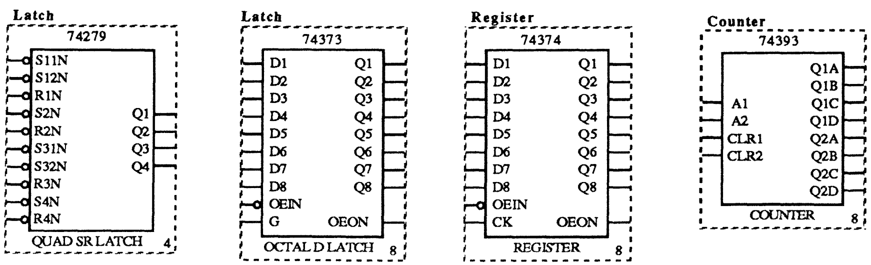
Design Library

The following foldout pages show the LogiCaps design library containing the Altera primitives and standard MacroFunctions. When you unfold the pages, you may flip to other pages in the manual while still having this library visible.



Altera Primitives





Standard MacroFunctions

APPENDIX B

Command Summary

The following foldout page shows a complete list of the available LogiCaps commands. When you unfold the page, you can flip to other pages in the manual while always having this summary visible.

LogiCaps Command Summary

Invoking LogiCaps

LOGICAPS [filename] [options] <Enter>

The HELP Command

H	Display information about the use of LogiCaps

The QUIT Command

Q	Exit LogiCaps

The Area Menu

AB	Define area boundary
AC	Copy an area
AD	Delete an area
AF	Display list of area files
AH	Display area help information
AL	Load an area file
AM	Move an area
AT	Toggle an area ON or OFF
AW	Save area and write to file

The Macro Menu

MA	Assign a function key to macro
MC	Clear the function key from the macro
MF	Display list of macro files
MH	Display macro help information
MP	Play back a recorded macro
MR	Record and save a macro
MS	Stop recording a macro

Command Line Editing

<LeftArrow>	Move text cursor left
<RightArrow>	Move text cursor right
<Ins>	Toggle character insertion mode
<Del>	Delete character at text cursor
<Backspace>	Delete character left of text cursor
<End>	Move to end of text
<Home>	Move to beg. of text
<Ctrl><LeftArrow>	Move cursor to end of previous word
<Ctrl><RightArrow>	Move cursor to beg. of next word
<Ctrl><End>	Move cursor to the right to the nearest pin number

The Drawing Menu

DA	Generate an ADF
DD	Delete a drawing
DF	Display list of drawing files
DH	Display drawing help information
DL	Load a drawing
DS	Select drawing size
DW	Save a drawing and write to file

The Symbol Menu

SC	Copy a symbol
SD	Delete a symbol
SE	Enter a symbol
SF	Find a specified symbol
SH	Display symbol help information
SL	Change symbol library
SM	Move a symbol
SN	Display/hide symbol reference numbers
SR	Reflect (flip) a symbol

Mouse Commands

<Left>	Move objects Draw lines
<Center>	Copy objects Connect lines
<Right>	Cancel move and copy commands Delete lines

The Line Menu

LB	Toggle the routing of a line
LC	Copy a line
LD	Delete a line
LE	Enter a line
LH	Display line help information
LJ	Enter a connection dot (join lines)
LM	Move a line
LR	Enable/disable rubberbanding
LS	Select line type

The Text Menu

TB	Display/hide text field borders
TC	Copy text in text field
TD	Delete text
TE	Enter text
TF	Search for specified text
TH	Display text help information
TI	Import text to current cursor location
TM	Move text field
TS	Select text type

The Function Keys

<F1>	Recall a window view
<F2>	Recall a window view
<F3>	Select line type
<F4>	Select character type
<F5>	Toggle area boundary
<F6>	Define area boundary
<F7>	Enter text
<F8>	Enter a symbol
<F9>	Toggle grid display
<F10>	Toggle window split
<Tab>	Leap to previously saved location
<Home>	Home the cursor
<+>	Enter cursor MKY mode
<PgUp>	Zoom out
<PgDn>	Zoom in
<End>	Go directly to full scale or maximum reduced scale
<*>	Redisplay previous message
<Space>	Refresh the screen
<Esc>	Cancel a command
<Shift><F10 >	Stop recording a macro
<'<'>	Move window split to left
<'>'>	Move window split to right

The Window Menu

WA	Enable/disable autopanning
WC	Select window color
WG	Set grid spacing
WH	Display window help information
WM	Move to specified cursor location
WP	Put cursor in PAN mode
WR	Recall saved window
WS	Split the window display
WT	Save the window and assign to <F1> or <F2>

APPENDIX C

Plotter and Printer Support

LogiCaps offers plotter and printer support for your schematic design. Following are descriptions and instructions for using both.

