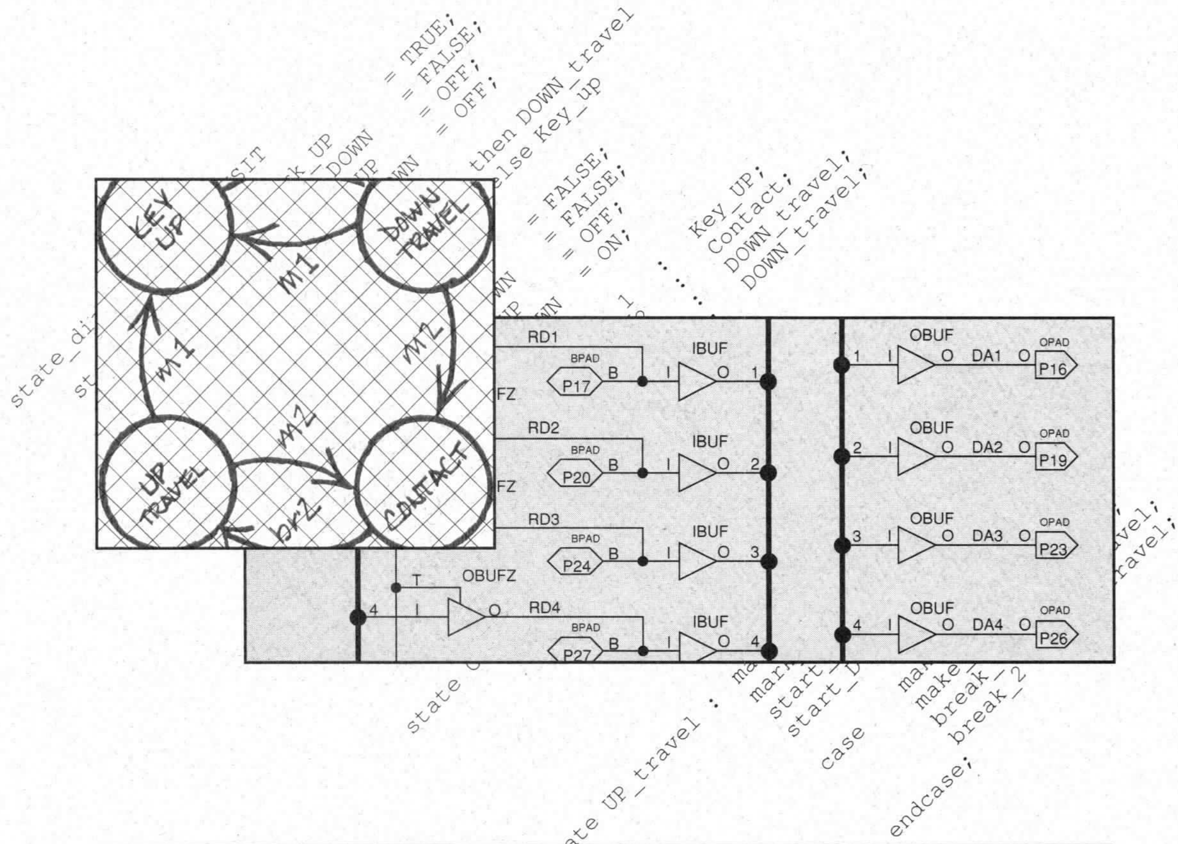
FutureNet®

SCHEMATIC DESIGNER



SCHEMATIC DESIGN TOOL

DATA I/O Corporation

981-0248

FutureNet®

Schematic Designer

User Manual

DATA I/O
Corporation

096-0013-003

August 1991

096-0013-003

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA
(206) 881-6444

Acknowledgments:

Data I/O and FutureNet are registered trademarks of Data I/O Corporation.
Sun is a registered trademark of Sun Microsystems, Inc.

Table of Contents

3. Using FutureNet

4. Menus and The Command Line

5. Mouse, Modes and Cursors

List of Figures

List of Tables

Preface

The Preface contains details about telephone support, warranty service, the Bulletin Board Service, typographic conventions and more.

Customer Support Offices

United States

For technical assistance, contact

Data I/O Customer Resource Center
Telephone: 800 247-5700
Fax: 206 882-1043

For warranty service, contact your nearest Data I/O Service Center below:

Data I/O Corporate Office
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA 98073-9746
Telephone: 206 881-6444
Fax: 206 882-1043
Telex: 152167

Data I/O California
1701 Fox Drive
San Jose, CA 95131
Telephone: 408 437-9600
Fax: 408 437-1218

Data I/O Northeastern United States
20 Cotton Road
Nashua, NH 03063
Telephone: 603 889-8511
800 858-5803 (NJ & NY only)
Fax: 603 880-0697

Canada

For technical assistance, contact:

Data I/O Customer Resource Center
Telephone: 800 247-5700
Fax: 206 882-1043

For warranty service, contact:

Data I/O Canada
6725 Airport Road, Suite 302
Mississauga, Ontario, L4V 1V2
Telephone: 416 678-0761
Fax: 416 678-7306

United Kingdom

For technical assistance or warranty service, contact:

Data I/O Limited
660 Eskdale Road
Winnersh, Wokingham
Berkshire RG11 5TS
Telephone: 0734 440011
Fax: 0734 448700

Japan

For technical assistance or warranty service, contact:

Data I/O Japan
Sumitomoseimei Higashishinbashi Bldg. 8F
2-1-7, Higashi-Shinbashi
Minato-Ku, Tokyo 105
Telephone: 03 3432-6991
Fax: 03 3432-6094 (Sales)
03 3432-6093 (Other)
Telex: 2522685 DATAIO J

Germany

For technical assistance or warranty service, contact:

Data I/O Electronic Systems Vertriebs GmbH
Lochhammer Schlag 5a
D-8032 Gräfelfing
Telephone: 089 858580
Fax: 089 8585810

Other European Countries

For technical assistance or warranty service, contact the office below and ask for the number of your local Data I/O representative:

Data I/O Europe
World Trade Center
Strawinskylaan 537
1077 XX Amsterdam, The Netherlands
Telephone: +31 (0)20 6622866
Fax: +31 (0)20 6624427

Other Countries Worldwide

For technical assistance or warranty service, contact the office below and ask for the number of your local Data I/O representative:

Data I/O Intercontinental
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA USA 98073-9746
Telephone: 206 881-6444
Fax: 206 882-1043
Telex: 4740166

Technical Assistance

Calling

To help us provide quick and accurate assistance, please be at your programmer or computer when you call, and have the following ready:

- Product version number
- Product serial number (if available)
- Detailed description of the problem you are experiencing
- Error messages (if any)
- Device manufacturer and part number (if device-related)
- Product manual

For technical assistance, contact the appropriate Customer Support office listed at the front of the Preface.

Electronic Mail

You can also reach Data I/O via electronic mail (e-mail). To help us provide quick and accurate assistance, please include the information listed above. Also, include your name, phone number, and e-mail address in your message, and send it to one of the following addresses:

`techhelp@Data-IO.COM`

or

`{apple|decwrl|rutgers|gatech|uunet}!pilchuck!techhelp`

Note: Select one of the five addresses listed above in braces. For example, you might send e-mail to the following address: `uunet!pilchuck!techhelp.`

See your system administrator if you need more information on which address to use.

Warranty Information

Data I/O Corporation warrants this product against defects in materials and workmanship at the time of delivery and thereafter for a period of ninety (90) days.

The foregoing warranty and the manufacturers' warranties, if any, are in lieu of all other warranties, expressed, implied or arising under law, including, but not limitied to, the implied warranties of merchantability and fitness for a particular purpose.

Data I/O maintains customer service offices throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. For warranty service, contact the appropriate Customer Support office listed at the front of the Preface.

Bulletin Board Service

From the Data I/O Bulletin Board System (BBS) you can obtain a wide range of information on Data I/O products, including current product descriptions, new revision information, technical support information, helpful application notes, and other miscellaneous information. In addition, the BBS has a large collection of useful DOS utilities you can download.

The BBS message facility allows you to leave messages for the BBS system operator, Customer Support personnel, or other customers. Functions are available to provide device support information or to request support for a particular device.

To learn more about the U.S. Data I/O BBS, call 1-206-882-3211. Multiple lines are available, all supporting 1200/2400/9600/19200 baud with U.S. Robotics Courier HST modems with V.42bis compatibility, set to 8 data bits, 1 stop bit and no parity. Call 1-206-861-6959 to contact the BBS connected to a U.S. Robotics Dual/HST modem supporting 1200/2400/9600/19200 baud with V.32bis/V.42bis compatibility. Online help files provide more information about the BBS and its capabilities.

For your nearest Data I/O Bulletin Board System outside the U.S., contact the appropriate Customer Support office listed at the front of the Preface.

End User Registration and Address Change

If the end user for this product or your address has changed since the Registration Card was mailed, please notify your nearest Customer Support office as listed at the front of the Preface. This ensures that you receive information about product enhancements. Be sure to include the product serial number, if available.

Typographic Conventions

Throughout this manual different typographic conventions represent different cases of input and output.

Keyboard Keys

Keyboard keys may be shown in boxes (for example, [Q]) or as bolded text.

The **Enter** key (or on some keyboards, the **Return** key) is represented by this symbol: [↵].

Key Combinations

Key combinations, such as **Control-Z**, are shown as two key boxes separated by a dash; for example, [Ctrl] – [Z].

A key combination like [Esc] [Ctrl] – [T] means to press and release [Esc], then press [Ctrl] and [T] at the same time.

Variable Input

Variable input is italicized and should be replaced with the requested information. For example, enter **copy** ***filename.hex*** means type **copy** just as you see it and replace ***filename.hex*** with the name of your file.

Optional Input

Optional items of a command are shown in brackets; for example

[*option1*] [*option2*]...[*optionn*]

Items separated by a vertical bar (for example, OR | OR | ...) are mutually exclusive; that is, only one of the options listed can be specified.

Displayed Messages

Text that appears on the screen will be displayed in a typewriter-like typeface; for example,

`You will see this text displayed on the screen.`

1 *Introduction*

The FutureNet® Schematic Designer (FutureNet) can be used as both a drawing editor and a schematic design tool. To use FutureNet as a drawing editor, you only need to know the basic drawing and editing commands. To use it as a schematic design tool, you need to know the drawing conventions that allow the FutureNet Post Processing Tools (Post) to extract circuit and connectivity information from your design.

This chapter gives a quick introduction to the basic drawing and editing commands. Chapter 1 gives the drawing conventions required to use the Post processors, and can be skipped if you use FutureNet only as a drawing editor.

Drawing Elements

FutureNet drawings consist of three elements: lines, symbols, and text. Elements are placed on a positive coordinate graph with the origin in the upper left corner. Elements and coordinates can be in three possible scales: screen pixels, display units (4 screen pixels) or drawing reference coordinates (8 display units)

About Display Units

The FutureNet graphic environment is based on display units, which are 4 screen pixels wide and 4 pixels high. The display unit is a standard of movement and measure, and forms a grid on which schematics are built. You can view this grid by entering

GRID 1,1

Symbols and lines are placed on a display unit grid, which is 20 display units to the inch in both the horizontal (or x) and vertical (or y) axes. Drawings can be sized in inches or millimeters, but these dimensions are converted to display units, and most command coordinates are specified in display units.

In the symbol editing mode, lines and graphics are placed on a screen pixel grid. Using pixels to draw allows you to create the intricate shapes necessary for graphic symbols. However, once these symbols are completed and copied to the drawing workspace, their locations and movements are subject to the display unit.

Following are some key points to remember about FutureNet and display units:

- The XY status field (in the status bar on the right of the screen) reports the graphics cursor position as *x,y* coordinates given in display units (except in symbol editing mode, where coordinates are given in pixels).
- FutureNet libraries are built on a 3-display unit convention. For instance, most ICs in the libraries have pins that are spaced 3 display units apart, based on the default text font of '2, which is 3 display units high.
- Lines may be drawn only in whole display unit increments.
- Cursors that control graphic elements do not move in display units, but the graphic elements they control are placed on the display unit grid. For example, you can place the graphics cursor point between display units, but when you enter the command to begin drawing a line, the line start point snaps to the nearest display unit (positive *x,y*—to the right and below the current cursor location), while the line drawing cursor remains between display units.

Lines

Basic information on lines is given below:

- Lines can be drawn horizontally and vertically, but not diagonally.
- Lines can be used to indicate signals and buses (electrically significant), or as graphic elements (not electrically significant).
- Lines can be solid or dashed, thin or thick. There are 10 line types.
- Lines can include special line graphics, such as arrows and dots.
- Lines cannot be routed through a symbol body.

Symbols

Basic information on symbols is given below:

- Symbols are read from a library and copied into the drawing.
- Symbols are sized in display units and are drawn in pixels, with a fixed origin in the upper left-hand corner.
- Symbols have a rectangular border used for boundary conflict checking.
- Updates to the symbol libraries do not change symbols in drawings.
- Each symbol loaded into a drawing is assigned an integer number, called a symbol reference number, that uniquely identifies that symbol for the drawing. The symbol reference number appears in the upper left-hand corner of the symbol.

- Symbols can have connection points on the symbol boundary called pins. Pins can be placed on any display unit. Normally pin spacing is at least 2 display units apart, because FutureNet text fields require at least 2 display units.

Text

Basic information on text is given below:

- Text can be placed on any unoccupied space on a drawing: inside, outside, or across a symbol boundary.
- Text can be visible or hidden (layered).
- There are seven fonts, plus overscoring and underscoring.
- Text is used to comment on the drawing, assign electrical properties to drawing elements, or provide information for post processing.

Connectivity

FutureNet has a built-in understanding of connectivity, for example, pinsnap and rubberbanding.

Pinsnap causes a signal line to snap to a pin when the line is within a certain distance of that pin, making the connection without requiring you to line up the signal and pin exactly.

Rubberbanding is a temporary display of connections that have been maintained when a symbol is moved. These temporary lines can be made permanent. Within limits, you can move a symbol and not have to redraw the connections.

These two features can be turned on or off as needed.

You can bump two symbols against one another and any pins that line up along the symbol boundaries are connected and remain connected when one of the symbols is moved away.

FutureNet recognizes two electrically significant line types: bus and signal lines. Temporary lines are also electrically significant, but you should make all temporary lines permanent before post processing.

If You Do Not Intend to Use the Post Processors

If you want to create drawings only, and won't use the Post processing tools, then the information in Chapter 2 doesn't apply to you. You can skip to Chapter 3, "Using FutureNet."

2 Understanding FutureNet

FutureNet can be used as both a drawing editor and a schematic design tool. This chapter covers the conventions you need to follow to use the Post processors. You can skip this chapter if you use FutureNet only as a drawing editor.

Designing Schematics

The FutureNet Post Processing Tools check your design, run pinlists and netlists, and translate your drawing for use with other supported tools. If the appropriate information is not available to the Post processors, they will not be able to process your design. This chapter provides conceptual information about how FutureNet and the Post processors work, and how you provide the information needed to process your design.

Post Processing Tools

The FutureNet Post Processing Tools are

Drawing Connectivity Model Generator	Establishes local and global connection data.
Design Rule Check	Checks for basic wiring errors.
Pinlist Generators	Produces two forms of pinlist.
Netlist Generators	Produces two forms of netlist.
Parts List Generator	Counts and lists the parts in the design.
EDIF 2 0 0 Netlist Generator	Produces an EDIF 2 0 0 netlist.
DPLOT	Produces plot files for a variety of printers and plotters.

The EDIF Netlist Generator and DPLOT are documented in individual manuals. The rest are covered in the *FutureNet Post User Manual.*

Many new terms are introduced in the following paragraphs. Refer to the glossary if a term is unfamiliar.

Alphanumeric Fields

An **alphanumeric field** is a displayed line of text, with an associated justification, orientation and point of effect, and either an predefined property (attribute) or a custom property assigned through layered text property sheet.

Alphanumeric fields assign properties to schematic elements. They identify and describe lines and symbols and their components and can be used to imply connections, add data required for post processors, or as comments.

The following rules apply to alphanumeric fields:

- Display text cannot overlap other schematic elements (except for symbol boundaries).
- Alphanumeric entries can be made in one of seven font sizes.
- Display text is limited to 80 characters, including spaces. Fields longer than 80 characters can exceed limitations imposed later by the host machine or Post processor target output.
- Inversion is denoted with an overbar, which is equivalent to a trailing minus. (An overbar is translated to a trailing minus (-) in the Post processor output.) Figure 2–1 shows signal SIG1 inverted.

Figure 2-1
Example of Inversion Notation

Justification and Orientation

The justification and orientation of alphanumeric fields determine how the text appears on the screen.

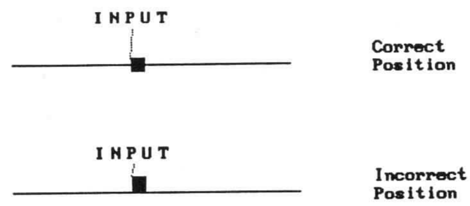
Point of Effect

The point of effect is a display coordinate used to associate an alphanumeric field with a schematic element. All alphanumeric fields have a point of effect.

The point of effect is a small, highlighted box. You place it by entering the **'P** command and moving the point of effect cursor to the drawing element you want the alphanumeric field associated with. Placing a signal name's point of effect on the signal line it references allows Post processors to associate the signal name with the signal line. Refer to the example in Figure 2–2. Points of effect are displayed with **POEDISP**.

When certain combinations of commands have been entered, the point of effect on fields with signal attributes snaps to the signal line nearest the display text. This feature lets you assign alphanumeric fields to signals without manually placing each point of effect.

Figure 2-2
Correct/Incorrect Point of Effect Placement



You can place more than one point of effect on the same location (for example, at pin references).

Attributes

All alphanumeric fields are assigned a set of properties that determines how the text is interpreted. FutureNet has several predefined property sets called attributes, or you can assign custom property sets through layered text property sheets.

Layered Text Property Sheets

Alphanumeric fields can have layered text, which is added on a property sheet dialog box or from the command line. Layered text does two things:

- Provides space for property information that does not fit in display text fields on the drawing.
- Accommodates multiple translators that have different property, or property and value requirements.

Properties

A property is a single characteristic of a schematic element. All schematic elements have at least one property, though most have more. Properties identify things about an element such as:

Characteristic	Examples
Type	tristate, output, input
Name	GND, tsv, sig01
Reference designator	u21, u0001, u15
Part name	and/2, 8080, 244

Property information must be available before the Post processing tools can interpret and process your schematic.

Property values have top-down precedence in the design hierarchy. A property specified in a higher-level drawing takes precedence over the same property specified in a lower functional block drawing. A property for a given schematic element should not be specified more than once.

Property Assignment

All properties are assigned to schematic elements through alphanumeric fields. Alphanumeric fields can assign properties to schematic elements in four ways:

- Attributes (predefined sets of properties)
- Layered text only
- A combination of layered and display text
- Display text only

Properties Derived from Attributes

Most property information needed for the Post processors is supplied by predefined attributes. Processing tools other than the FutureNet Post processors may require property information not provided by FutureNet attributes. You need to assign property information to the schematic elements in your design when the needed property information cannot be obtained from attributes. See "Property Assignment Statement" below.

An attribute is a predefined set of properties that identifies one or more of the properties characteristic of a schematic element. Attributes are assigned to a schematic element through the alphanumeric field associated with the element. Each alphanumeric field must have an assigned attribute (indicated by a number or name), which identifies the predefined property set associated with the schematic element.

Property Assignment Statement

Property assignments are normally entered in layered text using either of the two property assignment statements listed below:

[*tool*:]*property*=[*value*] (or)

[*tool*:]*property*#[*value*]

tool	(Optional) Names the Post processing tool the *property* is used with.
property	Identifies a single characteristic of the schematic element.
value	The data value for the property.

You can use the = and # characters in the *value* field of the property assignment statement, but not in the *tool* or *property* fields, since these characters are used to divide these fields.

The Post processors do not use the hyphen (-) character in tool or property names. You can use the hyphen if you are writing your own translators and want to make the tool and property names unique.

Each property assignment is entered as a single line of text using the layered text editor or layered text commands.

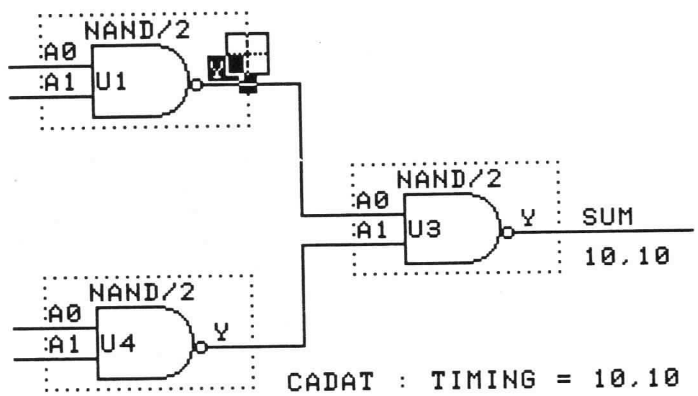
Properties in Layered Text

Figure 2–3 shows a portion of a schematic diagram that has property information contained entirely in layered text. The drawing cursor is placed on the output pin of the NAND at reference U1. Layered text for the alphanumeric field contains the string:

```
EDIF : TIMING = 10,10
```

where EDIF is the tool, TIMING is the property, and 10,10 is the value.

Figure 2-3
Cursor on a Pin with Layered Text

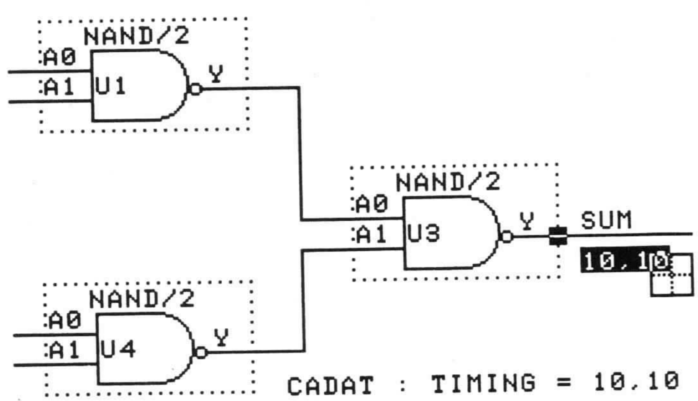


Properties in Both Layered and Display Text

Properties can be assigned in both layered and display text by entering the *value* portion of the property assignment statement in the display text portion of the alphanumeric field, and then entering the [*tool*]:*property*= or [*tool*]:*property*# portion of the layered text statement in layered text, omitting the value portion of the line.

Figure 2–4 shows the same portion of the schematic diagram as Figure 2–3 , but with the drawing cursor placed on a display text field that has its point of effect at the **Y** pin of the NAND at reference U3.

Figure 2-4
Cursor on a Pin with Displayed Property Text



In this example, there is no layered text under the pin **Y** identifier field. Property information for the pin is in both the displayed text field **10,10** and its associated layered text. The layered text entry looks like this:

```
EDIF:TIMING =
```

where **EDIF** is the *tool* and **TIMING** is the *property*. Since the *value* is not specified, then, by convention, the display text is the value.

Properties in Display Text

Properties can be entered entirely in display text using the [*tool*]:*property=value* property assignment statement and one of the layered text attributes. Enter the property assignment statement in an alphanumeric field and assign the appropriate layered text attribute to the alphanumeric field. Assigning a layered text attribute to an alphanumeric field causes FutureNet to treat that field as a layered text entry. There are five layered text attributes:

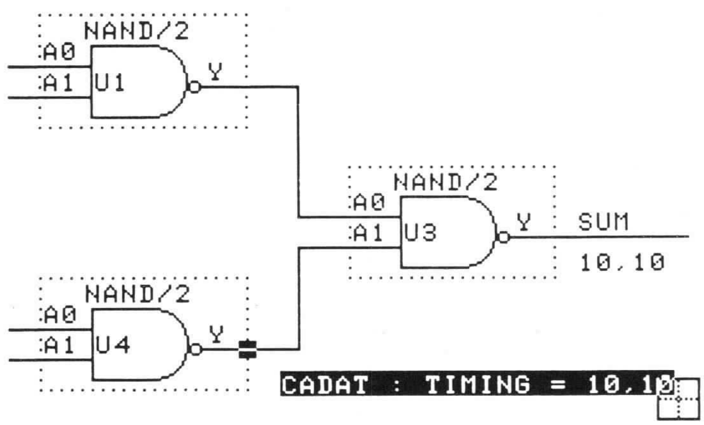
CRLT	Circuit-related
DWLT	Drawing-related
PNLT	Pin-related
SGLT	Signal-related
SYLT	Symbol-related

Figure 2–5 shows the same portion of the schematic diagram as Figures 2–3 and 2–4, but with the drawing cursor placed on a display text field that contains a complete property assignment statement. The display text field is

```
EDIF : TIMING = 10,10
```

and has its point of effect at the **Y** pin of the NAND at reference U4. There is no layered text.