Plotting the Circuit Design

LogiCaps supports the HP7440/GE, 7475, 7550, 7570, 7580, 7585 drafting plotters.

Invoking LCPLOT

To invoke LCPLOT, type from DOS:

lcplot <options> <filename> <Enter>

Available Options

The following options are available (defaults are given in parentheses):

–b	Plot symbol number and border (OFF).
–m	Maximize drawing to sheet (OFF).
–o filename	Plot to file, filename (OFF).
–p [12]	Plot to port 1 or 2 (port 1).
–q [0..4]	Plot in the specified quadrant. Numbering of quadrants follows the Cartesian coordinate system (0 = center of drawing sheet).
–s	Plot sheet border (OFF).
–#	Plot symbol pin numbers (OFF). Pin numbers assigned to pin names with the @ symbol will not be printed.
–fa [1..4]	Pen acceleration (system default).
–ff [1..8]	Pen force (system default).
–fp [1..8]	Pen number (pen 1).
–fv [1..60]	Pen velocity (system default).

Available Drawing Sizes

Following are the dimensions of plotted drawings in inches and database units (DBUs):

Size A	8.5 x 11 in.	85 x 110 DBU
Size AV	11 x 8.5 in.	110 x 85 DBU
Size B	17 x 11 in.	170 x 110 DBU
Size BV	11 x 17 in.	110 x 170 DBU
Size C	22 x 17 in.	220 x 170 DBU
Size CV	17 x 22 in.	170 x 220 DBU
Size D	34 x 22 in.	340 x 220 DBU
Size DV	22 x 34 in.	220 x 340 DBU
Size E	44 x 34 in.	440 x 340 DBU
Size EV	34 x 44 in.	340 x 440 DBU
Size F	90 x 90 in.	900 x 900 DBU

Examples

1. **lcplot –p2 –q3 –# <filename> <Enter>**

   (The specified schematic is plotted to serial port 2 in quadrant 3 of the plotting paper with the symbol pin numbers included.)

2. **lcplot <Enter>**

   (A list of the available options is displayed.)

Baud Rate Settings

When LCPLOT is invoked, your systems's serial port is automatically set to 9600 baud, 8 data bits, 1 stop bit, and no parity regardless of what your current setting is. Once plotting is completed, the system is switched back to the original baud setting.

Printing the Circuit Design

LogiCaps supports the Epson FX and MX series of printers as well as compatibles.

Invoking LCPRINT

To invoke LCPRINT, type from DOS:

lcprint <options> <filename> <Enter>

Available Options

The following options are available (defaults are given in parentheses):

–b	Print symbol numbers and borders (OFF).
–e	Expand the scale to approximately 1:1.
–f	Fast print (OFF). This command prints the design at twice the speed but with about half the resolution.
–g	Print grid lines (OFF).
–m	Maximize printing area (OFF). This command positions the printed design close to the upper left corner of the sheet to reduce white space around the design.
–o name	Print to file or device (LPT1:).
–r	Reduce scale (OFF). This command reduces the size of the printed design by 50 %. A second **–r** command in the options string would reduce the design by 75 %.
–s	Print sheet border (OFF).
–w	Adjusts the setting to accommodate wide-carriage printers, e.g., Epson FX100 (OFF).
–#	Print symbol pin numbers (OFF). Pin numbers assigned to pin names with the @ symbol will not be printed.

Available Drawing Sizes

Unlike a plotted design, a printed design will not correspond to the sheet size specified in the title block because the print size depends on the resolution of the printer. Printed designs are generally smaller. For example, an Epson FX100 printer prints 15 DBUs per inch horizontally and 18 DBUs per inch vertically.

The following list shows the dimensions of drawings printed on the Epson FX100. These dimensions may vary on other printers.

Size A	4.7 x 7.3 in.	85 x 110 DBU
Size AV	6.1 x 5.6 in.	110 x 85 DBU
Size B	9.4 x 7.3 in.	170 x 110 DBU
Size BV	6.1 x 11.3 in.	110 x 170 DBU
Size C	12.2 x 11.3 in.	220 x 170 DBU
Size CV	9.4 x 14.6 in.	170 x 220 DBU
Size D	18.8 x 14.6 in.	340 x 220 DBU
Size DV	12.2 x 22.6 in.	220 x 340 DBU
Size E	24.4 x 22.6 in.	440 x 340 DBU
Size EV	18.8 x 29.3 in.	340 x 440 DBU
Size F	50 x 60 in.	900 x 900 DBU

Examples

1. **lcprint –o LPT2: –r –# –f <filename><Enter>**

   (The schematic is printed at accelerated speed to the second parallel printer at 50 % reduction with the symbol pin numbers included.)