Figure 2-5
Cursor on Displayed Property Assignment Statement



Specifying the Correct Attribute

All properties assigned to an alphanumeric field must be of the same class as the alphanumeric field. For example, if the alphanumeric field is assigned a pin-related attribute, then the layered text properties, or display text properties which share the point of effect of the alphanumeric field, must also be pin-related.

Attributes and Properties

An attribute is a predefined set of properties represented by a name (mnemonic, for example, LOC, SIGI, and PIN) or number.

For example, the attribute LOC identifies the reference designator for the symbol element it is associated with. The attribute SIGI identifies three different properties of the signal or bus element it is associated with: first, the drawing element is an input signal; second, its connectivity is limited to the drawing set; and third, the display text is the signal name.

You can assign any attribute to any element, but for successful post processing, only signal-related attributes should be assigned to signals, only pin-related attributes to pins, and so on.

Attributes are divided into seven classes:

- Circuit-related (that is, design-related)
- Drawing-related
- Pin-related
- Signal-related
- Symbol-related
- Connectivity-related
- Miscellaneous

The default attribute is COM 0. Until you specify a different default attribute, this is the one assigned to all new alphanumeric fields. You can specify a different default attribute for new fields, but for the current editing session only. You can change the attribute assignments for existing alphanumeric fields anytime.

The FutureNet predefined attributes and properties are listed in Tables 2–1 through 2–9.

Circuit-related

Attributes

Circuit-related attributes are generally assigned to text fields that pertain to the complete design or set of drawings. Table 2–1 lists the circuit-related attributes.

Table 2-1
Circuit-related Attributes

Number	Name	Description
148	CNUM	Circuit number. This attribute defines the text as the number of the complete design or circuit.
149	CREV	Circuit revision. This attribute defines the text as the revision of the complete design or circuit.
150	CDAT	Date of the circuit. This attribute defines the text as the date on which the design was last modified.
151	CTIT	Circuit title. This attribute defines the text as the title (name) of the complete design or circuit.
152	CREL	Circuit release. This attribute defines the text as the release number of the complete design or circuit.
153	CENG	Circuit engineer. This attribute defines the text as the name of the project engineer for the complete design or circuit.
158	CRLT	Circuit-related layered text. This attribute defines display text as circuit-related layered text.
163	CRRL	Circuit-related comment data. This attribute defined the text as comment data on the drawing. Associated layered text is related to the complete design.

Properties

The circuit properties **CNUM, CREV, CDAT, CTIT, CREL,** and **CENG** are identical to the corresponding circuit-related attributes and can be used in a property assignment statement to assign tool-specific or non-displayed circuit property information.

Drawing-related

Attributes

Drawing-related attributes are generally assigned to text fields contained in the title block of the individual drawing(s). Table 2–2 lists the drawing-related attributes.

Note: Alphanumeric fields with attribute COM are drawing-related if not placed within a symbol.

Table 2-2
Drawing-related Attributes

Number	Name	Description
50	TITL	Drawing title. This attribute defines the display text field as the title of the drawing.
51	DNUM	Drawing number. This attribute defines the display text field as the drawing number.
52	DREV	Drawing revision. This attribute defines the display text field as the revision level of the drawing.
53	DPAG	Drawing page. This attribute defines the display text field as the page number of the drawing.
54	DATE	Drawing date. This attribute defines the display text field as the date.
129	DWRL	Drawing-related text field. This attribute defines the display text as comment data on the drawing. Any associated layered text is drawing-related.
154	DENG	Drawing engineer. This attribute defines the display text as the name of the engineer responsible for the drawing.
159	DWLT	Drawing-related layered text. This attribute defines display text as drawing-related layered text.

Properties

The drawing properties **TITL, DNUM, DREV, DPAG, DATE,** and **DENG** are identical with the corresponding drawing attributes and can be used in a property assignment statement to assign tool-specific or non-displayed drawing property information.

Pin-related

Attributes

Pin-related attributes are further divided into two basic categories:

- Attributes that assign the associated display text as the pin identifier.
- Attributes that do not assign a pin identifier.

Attributes that Assign an Identifier

A pin is named by an alphanumeric field with an attribute that assigns the display text as the pin identifier. The point of effect of the field is the pin connection point. For any pin connection point, you cannot have more than one text field that assigns a pin identifier. Attributes that assign the pin identifier to a symbol pin are listed in Table 2–3.

Table 2-3
Pin-related Attributes that Assign an Identifier

Number	Name	Description
1	PIN	Symbol pin identifier. This attribute conveys no input or output information regarding the pin and is generally used in resistors, capacitors, and similar analog-type symbols.
20	PINT	Symbol output pin identifier for an output that can be switched to high-impedance or tristate.
21	PINO	Symbol output pin identifier. This attribute defines a pin that serves as a TTL output only. This type of output cannot be an open-collector type and cannot be a wired-OR type.
22	PNBT	Symbol bidirectional 3-state pin identifier. This attribute defines a pin that serves as an input or an output, and can be switched to a high–impedance state.
23	PINI	Symbol input pin identifier. This attribute defines a pin that serves as an input only.
24	CONI	Connector input pin identifier. This attribute defines a connector pin that serves as an input only.
25	CONO	Connector output pin identifier. This attribute defines a connector pin that serves as an output only.
26	CONB	Connector bidirectional pin identifier. This attribute defines a connector pin that serves as an input or an output.
27	PNOC	Symbol open collector pin identifier. This attribute defines a connector pin that can be switched to an open collector output.
28	PINB	Symbol bidirectional pin identifier. This attribute defines a symbol pin that serves as an input or an output.
29	PNBO	Symbol open collector bidirectional pin identifier. This attribute defines a pin that serves as an input or an output, and can also be switched to an open collector output.

100	GND	Symbol ground pin (signal ground) identifier. This attribute implies an implicit signal name GND is connected to this pin, and will be part of any signal net explicitly named GND.
101	+5V	Symbol power pin (+5V) identifier. This attribute implies an implicit signal name +5V is connected to this pin, and will be part of any signal net explicitly named +5V.
103	PINN	Symbol pin identifier. Same as PIN but default printability of the text field is off. See SGNU in Table 2–5.
105	+12V	Symbol power pin (+12V) identifier. Used with emitter-coupled logic. This attribute implies an implicit signal name +12V is connected to this pin, and will be part of any signal net explicitly named +12V.
106	-12V	Symbol power pin (-12V) identifier. Used with emitter-coupled logic. This attribute implies an implicit signal name -12V is connected to this pin, and will be part of any signal net explicitly named -12V.
107	VEE	Symbol power pin identifier. This attribute implies an implicit signal name VEE is connected to this pin, and will be part of any signal net explicitly named VEE.
139	SPIN	Signal pin identifier.
141	BPIN	Bus pin identifier. Used to pass a bus to within a functional block.

Attributes That Do Not Assign an Identifier

Attributes that do not identify a pin can be used to provide additional pin information. They are listed in Table 2–4.

Remember, for any point of effect, you cannot have more than one text field with an attribute from Table 2–3. You can assign additional text fields to the point of effect by using the attributes listed in Table 2–4 to add additional properties to a pin.

Table 2-4
Pin-related Attributes that Do Not Assign an Identifier

Number	Name	Description
145	PNRL	Pin-related text field that is comment data on the drawing. Associated layered text is pin-related.
162	PNLT	Display text should be treated as pin-related layered text.
144	PNUM	Display text that is the physical pin number of a symbol. (See note below.)
169	PNAM	Display text that is the logical pin name of a symbol, such as CLK, Q1, etc.

Creating Pin Identifiers

Pin identifiers can be alphabetic or numeric. Alphabetic identifiers can use any alphabetic or numeric character except the left bracket ([), the less-than symbol (<), or the comma (,). These characters are used as *delimiters,* indicating the end of a pin name. A delimiter separates elements in a line of text. Characters after a delimiter and blanks at the beginning of a field are ignored by post processors.

Properties

The reserved pin properties **PNUM** and **PNAM** are identical to the corresponding pin attributes and can be used in the property assignment statement to assign tool-specific or non-displayed property information.

Name	Description
PNAM	Alphanumeric name of a pin. It can be in the display text of an alphanumeric field with a pin attribute or in layered text. If the pin display text is alphanumeric and PNAM is not specified, the PNAM property value is the pin display text.
PNUM	Pin number. It can be derived form the display text of an alphanumeric field with a pin attribute or from layered text. If the pin display text is numeric and the PNUM property is not specified, the PNUM property value is the pin display text.
TYPE	Pin characteristics that are predefined by FutureNet and derived from the pin attribute. This property and the associated values are not user-specifiable. The following values are possible:

Attributes	Values	Definitions
PIN,PINN,SPIN	P	simple pin
PINT	T	tristate
PINI	I	input
PINO	O	output
PNBT	BT	bidirectional tristate
CONI	CI	input connector
CONO	CO	output connector
CONB	CB	bidirectional connector
PNOC	OC	open collector
PINB	B	bidirectional
PNBO	BO	bidirectional open collector
GND	GND	ground pin
+5V	+5V	+5V power pin
+12V	+12V	+12V power pin
-12V	-12V	-12V power pin
VEE	VEE	VEE power pin
BUSP	BP	bus pin

Signal-related

Attributes

Signal-related attributes are divided into two basic categories:

- Attributes that assign the associated text field as the signal identifier.
- Attributes that do not identify a signal.

A signal (or bus) line can have an alphanumeric text field with an attribute that assigns the display text as the identifier of the signal (or bus) line. The signal identifier field is used by post processors to specifically identify the signal or bus.

The only signals that *must* be named with an identifier are bused signals, signals that appear on more than one sheet of the drawing set, and all bus signals. The identifying names are required to establish connectivity of the signals.

Attributes that assign signal identifiers are listed in Table 2–5. Attributes that do not assign a signal identifier, but can be used to provide additional signal information, are listed in Table 2–6.

Buses and signals can have the same identifier assigned as many times as desired for readability and establishing connectivity.

Signals can have more than one identifier assigned. However, no more than one identifier can be assigned to a bus line.

Table 2-5
Signal-related Attributes that Assign an Identifier

Number	Name	Description
5	SIG	Drawing set signal identifier.
9	BUS	Drawing set bus signal identifier.
10	SIGI	Drawing set input signal name. This attribute defines an input that originates off the drawing, such as from a connector.
11	SIGO	Drawing set output signal identifier. This attribute defines an output that goes off the drawing, such as to a connector.
12	SIGB	Drawing set bidirectional signal identifier for a bidirectional signal name. This attribute defines a signal that goes to, or originates from, off the drawing.
13	SIGR	Drawing set registered signal identifier for a registered signal name. This attribute defines a signal that goes to, or originates from, off the drawing.
14	SIGU	Universal signal identifier. This attribute should be assigned to signal name fields that are to be universal in scope.
102	SIGN	Set signal identifier. Similar to SIG/5, except it does not print as part of the drawing unless you specify it should (used with older editors).

104	SGNU	Universal signal identifier. Similar to SIGU/14, except it does not print as part of the drawing unless you specify it should (used with older editors).
115	BUSN	Set signal bus identifier. Same as BUS/9, except it does not print as part of the drawing unless you specify it should (used with older editors).
130	SNAM	Set signal identifier. Same as SIG/5.
132	BNAM	Set bus identifier. Same as BUS/9.
134	USN	Universal signal identifier. This attribute should be assigned to signal name fields that are universal in scope.
147	SIGL	Local signal identifier. Used to identify a signal as existing only within the drawing sheet. In a post processor listing, a text field with this attribute is appended with ****ref*, where *ref* is the unique drawing occurrence number.
156	PWRS	Universal power signal identifier.
157	GNDS	Universal ground signal identifier.
164	SIGP	Packaged signal identifier. Names a signal that uses a bus for connection purposes within a drawing sheet, but is not identified with the bus.
166	SGLP	Local packaged signal name. Names a signal that uses a bus for connection purposes within a drawing sheet, but is not identified with the bus. In a post processor listing, a text field with this attribute is appended with ****ref*, where *ref* is the unique drawing occurrence number.
167	PRLS	Local power signal identifier. In a post processor listing, a text field with this attribute is appended with ****ref*, where *ref* is the unique drawing occurrence number.
168	GNLS	Local ground signal identifier. In a post processor listing, a text field with this attribute is appended with ****ref*, where *ref* is the unique drawing occurrence number.

Table 2-6
Signal-related Attributes that Do Not Assign an Identifier

Number	Name	Description
136	SGRL	Signal-related text field that is treated as comment data on the drawing. Associated layered text is signal-related.
160	SGLT	Display text field that is to be treated as signal-related layered text.
165	LOGC	Logical page connection. (Reserved for future products.)

Properties

Property	Description
NAME	The NAME property specifies the signal NAME. If a NAME property is not assigned, a value for the NAME property is derived. Name values can come from display text and signal attributes; or from FutureNet through implicit connection of pins with attributes +5V,+12V,-12V,VEE, or GND; or from FutureNet-assigned names for unnamed signals. This property and the associated values are not user-specifiable.

Note: The property NAME cannot be used in property assignment statements to alias signals. The NAME value is used in post processor output only, not in determining connectivity.

TYPE

The TYPE property specifies signal characteristics. Values for the TYPE property are predefined by FutureNet and derived from the signal attribute. This property and the associated values are not user-specifiable. The following values are possible.

Attributes	Values	Definitions
SIG,SIGR,SIGU SIGN,SGNU,SNAM, SIGL,SIGP,SGLP,BUS, BUSN,BNUM,USN	**S**	signal or bus
SIGI	**I**	input
SIGO	**O**	output
SIGB	**B**	bidirectional
PWRS,PRLS	**PWR**	power signal
GNDS,GDLS	**GND**	ground signal

SCOPE

The SCOPE property specifies the range of connectivity of a signal. Values for the SCOPE property are predefined by FutureNet and derived from the attribute. This property and its associated values are not user-specifiable.

The following values are possible:

Attributes	Values	Definitions
SIGU,SGNU,PWRS, GNDS	U	universal
SIG,SIGR,BUS,BUSN, SIGN,BNAM,SIGP, SIGI,SIGO,SIGB	S	drawing set
SIGL,SGLP,PRLS, GDLS	L	drawing

Symbol-related

Attributes

Symbol-related attributes are assigned to text fields that pertain to symbols. Table 2–7 lists the allowable symbol-related attributes.

Table 2-7
Symbol-related Attributes

Number	Name	Description
2	LOC	Circuit reference designator, (for example, U11, C81, R22).
3	PART	Part number (for example, 74LS04) or part name (for example, Hex Inverter).
4	VAL	Component value (for example, 75 ohms, 0.002 microfarads).
6	TOL	Component tolerance (for example, 5%).
7	STR	Component stress (for example, 60 Vac, 0.25 watts).
138	SYRL	Symbol-related text field that is treated as comment data on the drawing. Associated layered text is symbol-related.
146	LOCL	Local reference designator. This attribute causes the reference name to be local to the drawing file. In a post processor listing, a text field with this attribute is appended with ****ref*, where *ref* is the unique drawing occurrence number.
155	GATE	Multi-gate or multi-symboled part (for example, hex inverter and quad two-input AND). This attribute is assigned to the gate identifier of a multi-gate or multi-symboled part.
161	SYLT	Assigned to display text that is to be treated as symbol-related layered text.

Properties

The reserved symbol properties **LOC, PART, VAL, TOL, STR,** and **GATE** are identical to the corresponding symbol attributes and can be used in the property assignment statement to assign tool-specific or non-displayed symbol property information.

Connectivity

Attributes

Table 2–8 lists the special connectivity attributes. They are used to show connectivity within structured designs.

Table 2-8
Connectivity-related Attributes

Number	Name	Description
8	FILE	Filename pointer. This attribute is assigned to the displayed name(s) assigned to a functional block in a hierarchical design to reference a lower-level drawing file. The name is a printable text field. If layered text is associated with the field, the display text is treated as a comment. The layered text contains a list of the drawing files that represent the functional block.
114	FILN	Filename pointer. Same as FILE/8, except that it will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).

Properties

The reserved property **FILE** can be used in the property assignment statement to override the **FILE** attribute.

Name	Description
FILE	This property is a filename pointer. This property is associated with an alphanumeric field in a functional block in a hierarchical design structure. The value of the property references a drawing file within the drawing design. The display text in the alphanumeric field is treated as comment data. There should be one property assignment for each drawing to be referenced. In the property assignment statement, *tool* is optional. If one is specified, it must be DCM.

Miscellaneous

Attributes

Table 2–9 lists several attributes that do not fall into the classes described in Tables 2–1 through 2–8.

Table 2-9
Miscellaneous Attributes

Number	Name	Description
0	COM	Indicates a comment field. The text field is passed to the Post processors. If used in a symbol, the comment is considered symbol-related; otherwise, the comment is considered drawing-related.
128	NULL	Indicates a comment field. The text field is not passed to Post processors.

Properties

The reserved properties COM and NULL match the corresponding attributes and can be used in the property assignment statements.

Signals

Signals are drawn with the single thickness (/1) line only. A line is considered to be a signal when it is connected to a signal pin, or has associated display text with one of the attributes listed in Tables 2–5 and 2–6. You can use all of the listed attributes except BUS and BUSN for signal names.

Signal Connections and Crossovers

Signal lines can cross over each other without any implied connection as long as no connection dot is placed at the intersection. See the examples in Figure 2–6.

Note: Signal connections can also be made by entering duplicate signal names and the appropriate attribute (one that assigns an identifier and has the correct scope). Refer to Signal Connectivity Scopes.

Using the Junction Symbol

The junction symbol (diagonal or curved) must be used in accordance with two basic rules. First, the junction symbol is used ***only*** to connect a signal to a bus. Second, two junction symbols cannot share a common border. Other uses are not supported by the post processors.

Using the Power and Ground Symbol

A power/ground symbol is a symbol with no pins. Instead the symbol contains a signal with the power/ground signal name with a universal attribute and a point of effect at the connection.

Using the Connector Symbol

Typically, post processors expect output pins of symbols and functional blocks to be connected to input pins of the next symbol or functional block. For example, in Figure 2–7 the inputs have attribute 23 (PINI), which is an input attribute, and the outputs have attribute 20 (PINT), which is an output attribute.

Figure 2-6
Signal Line Connections

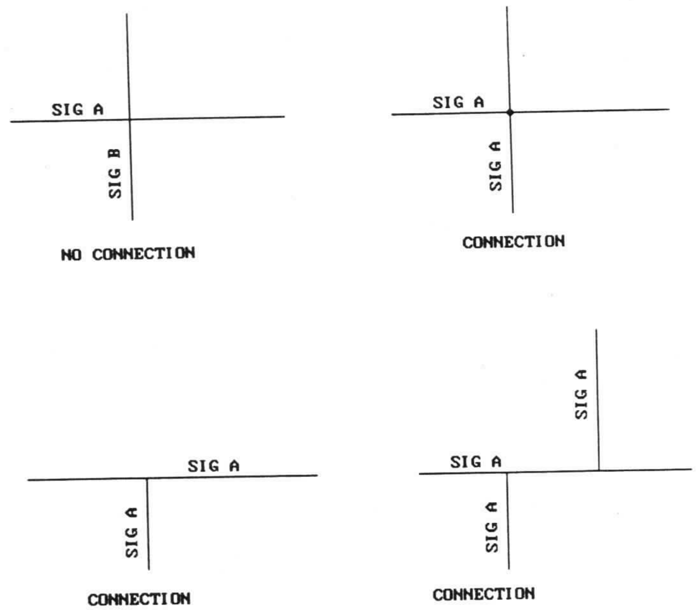
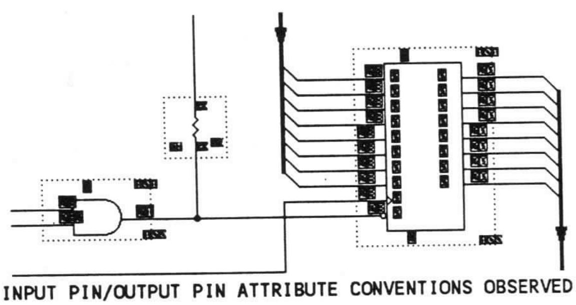
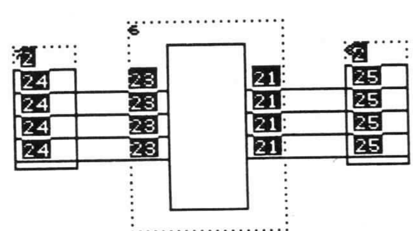


Figure 2-7
Using Input Pin and Output Pin Attributes



An exception to the input-pin-to-output-pin convention occurs when a connector symbol is used. In this case, connector input pins connect to symbol input pins, and connector output pins connect to symbol output pins. This convention is illustrated in Figure 2–8, where the pins of the input connector (attribute 24 (CONI)) connect to the input pins (attribute 23 (PINI)) of the symbol. The output pins of the symbol connect the output pins of the other connector.

Figure 2-8
Using Input Pin and Output Pin Attributes: Exceptions



Point of Effect

Point of effect rules apply to how alphanumeric fields are associated with specific signal lines. The required point of effect location for a signal name or a signal-related attribute is at any point(s) on the signal line. Multiple signal names can have the same point of effect.

Creating Signal Names

You can use any alphanumeric character in the signal name except for the left bracket ([ ) the less-than symbol (<), and the comma (,). These characters are used as *delimiters,* indicating the end of a signal name, or separating elements in a line of text. Characters after a delimiter and blanks at the beginning of a field are ignored by post processors.

Do not use names beginning with three asterisks (***), since names of this form are generated for unnamed signals. Also, the signal name N/C with attribute SIGU or SGNU is reserved to indicate a pin that is intentionally left unconnected, by connecting a signal named N/C to that pin.

Identical Signal Names

Identical signal names with the same attribute denote connectivity within a drawing file, and also across multiple drawing files, according to their attribute connectivity scope. Refer to the section "Signal Connectivity Scopes" later in this chapter.

A signal line can be assigned multiple signal names by placing the points of effect of the name fields on the signal line. The multiple signal names must have the same attribute (except SGLT, SGRL, and LOGC) and become equated by the Post processors. The SGLT, SGRL, and LOGC attributes do not assign their associated text field as a signal name, but as text fields attached to the signal line.

Signal Connectivity Scopes

There are three scopes or levels of signals with regard to intrasheet and intersheet connections. The signal scope is established by the attributes assigned to the displayed signal name text. The three scopes for signals are

- Local
- Set
- Universal

Set signal names can be further categorized as input, output, and bidirectional without any effect on the connection rules. Refer to Tables 2-5 and 2-6.

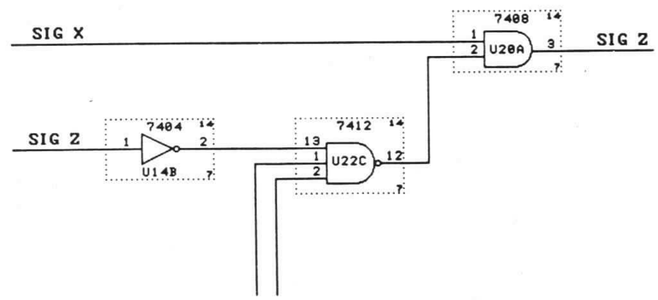
The connection rules for the three scopes of signals are outlined in Table 2–10.

Table 2-10
Signal Scope Connection Rules

Sheet Connectivity	Local	Set	Universal
Across all drawings	No	No	Yes
Across a drawing set	No	Yes	Yes
Within a drawing	Yes	Yes	Yes

In Figure 2–9, the signal named SIG Z at the right side of the drawing has an implied connection to the signal named SIG Z at the left side of the drawing. In this example, SIG Z could be a local, a set, or a universal signal name.

Figure 2-9
Signals Named SIG Z *Are Connected*



Local Signal Names

Local signal names should be assigned to signals that are to be connected together only within a particular drawing file, root drawing or functional block drawing. Identical signal names appearing on any *other* sheet in the drawing set, regardless of the assigned attribute, will not be connected to the local signal name.

Local signal names have the attribute SIGL (147) or SIGP (164).

Set Signal Names

Set signal identifiers (such as SIGO, SIGB, and SIGN) indicate connectivity when they are duplicated within a drawing or within a drawing set; that is, set signals of the same name appearing in other drawings of the set are connected. Set signal names that are duplicated outside the drawing set are not connected.