2. **lcprint <Enter>**

   (A list of available options is displayed.)

Glossary

This glossary defines terms that are specific to LogiCaps. For additional terms that are specific to A+PLUS, you should consult the glossary in the ***A+PLUS Reference Guide***.

.ADF (Altera Design File) The filename extension assigned automatically to an Altera Design File generated from the schematic. This file is entered into the Altera Design Processor (ADP) for further processing and eventual programming of the EPLD.

.LIB (MacroFunction library file) A file recognized by the ADP as containing custom-made MacroFunctions. The file may not be named MACRO, since this name is reserved for Altera's MacroFunction library file MACRO.LIB.

.SD (Symbol Drawing file) A filename extension used in LogiCaps. It is assigned automatically to a schematic when it is saved.

.SDA (Symbol Drawing Area file) A filename extension used in LogiCaps. It is assigned automatically to an area (including MacroFunctions) when it is saved.

Active Window That portion of the split screen display into which entries may be made. When you are in Window Split mode, one portion (indicated by a solid borderline) of the screen is active, the other (indicated by a dashed borderline) is inactive. You can edit only the active window; however, all changes made are reflected in the inactive window. You may toggle a window from active to inactive.

ADLIB (Altera Design Librarian) An Altera software program that enables you to create your own MacroFunctions. It takes the data of a Logic Equation File generated by the Altera Design Processor and converts these data into a MacroFunction Symbol Drawing Area file. These files may then be used in future logic designs created with LogiCaps.

Area A rectangular portion of your design including one or more objects. An area may be moved, copied, edited, deleted. An area that has been saved into an area file may be re-entered into a drawing at the current cursor location, provided it does not cause any conflict such as symbol overlap.

Area Box Cursor The cursor—initially resembling a large flashing cross and then turning into a flashing box once you move the cursor—displayed when you enter Area Boundary mode. It allows you to define a rectangular portion of your design as an area.

Arrow Keys The keys used to move the cursor in the respective direction. The smallest increment an arrow key can move is one database unit (DBU), the largest increment is a three-digit number.

Automatic Panning LogiCaps feature that automatically moves and relocates the drawing when you move the cursor over the screen edge.

Boundary Box The dotted line surrounding a symbol or text field indicating the limits of that field. Any changes made to the symbol or text field must be made when the cursor is inside or on top of the boundary line.

<Center> The center button of a three-button mouse interface.

Character Type The size of character used for text entry. There are four available types: small caps, medium, medium reverse video, and large. The current type specified is shown in the **TXT:** field in the status line.

Command line The line at the bottom of the display screen reserved for typing in commands. Normally, the text cursor is displayed at the location where text should be entered. The right side of the command line is designated for messages output by LogiCaps.

Connection Dot A dot entered with the mouse or the keyboard command at an intersection of two signal wires. It indicates that the wires are connected.

Control Keys The <Tab>, <Home>, <+>, <*>, <PgUp>, <PgDn>, <End>, <Space>, and <Esc> keys used as shortcuts to some of the more frequently used functions.

Database Unit (DBU) The smallest unit of measure on the LogiCaps screen. One DBU corresponds to 0.1 inch in the plotted drawing.

Display Field One of the fields on the screen that shows the current status of your design entry, such as status line display, prompt line display, etc.

Full Scale Mode The display mode in which the display on the screen is shown in full scale. All text is readable and grid lines are visible.

Function Keys The keys labeled <F1> to <F10>. These keys are used as "shortcuts" for frequently used functions or long key sequences such as macros.

Graphics Cursor The cursor—resembling a flashing X— displayed on the screen when you are in drawing mode. When LogiCaps is invoked, it is positioned in the upper left corner of the display screen. It may then be moved with the mouse or with the arrow keys.

Help File A file displayable at any time giving information about a specified LogiCaps menu. Each LogiCaps menu has its own Help command and menu-specific help information.

<Home> A control key that centers the viewing window around the current graphics cursor position.

Inactive Window That portion of the split screen display into which entries may not be made. When you are in Window Split mode, one portion (indicated by a solid borderline) of the screen is active, the other (indicated by a dashed borderline) is inactive. You can edit only the active window; however, all changes made are reflected in the inactive window. You may toggle a window from inactive to active with the <Tab> key.

INIT.MAC A macro file called up every time LogiCaps is invoked. It allows you to initially set certain drawing requirements, such as display color, so that you don't need to specify these requirements each time you invoke the program.

<Left> The left button of a three-button mouse interface.

Line Draw Cursor The cursor displayed when you enter Line Draw mode. It is a large flashing cross resembling coordinate lines. A line is drawn along either the horizontal or the vertical line of the cursor.