Universal Signal Names

Universal signal names indicate connectivity when used anywhere in the drawing set. Universal signal names have the attribute SIG (5), SIGU (14) or SGNU (104).

Signal Name/Scope Conflicts

Within a Drawing

If a drawing contains signals of the same name, they must have the same attribute, and, therefore, the same scope to be connected.

Across Drawings

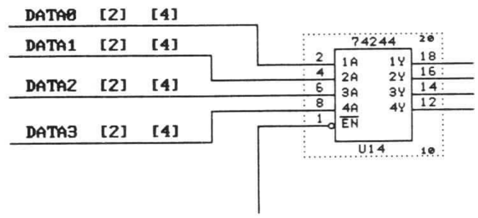
If signals of the same name appear across drawings, but are of a different scope (as determined by the attribute), the signals in conflict are connected, or not connected, as follows:

- A local/set signal conflict results in no connection.
- A local/universal conflict results in no connection.
- A set/universal conflict results in a connection.

Notation for Intersheet Connections

Intersheet connections are noted on a drawing by numbers in brackets ([ ]). The example in Figure 2–10 shows how numbers in brackets following a signal name are used to refer to other pages where the signal is used. Page references enclosed in brackets are commentary only and not used by any post processors.

Figure 2-10
Intersheet Connection Notation



Signal to Symbol Pin Connections

Signals can connect to symbol pins in any of four ways:

- Direct connection
- Point of effect connection
- Implicit connection
- Adjacent symbols

Direct Connection

A direct connection of a signal to a symbol pin occurs when the signal line is drawn to where the symbol pin's point of effect touches the symbol cell.

Point of Effect Connection

A connection by point of effect is achieved when the pin's point of effect is located within the symbol, and the signal name field's point of effect is at the same location.

Implicit Connection

This type of connection results from assigning certain pin attributes to the pin identifier (alphanumeric field). The attributes are +5V, +12V, -12V, VEE, and GND. When these pin attributes are used, no pin stub, signal line, or signal name is required. The only requirement is that the displayed alphanumeric field that identifies the pin is located within the symbol boundary. The identifier (name) of the signal that makes the implicit connection is the name of the pin attribute. For example, if pin-related attribute VEE (107) is assigned to several pins, the implied signal line that connects the pins will be identified as VEE. Implicit connections are universal in scope, and are connected to any explicitly named universal signal with the same identifier.

Adjacent Symbol Connections

When two symbols share a common boundary (that is, they are placed adjacent to each other), any pins whose points of effect overlap will have to be considered connected.

Signal Lines Through Symbols

FutureNet allows you to make pin-to-pin connections of a symbol by drawing the interconnecting line directly through the symbol instead of routing the line around the outside. Refer to Figure 2–11 and the /C command described in the *Command Reference*.

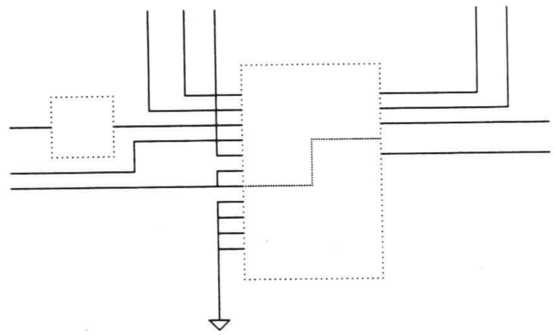


Figure 2-11
Through-symbol Connections

Buses

A bus is a triple thickness (/2) line that carries one or more signals. Such a line is considered to be a bus when it is connected to a signal line, connected to a bus pin, or has associated display text with one of the attributes listed in Table 2–11.

The attributes of display text associated with a bus are signal class attributes. Of these, BUS and BUSN are used specifically for buses. Refer to Table 2–5 for a more detailed explanation of the listed attributes.

Table 2-11
Allowable Bus Name Attributes

Name	Number	Bus Scope
SIGU	14	Universal
SGNU	104	Universal
SIGL	147	Local
BUS	9	Set
BUSN	115	Set
SIG	5	Set
SIGN	102	Set
SIGR	16	Set
SIGR	17	Set
SIGR	18	Set
SIGR	19	Set
SIGI	10	Set
SIGO	11	Set
SIGB	12	Set

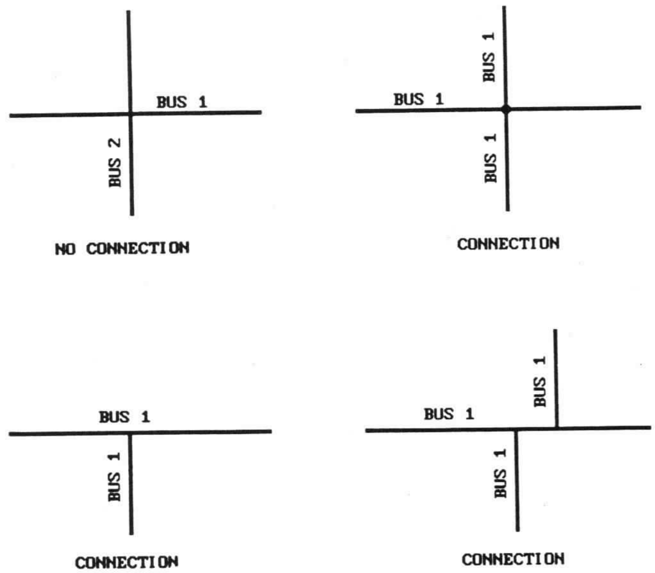
For information on bus scopes, refer to the section "Bus Connectivity Scopes" later in this chapter.

Bus Connections and Crossovers

Bus lines can cross over each other without any implied connection as long as no connection dot is placed at the intersection. See the examples in Figure 2–13.

Note: *Connections can also be made by entering identical signal names and the appropriate attribute. Refer to the section "Bus Connectivity Scopes" later in this chapter.*

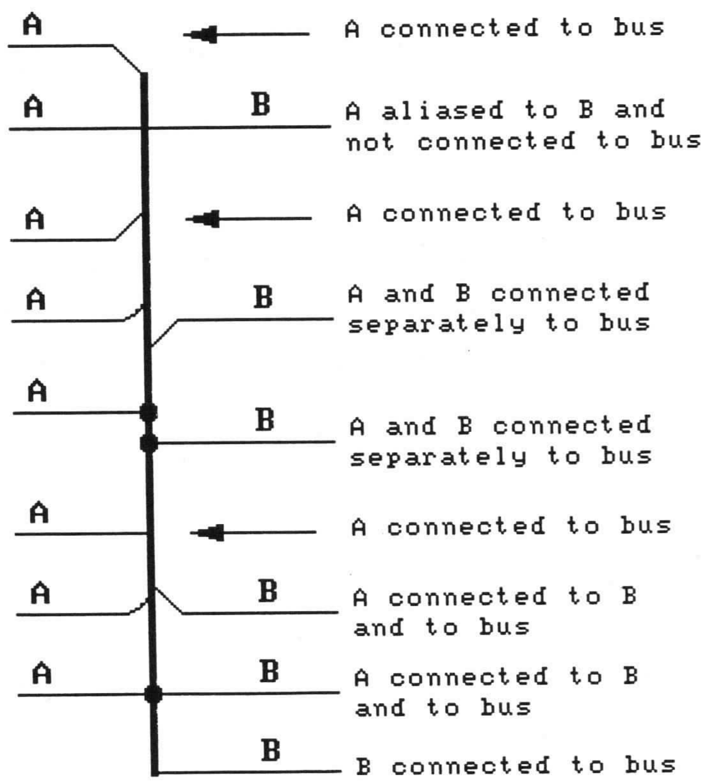
Figure 2-13
Bus Line Connections



Bus/Signal Line Connections

Signal lines can be broken out from a bus by means of the diagonal or curved line symbol (/J command), a T connection, or a connection dot. Refer to Figure 2–14 for examples of bus/signal line connections, and also instances where buses and signal lines touch but do not connect. Buses can cross signal lines and other bus lines without any implied connection.

Figure 2-14
Bus/Signal Connections



A smaller bus, that is, one that contains a subset of signals, cannot be broken out from a larger bus. For example, if you have an eight-line bus containing signals D0 through D7, you cannot directly form a bus containing signals D0 through D3.

Signals that are identified with the bus must have the same scope as the bus to which they connect. If there is a mismatch, the bus scope is used for interdrawing connections. For example, if a set signal named CLOCK is connected to a universal bus named BUSI, all signal lines with the name CLOCK on bus BUSI in the circuit will be connected.

Signals that are identified with a bus must be named; i.e, they must have an alphanumeric field that names the signal. (Attributes SGRL, SGLT, and LOGC cannot be used to assign a signal name.)

A signal cannot be connected to a bus and also to ground.

Bus Names

The description presented under Signal Names applies to buses except that a bus line can have only *one* unique bus (signal) name assigned to it. For example, if a bus is named BUS1 at any point on the drawing, it can only be named BUS1 at any other point. You can add information within square brackets [ ] or angle brackets < > such as in BUS1<0:3> since post processors recognize the bus name only to the left of a bracket character ([or <). Also, non-naming attributes (that is, SGRL, SGLT and LOGC) can be used to associate other alphanumeric fields with a bus without violating the single bus name convention.

Identical Signal Names

Identical bus names with the same attribute denote connectivity within a drawing file, and also across multiple drawing files, if the connectivity scopes allow. Refer to the section "Bus Connectivity Scopes" later in this chapter.

Bus Connectivity Scopes

There are three scopes or levels of buses with regard to intrasheet and intersheet connections. The bus scope is established by the attributes assigned to the displayed bus name text. The three scopes of buses are:

- Local
- Set
- Universal

The connection rules for the three scopes of bus names are outlined in Table 2–12 and described in the following paragraphs. The allowable bus name attributes are listed in Table 2–11.

Table 2-12
Sheet Connectivity of Signals

Sheet Connectivity	Local	Set	Universal
Across all drawings	No	No	Yes
Across a drawing set	No	Yes	Yes
Within a drawing	Yes	Yes	Yes

Local Bus Names

A local bus name attribute should be assigned to buses that are to be connected together only within a particular drawing file, root drawing or functional block drawing. Buses that are assigned local bus attributes in drawing files connect only within the drawing and do not exist outside the drawing. Identical bus names (of any attribute) appearing on any other sheet in the drawing set will not be connected. Local bus names have the attribute SIGL (147). Uunnamed buses are local buses.

Set Bus Names

Set bus name attributes indicate connectivity when they are duplicated within a drawing or within a drawing set. That is, a bus of the same name and appearing in other drawing(s) of a drawing set are connected. Set bus names that are duplicated in different drawing sets are not connected.

Universal Bus Names

Universal bus names indicate connectivity when used anywhere in the drawing set. Universal bus names have the attribute SIGU (14) or SGNU (104).

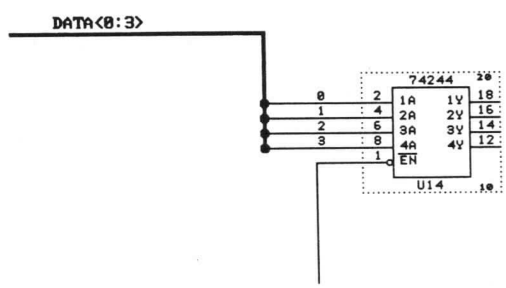
Creating Bus Names

You can use any keyboard character in the bus name except for the left bracket ([ ), the less-than symbol (<), and the comma (,). These characters are used as *delimiters*, indicating the end of a name and separating elements in a line of text. Characters after a delimiter and blanks at the beginning of a field are ignored by Post processors.

Do not use names beginning with three asterisks (***). This form is used for unnamed signals and buses.

To indicate a grouping of lines with the same basic signal name, you could use two digits separated by a colon and enclosed in < > symbols. This notation should immediately follow the basic signal name and precede any page references. For example, DATA<0:3> in Figure 2–15 indicates that lines 0 through 3 are named DATA. Note that information within the < > symbols is commentary only and most post processors ignore it.

Figure 2-15
A Bus Containing Lines with the Signal Name



Only one name can be assigned to a bus. One of the two fields can use the notation for multiple signal names as described in the previous paragraph, and still be considered identical to the other signal name. That is, as long as the bus names are identical up to the < delimiter, they are the same: the bus name DATA is the same as DATA<0:3>.

Bus Name/Scope Conflicts

Within Drawings

If a drawing contains buses of the same name, they must be of the same scope to be connected.