Line Segment A straight line in the drawing that ends at a connection dot, a symbol pin stub, or a line corner.

Line Type The type of line chosen for line entry. There are five available types: solid, dotted, high-density dotted, dashed (or broken), and bus lines. The current type specified is shown in the **LINE:** field in the status line. The solid line is the default.

.MAC The filename extension assigned automatically to a macro being recorded.

Macro A sequence of keystrokes and mouse movements recorded and stored in a file with the extension **.MAC**. A macro may be recalled with one keystroke or by filename and may be executed within the currently displayed schematic.

MACRO.LIB The file containing all MacroFunctions supplied by Altera. This file is used by the ADP. It describes the logical function of MacroFunction symbols. You may not add your own MacroFunctions to this file or alter it in any other way.

MacroFunction A high-level building block used together with existing gate and flipflop primitives to provide a versatile design environment for EPLD logic development.

MacroFunction library A library file containing your custom-made MacroFunctions. It may not have the name **MACRO**. This file is used by the ADP. It describes the logical function of your MacroFunction symbols.

Maximum Reduced Scale Mode The display mode in which the display on the screen is shown reduced to the smallest size so that the entire drawing is displayed on one screen. Text fields are indicated with shaded areas, but text cannot be read. Grid lines may be shown only by grid line interconnection dots.

Message Line The field to the right of the command line. It is reserved for error messages and other types of information output by LogiCaps.

MKY Mode The cursor mode in which the cursor movements respond to the mouse movements. (Useful if you have to use LogiCaps without a mouse.) Note that in reduced drawing mode the mouse movements do not correspond to the database increments.

Pin Stub The portion of a line inside the symbol boundary of a symbol indicating the input and output pins. A line drawn in the schematic must be right up to the boundary line and to a pin stub to be recognized as a connection in the Altera Design File.

PLCAD4 A fully integrated Altera Development System that includes the LogiCaps schematic capture program, A+PLUS software, and the TTL MacroFunction library, as well as a programming card, the Master Programming Unit, LogicMap II software, and all documentation.

READ.ME File A file on the LOGICAPS distribution diskette. It contains information about changes made to the program since the release of the documentation, and should be read before LogiCaps is installed.

Reference Index The fixed numbers shown along the left and bottom sides of the screen indicating increments of database units.

<Right> The right button of a three-button mouse interface.

Rubberbanding A feature provided by LogiCaps that allows you to move symbols and areas without losing signal connectivity. The feature may be enabled or disabled.

SDFILE The default filename provided by LogiCaps and displayed in the **FILE:** field in the status line when LogiCaps is invoked.

Security bit A control bit that, when set to ON, prevents interrogation or inadvertent reprogramming of an Altera EPLD .

Split Screen The display mode that shows two portions of the screen, each containing a view of the schematic, but not necessarily the same view. One of the portions is active so that changes may be entered into the schematic. These changes are reflected in the inactive portion. The cursor is always in the same location in both portions.

Symbol Reference Number The number displayed (with the SN [Symbol Numbers] command) just inside the upper left corner of a symbol indicating the sequence in which symbols were entered. If a symbol is deleted, the number sequence will show a gap. This gap will be closed and the database will be repacked as soon as the drawing is saved or just before an Altera Design File is generated.

Temporary Line A line generated by LogiCaps during a symbol or area move and while the rubberbanding feature is enabled. This line maintains connectivity during the move even when there is a conflict with other symbols. However, once the move has been completed, any remaining temporary lines must be replaced with solid lines in order to be recognized in the Altera Design File.

Text Cursor The cursor—resembling an underscore—displayed on the command line always and in the display field when you are in Text mode. It indicates where a character will be placed when you type on the keyboard.

Text Field A defined field in the schematic that contains alphanumeric characters. A string may not be longer than 250 characters. A text field may be edited, moved, copied, and deleted, but you should not edit or delete those text fields that contain symbol reference numbers and symbol names; these are essential for generating an Altera Design File (ADF).

TITLE A schematic symbol required for all logic schematics created with LogiCaps. It is used to document a schematic.

Turbo-Bit A control bit that, when set to ON, allows you to choose the speed and power characteristics of an Altera EPLD.

Window That portion of your design that is currently displayed on the screen. You may look at any portion of your design regardless of the screen limitations by zooming in on or out of the drawing or by panning.

Zoom Levels The various levels of focus available in LogiCaps. Depending on the specified size of a drawing, you may view the schematic at full scale, maximum reduced scale, and a varying number of intermediate scales.

Index

B

C

D

E

F

G

M

N

O

P

Q

R

S

T

W

X

Z