Across Drawings

If buses of the same name appear in multiple drawings, but are of a different scope (as determined by the attribute), the buses in conflict are connected, or not connected, as follows:

- A local/set bus conflict results in no connection.
- A local/universal bus conflict results in no connection.
- A set/universal bus conflict results in a connection.

Allowable Signals On a Bus

Any of the signals listed in Table 2–13 can be connected to a bus. The Bused/Packaged column of the table indicates which signal types are identified with the bus when connected (bused), and which are not (packaged). Refer to Bused Signals and Packaged Signals for more information on bus-to-signal identification.

Table 2-13
Allowable Signal on a Bus

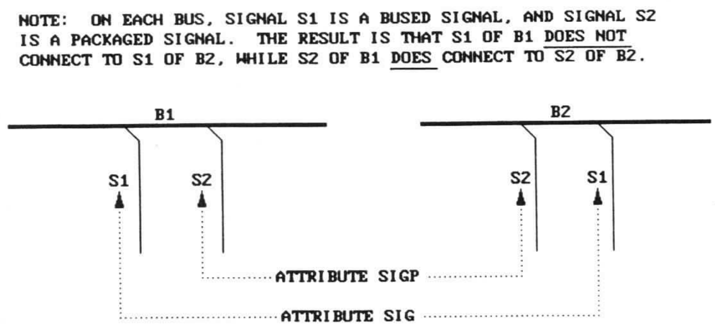
Name	Number	Bused/Packaged
SIG	5	Bused
SIGU	14	Packaged
SGNU	104	Packaged
SIGL	147	Bused
SIGN	102	Bused
SIGI	0	Bused
SIGO	11	Bused
SIGB	12	Bused
SIGR	13	Bused
SNAM	130	Bused
PWRS	156	Packaged
GNDS	157	Packaged
SIGP	164	Packaged
SGLP	166	Packaged
PRLS	167	Bused
GNLS	168	Bused

Bused Signals (Identified with the Bus)

A bused signal is a signal that connects to a bus and is named with one of the attributes listed in Table 2–13 as a bused attribute (for example, SIGB, SIGO, SIGI). Bused signals include the name of the bus as part of their identification by post processors. Therefore, to establish connectivity between multiple appearances of bused signals, the name of the signal name, and the name of the bus to which it connects, must be the same wherever the signal appears.

For example, in Figure 2–16, signal S1 on bus B1 and signal S1 on B2 are bused signals. (The attributes for bused signals are listed in Table 2–13.) Each signal is identified with its respective bus and, therefore, the two signals are not connected, although the displayed signal names are the same and appear on the same drawing sheet.

Figure 2-16
Bused and Packaged Signals Connected to a Bus



Packaged Signals (Not Identified with the Bus)

Packaged signals (attribute SIGP) are not identified with a bus, although they can use a bus line for the purpose of line routing.

Packaged signals can be connected to a bus in order to avoid multiple line routing. For example, in Figure 2–17 the routing of signal S1 can be simplified by making S1 a packaged signal and using the routing of bus BUS as shown in Figure 2–18.

Figure 2-17
Packaged Signal Example: S1 not packaged

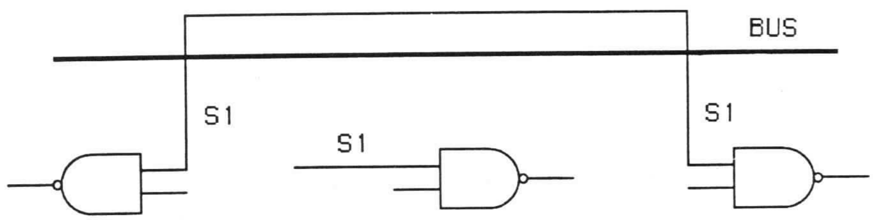
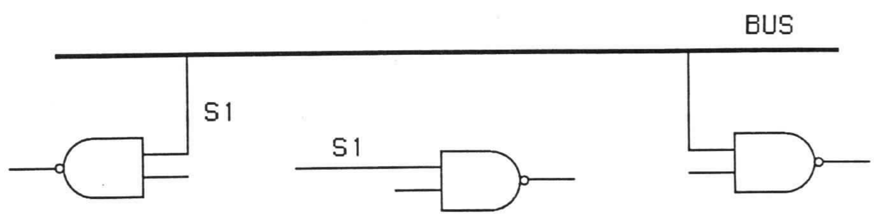


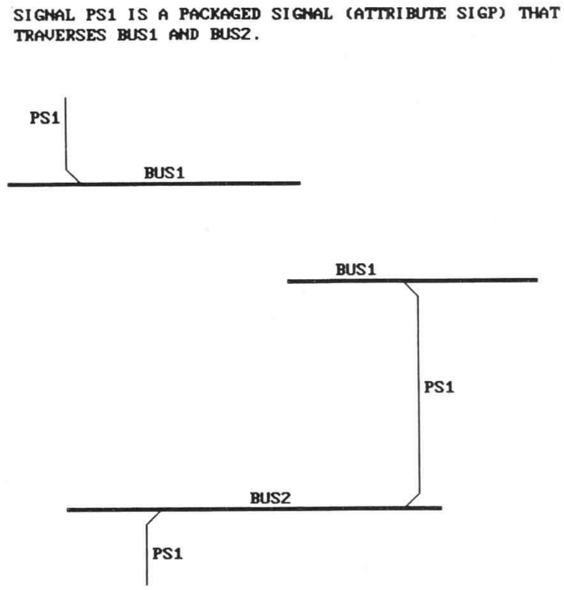
Figure 2-18
Packaged Signal Example: S1 packaged



In the example shown in Figure 2–16, signal S2 on bus B1 and signal S2 on bus B2 are packaged signals. Both signals have the attribute SIGP and are, therefore, *not identified* with the bus to which they are connected. However, since they have a common name, they are connected to each other.

Figure 2–19 shows an example of how a packaged signal can traverse from bus to bus. Signal PS1 appearing in the upper left of the example exists on both appearances of bus BUS1 and then connects over to bus BUS2. (A bused signal could not do this since it is identified with the bus it connects to and cannot be identified with more than one bus.)

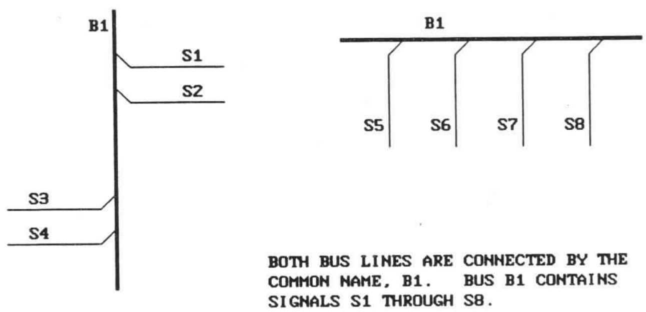
Figure 2-19
Packaged Signals Traverse Buses Without Bus ID



Buses Connected Together by a Common Name

Two buses can be connected by means of a common bus name, as long as they conform to the previously-described connectivity rules. When this type of connection is made, the resulting bus contains a union of all signals from both buses (now considered a single bus). For example, in Figure 2–20, bus B1 contains signals S1 through S8.

Figure 2-20
Buses with Common Name Contain All Signals



Buses Connected Together by Signal Lines

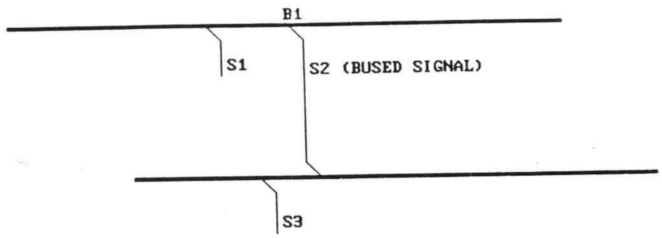
If a bused signal line connects two bus lines, the two bus lines become one bus. All signals connected through the two bus lines are considered to be on the bus.

For example, in the upper portion of Figure 2–21, the bused signal S2 connects two portions of the same bus, B1. Bus B1 contains all signals connected to both bus lines (S1, S2, and S3).

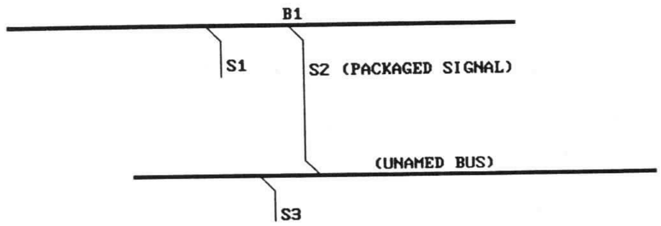
If a packaged signal line connects two bus lines together, the two bus lines remain as two separate buses.

In the lower portion of Figure 2–21, there are two buses (B1 and an unnamed bus), each of which connects to the packaged signal, S2. Bus B1 contains signal S1, while the unnamed bus contains signal S3.

Figure 2-21
Bus Signals Join Buses



WITH S2 AS A BUSED SIGNAL, THERE IS ONLY ONE BUS (B1).
B1 CONTAINS S1, S2 AND S3.



WITH S2 AS A PACKAGED SIGNAL, THERE ARE TWO BUSES (B1 AND AN UNAMED BUS). B1 CONTAINS SIGNAL S1. THE UNAMED BUS CONTAINS SIGNAL S3.

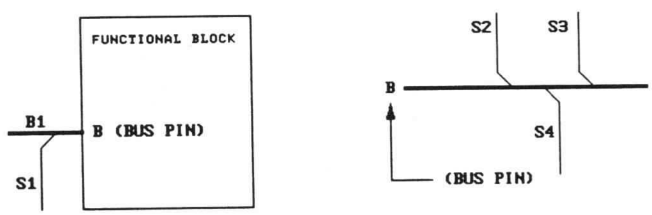
Connection to Symbol Bus Pins

When a bus is connected to a symbol bus pin (BPIN) on a functional block drawing, the bus contains a union of all bused signals currently connected to the bus, and all bused signals introduced to the bus by the functional block. For example, B1 in Figure 2–22 contains signals S1, S2, S3, and S4. Packaged signals are not associated with the bus and do not connect to the functional block.

Figure 2-22
Buses with Common Name Contain All Signals

THIS IS A PORTION OF A DRAWING THAT SHOWS A BUS CONNECTED TO A BUS PIN OF A FUNCTIONAL BLOCK.

THE DRAWING BELOW SHOWS THE BUS (NOW CALLED B) WITHIN THE FUNCTIONAL BLOCK DRAWING.



BUS B1 CONTAINS SIGNALS S1, S2, S3, AND S4.

Areas

An area is a user-definable, rectangular portion of a drawing. There is no limit to the size an area can be. Areas are probably most useful as a cut-and-paste tool, letting you restructure a drawing by moving, copying or erasing a defined portion of the drawing. Commonly used portions of a drawing can be saved as an area and then loaded into other drawings; however, if you use a particular circuit section often, save it as a drawing and reference it through functional blocks.

FutureNet automatically assigns an **.ara** file extension to areas when you save them.

Areas can be:

- Saved
- Erased
- Copied
- Moved
- Rotated
- Reflected

Symbols

Symbols can contain electronic components and other graphic items that are used in a drawing. You can create them yourself or use symbols from the symbol libraries that come with FutureNet.

Parts of a Symbol

Symbols are made up of all or some of the following parts:

Symbol Cell Boundary	The dotted line that defines the symbol cell used for boundary conflict checking. Everything inside the symbol boundary is considered to be part of the symbol. Block symbols can be created so that the symbol block overlays the symbol boundary. Symbol boundaries do not print. The size of the symbol cell is displayed in the CELL status field.
Symbol Block	The main portion of a block symbol. It is the block and its contents that you see when you print the drawing. Not all symbols have a block.
Pin Stub	A place on the symbol cell boundary where signals connect.
Reference Number	The number assigned to a symbol as it is loaded into a drawing.

Pin Alphanumeric Field — Alphanumeric field with pin attributes, assigned pin identifiers, implicit power and ground pin connections and other pin-related properties.

Symbol Alphanumeric Field — Alphanumeric field with symbol attributes and assigned symbol properties (such as reference designator and part number).

FutureNet Symbol Library Format

The symbols supplied in FutureNet libraries generally follow the format guidelines listed below and illustrated in Figure 2–23. Use these guidelines when creating symbols in the FutureNet symbol editor.

- All relevant data is located within the symbol cell boundary.
- Input pins are on the left of the symbol.
- Output pins and bidirectional pins are on the right of the symbol.
- The part number is at the top of the symbol.
- The reference designator ("U" number) is at the bottom of the symbol.
- The power supply pin number is in the field in the upper right corner of the symbol in '1-sized characters. Usually, the attribute +5V (101) is given to this field.
- Standard ground is in the lower right corner of the symbol in '1-sized characters and has the attribute GND.
- Non-standard and TTL power and ground pins are placed in the lower left corner of the symbol.

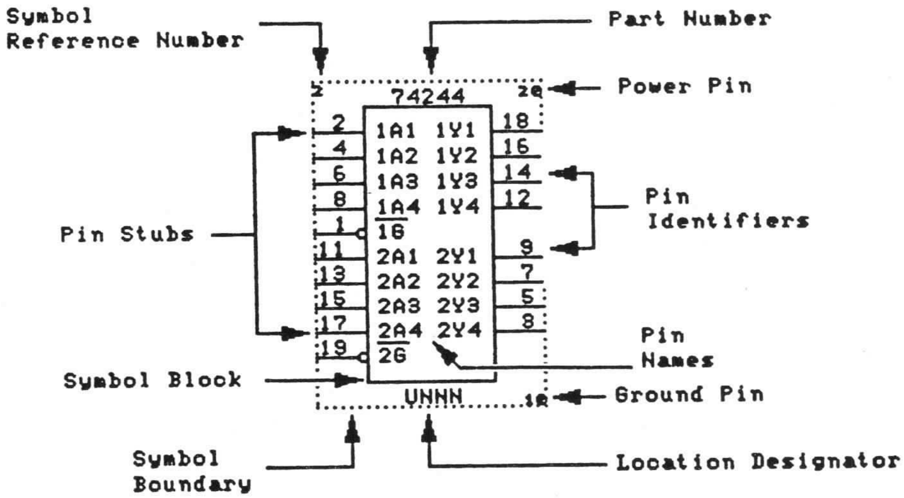
Alphanumeric Fields, Points of Effect, and Symbols

Alphanumeric fields that have a point of effect within a symbol cell contain information about that symbol. It is recommended that the alphanumeric field also be located within the symbol. Alphanumeric fields can overlap a symbol boundary if the FutureNet overlap mode is enabled.

If an alphanumeric field is located outside of a symbol, but its point of effect is inside the symbol, the following are possible:

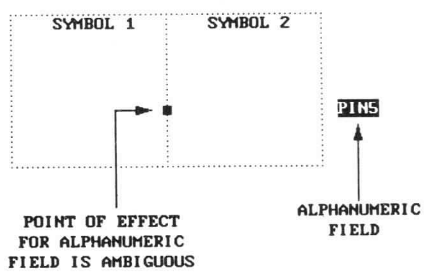
- If OVERLAP is ON, then the alphanumeric field maintains its relative position to the symbol. Boundary checks are done on the external alphanumeric field.
- If OVERLAP is OFF, but the appropriate MOVEPOEE/MOVEPOEG command is ON, then the point of effect remains with the symbol, but the text remains fixed. The point of effect snaps back to the alphanumeric field if the symbol is moved more than 127 display units from the alphanumeric field.
- If neither OVERLAP nor MOVEPOEE/MOVEPOEG is ON, then the point of effect snaps back to the alphanumeric field.

Figure 2-23
Symbol Format Guidelines



Alphanumeric fields located outside or on symbol boundaries, but having a point of effect on two or more common symbol boundaries, are ambiguous. Refer to Figure 2–24 for an example of an ambiguous point of effect for a displayed alphanumeric field.

Figure 2-24
Ambiguous Point of Effect



An alphanumeric field located within a symbol boundary with a point of effect on a common boundary is associated with the symbol containing the field.

For information on how overlapping text is treated in ambiguous cases, see OVERLAP in the *FutureNet Command Reference.*

Defining Symbol Pins

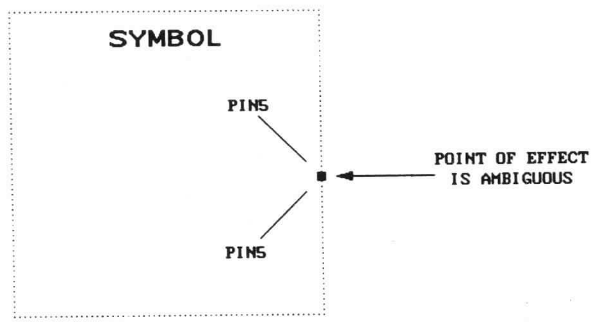
Symbol pins are alphanumeric fields with attributes that are listed in Tables 2–3 and 2–4. A pin attribute from Table 2–3 must be used to assign an alphanumeric identifier to the pin (referred to as the pin identifier). Additional pin-related information can be assigned to the pin by using attributes listed in Table 2–4 and sharing the point of effect of the established pin identifier. For example, an input pin with the PINI attribute can have one or more alphanumeric fields with PNUM, PNRL, or PNLT attributes assigned to the same point of effect.

Alphanumeric fields with PNUM, PNRL, PNAM, or PNLT pin-related attributes *must* share a common point of effect with a pin identifier.

You can assign only one pin identifier to a pin. That is, there can only be one alphanumeric field with an attribute taken from Table 2–3 for any given pin, even if both fields have the same text. For example, the two PIN5 fields in Figure 2–25 cannot have their points of effect at the same location on the drawing since they both have pin name attributes.

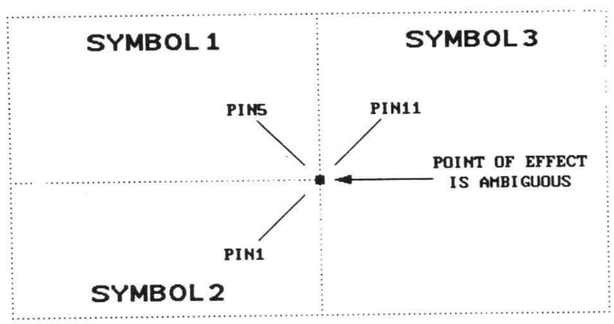
A pin name field can exist outside the symbol as long as the point of effect is at the symbol cell boundary.

Figure 2-25
Multiple Pins on a Single Point of Effect Are Not Supported



No more than two pins located in *different symbols* can share a common point of effect. Figure 2–26 shows an unsupported point of effect, where pin names in SYMBOL1, SYMBOL2, and SYMBOL3 all have their points of effect at the same location in the drawing.

Figure 2-26
Three Pin Names on One Point of Effect Are Not Supported



You can assign the same pin identifier to different pins of a symbol. The example in Figure 2–27 shows that multiple pins of the same symbol are named IN. The post processors will assign the same name to each pin but will maintain them as separate pins when connectivity data is generated.

Libraries

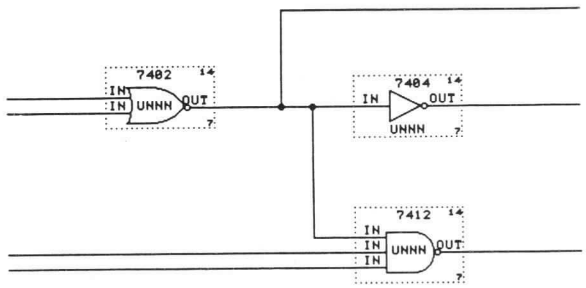


Figure 2-27
Duplicate Pin Names in the Same Symbol Are Supported

Many symbol libraries are included in your FutureNet package. These libraries contain many of the most popular electrical parts including TTL, CMOS, ECL, Intel Microprocessors, programmable logic, and FPGA, and documentation. To get a list of parts in each library, refer to the LIB and .DIRPR commands in the *FutureNet Command Reference,* and to your *Installation Guide.*

System Library

The system symbol library, **system.sym**, is a collection of the most frequently used parts from several of the libraries provided with FutureNet. When you start FutureNet, it searches for **system.sym** in the current directory (unless you have specified something different using the FNLIB environment variable), and accesses it for reference. You cannot write to this library using this filename. If you wish to update it, you must copy it to another filename.

Symbol Library Naming Conventions

The following conventions are used in naming the symbols in the libraries:

- Many IC packages (especially gates and inverters) have multiple elements. These separate elements are labeled with suffixes A to F. To load the 7400 gate element that has pins 1 and 2 as inputs and 3 as output, load symbol 7400A.
- Many parts (especially gates and inverters) have DeMorgan equivalents included in the library. The set of symbols for the 7400 Quad Two-input gate includes eight items. The DeMorgan equivalents are designated with the suffix M. The eight elements in the 7400 set are: 7400A, 7400AM, 7400B, 7400BM, 7400C, 7400CM, 7400D, 7400DM.
- The patterns included in the library correspond to the J and N packages (DIPS). The W and T (Flat Paks) are not included.
- In general, no distinctions are made between standard, low-power, H series, low-power Schottky, Schottky, advanced low-power Schottky, and advanced Schottky logic. There are a few exceptions, such as the 7451 and 74LS51 where the standard and low-power Schottky patterns are different. In these cases, the low-power Schottky part is called by appending an LS; for example, the 74LS51 (A element) is called 74ALS.

- A series of prototype SSI and MSI elements are included in the library. These elements have "placeholders" for pin numbers, pin descriptors, part numbers, and circuit designators. These placeholders have been assigned the proper attributes. To use them to create a new template, enter the appropriate data in the placeholder fields. Protoypes are listed in Table 2–14.

Table 2-14
SSI and MSI Prototype Elements

Part Type	Library Name
2 Input AND:	
AND	AND21
NAND	AND22
DeMorgan Equivalent NOR	AND23
DeMorgan Equivalent OR	AND24
3 Input AND:	
AND	AND31
NAND	AND32
DeMorgan Equivalent of NOR	AND33
DeMorgan Equivalent of OR	AND34
4 Input AND:	
AND	AND41
NAND	AND42
DeMorgan Equivalent of NOR	AND43
DeMorgan Equivalent of OR	AND44
2 Input OR:	
OR	OR21
NOR	OR22
DeMorgan Equivalent of NAND	OR23
DeMorgan Equivalent of AND	OR24
3 Input OR:	
OR	OR31
NOR	OR32
DeMorgan Equivalent of NAND	OR33
DeMorgan Equivalent of AND	OR34
4 Input OR:	
OR	OR41
NOR	OR42
DeMorgan Equivalent of NAND	OR43
DeMorgan Equivalent of AND	OR44
Buffer-Inverter	BUF
D Flip-Flop	DFLOP
JK Flip-Flop	JFLOP
Octal Buffer, 2ENA	MS182
Octal Latch, 1ENA, !CLK	MS191

- Some parts (for example the Intel 8086) have multiple modes where the pins take on different names. In these cases, more than one entry for the part is included in the library. The 8086 in maximum mode is called 8086X, and the 8086 in minimum mode is called 8086M.

- DeMorgan equivalents are included for inverters, buffers, and transceivers. Obviously many other parts have useful DeMorgan equivalents. It is easy to create these by loading the standard parts, enabling FutureNet symbol definition mode, and scanning for the signal stub commands DXL, DXR, DYU, DYL, BXL, BXR, BYU, and BYL, and replacing the Dxxs with Bxxs and the Bxxs with Dxxs as required.
- Some parts need to be user extendable (multiple resistors or diodes in one package, transformers etc.). Graphics macros (IGnn) are provided for easy extension of these types of parts. The FutureNet symbol definition commands allow you to repeat these macros within a block to provide these extendable parts.

Creating Structured Designs

A structured design is a design that has been modularized by function. The modules are then used as building blocks in the design.

Modularizing a design simplifies design management and reduces processing time when design changes occur. Changes to a functional area do not require reprocessing the entire design. Once tested and validated, modules can be used in other designs.

In FutureNet, an area that has been modularized by function is a separate sheet, and a separate drawing sheet is a separate drawing file.

Types of Structured Designs

Two types of structured designs can be created with FutureNet: flat and hierarchical.

Flat designs modularize design functions on separate sheets. Each sheet contains the circuit level detail of a design function. All the sheets required to specify the design create a drawing set.

Hierarchical designs modularize design functions using functional block symbols. A functional block symbol specifies a drawing or drawing set, but shows no circuit detail, only inputs and outputs. The referenced drawing set can also contain functional block symbols that specify other functional areas detailed in other drawing sets.

Drawing Sets and Signal Scope

In FutureNet, a drawing set is the top-level set of drawings, or the drawing(s) named in a functional block. A drawing set can be referenced many times in the design as long as the integrity of the set is maintained. All of the drawings in a set must be named in each functional block symbol that references the set.

Flat Designs

Flat designs consist of one drawing set, with all drawings on the same level. An example of separate sheet modularity is contained in Appendix E.

Hierarchical Designs

Hierarchical designs consist of a root drawing or drawing set at the highest level of the design, with major functional areas represented using functional blocks that show only inputs and outputs. All of the drawings that represent the top-level of a design are referred to as the *root level drawing set*. Successive lower levels of the design show increasing detail until, by the lowest level, all discrete components have been illustrated. There is no limit to the number of occurrences a drawing or drawing set can have or to the levels in which it can be used.

An example of a hierarchical design is contained in Appendix D.

Drawing sets that branch from any point below the root are lower-level drawing sets. The lower-level drawing sets provide detail of the drawing set immediately above where the branch originated. Each lower-level drawing set can consist of one or more drawings.

Functional Blocks

The functional block symbol acts as a placeholder for another drawing or drawing set. These symbols contain no circuitry and show only inputs and outputs. The name of the drawing set represented by the functional block symbol is specified inside the block in displayed alphanumeric fields with the FILE or FILN attribute. An alternate method is to place the filename(s) in layered text that is associated with the displayed alphanumeric field. Each line in the layered text is a filename and the displayed field is treated as comment data. Drawing names follow the operating system naming conventions. If no file extension is specified, FutureNet assigns **.dwg**. Using functional blocks in a design to reference the same drawing set eliminates the need to make multiple copies of sheets that detail identical functions. The FutureNet Post Drawing Preprocessor treats each occurrence of a lower-level drawing as if it were a unique, individually drawn sheet.

More Than One Drawing Name in a Functional Block

A functional block can contain multiple drawing names with the FILE or FILN attribute. This feature allows the functional block symbol to reference two or more drawings. When a functional block has multiple drawing names, each named drawing becomes part of a drawing set. For example, Figure 2–28 shows part of a drawing that contains three functional block symbols. Two of the functional blocks contain multiple drawing names, and, therefore, reference specific drawing sets. One functional block is made up of the drawing set, MEM1, MEM2, and MEM3, while the other block is made up of RAM1, RAM2, and RAM3. The third functional block symbol simply names a single drawing file (MEMDEC) which will replace that functional block.

Any drawing within the drawing set (that is, named with other drawings to a functional block) can only appear in that drawing set. The two functional block symbols shown in Figure 2–29 contain the same drawings as the example in Figure 2–28, except that an attempt has been made to include drawing MEMDEC in both blocks. Assigning MEMDEC to two different drawing sets in this manner will result in a post processor error.

Figure 2-28
Functional Blocks with Multiple Drawings

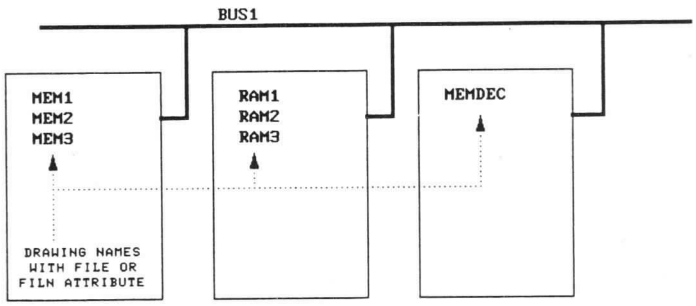
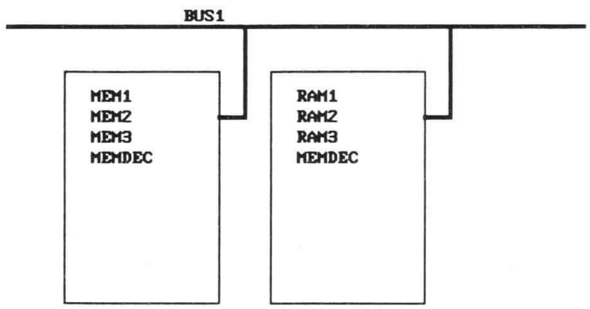
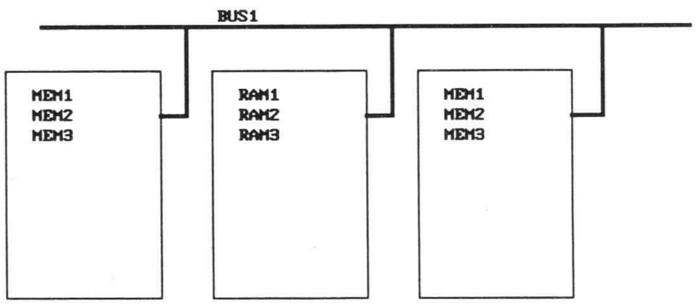


Figure 2-29
Functional Blocks with Identical Drawings



Once a drawing is named in a drawing set, the drawing can only be used as part of that drawing set. That is, you can name the drawing in other functional block symbols as long as all drawings in the set are named. Figure 2–30 shows an example of this, where MEM1, MEM2, and MEM3 are all used in the two outside functional blocks.

Figure 2-30
Functional Blocks with Illegal Drawings



Functional Block Symbols and Pin Names

Mapping signals into and out of a functional block requires that you place a pin identifier field with the point of effect on the symbol boundary. The pin identifier fields should match the names of the signals within the drawing set referenced in the functional block. Each functional block that references the same drawing set must have identical pin identifier fields and pin types (signal pin, bus pin, etc.).

A signal connected to a functional block is connected to the signal in the named drawing set that has the same name as the pin identifier of the functional block symbol. Signal names connected to the functional block do not have to match the pin identifier fields inside the functional block to maintain connectivity.

Figure 2–31 shows an example of an inconsistent pin identifier appearing in the functional blocks. The pin identifiers in one functional block are OUT and CLK1; the pin identifiers in the other functional block (which identifies the same drawings) are OUT and CLK2. To avoid a post processing error, the CLK1 and CLK2 pins must have the same identifier.

Figure 2-31
Violation of Pin Name/Attribute Conventions

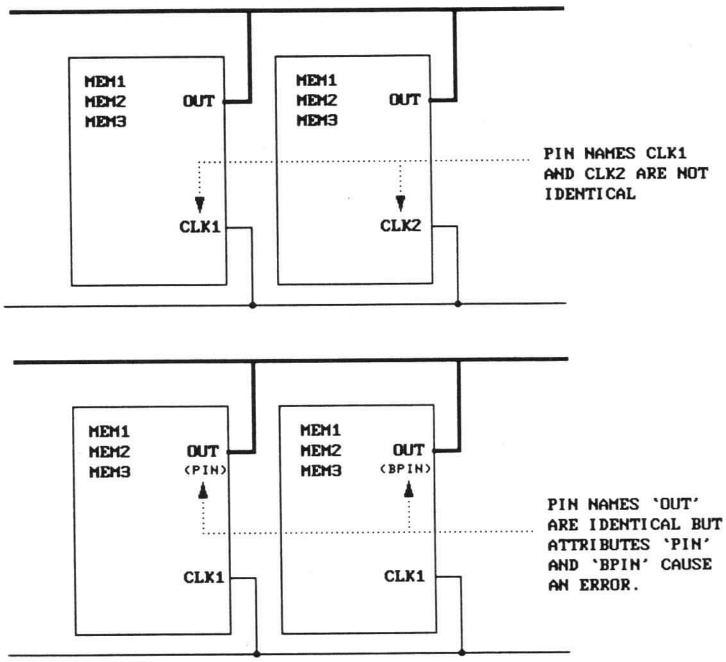


Figure 2–31 also shows an example of an inconsistent pin attribute. Both pins connected to the bus are named OUT. However, the displayed name fields are assigned different attributes. One is attribute PIN (signal pin), the other is BPIN (bus pin). To avoid a post processing error, both must be assigned the same attribute.

The use of pin stubs with functional blocks is optional. Signal mapping from outside the functional block to the drawing or drawing set referenced inside the functional block is the same whether you use pin stubs or not.

Mapping Bus Signals In/Out of the Functional Block

When a bus is connected to a functional block symbol, only the bused (not the packaged) signals are traced in the lower-level drawing. The bused signal names outside the functional block must match the bused signal names inside the functional block.

Hierarchical Design Example

The diagram shown in Figure 2–33 represents the drawing sheets of a structured design for an automobile cruise controller. At the root of the tree, the drawing **scontrol** (Figure 2–32) shows the input and output signals that attach to the controller. Looking at the inputs and outputs of the controller suggests that its design can be partitioned into two distinct sections. The first section is an equality test section to check against a reference speed. The second section is a logic routine to control speed up or slow down. Notice that the detailed composition of the controller itself is not shown in this drawing. Instead, a *functional block symbol* is used.

Figure 2-32
Cruise Controller Design Structure

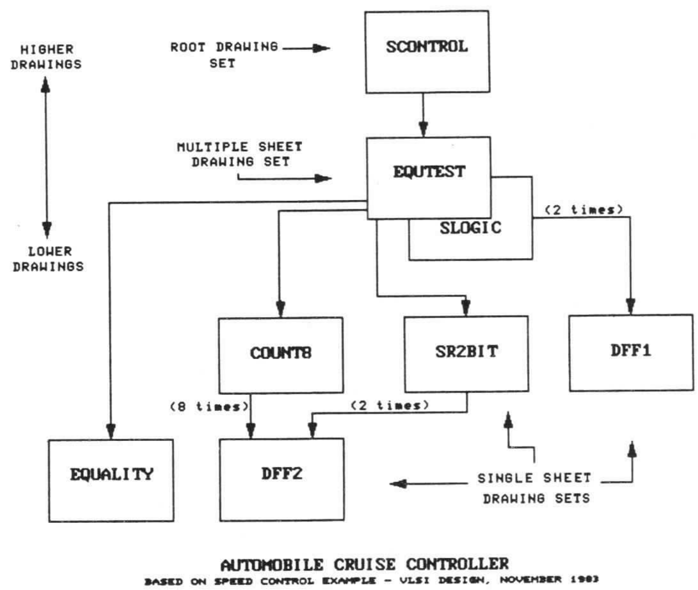
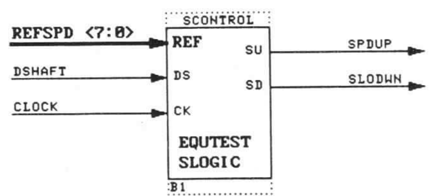


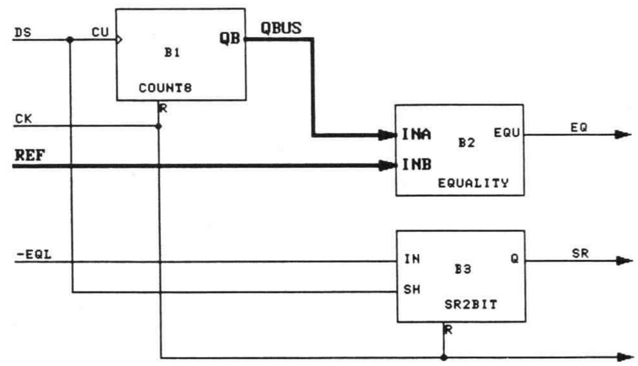
Figure 2-33
Root Drawing **scontrol**



Two of the alphanumeric fields in the functional block symbol contain the filenames **equtest** and **slogic**. These file names are the names of the drawing files that detail the equality test section and the speed control section. Even though they are separate drawings, they have signals in common that imply connectivity. They are both named in the functional block symbol. Together they constitute a single drawing set.

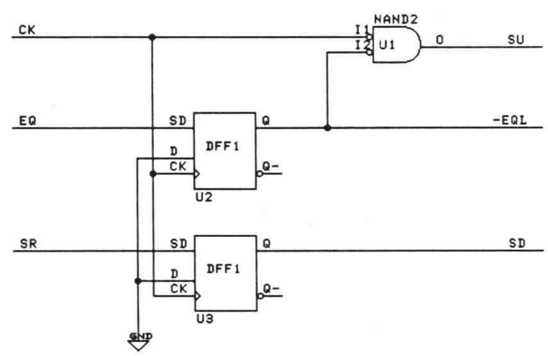
Equtest and **slogic** can themselves contain functional blocks. For example, in **equtest** (Figure 2–34), functional block symbols are used to represent an 8-bit equality checker, an 8-bit counter, and a 2-bit shift register. The contents of these functional block symbols are detailed in the next lower drawing sets, labeled **equality**, **count8**, and **sr2bit**. Because only one drawing name is specified in each of the functional block symbols shown in **equtest**, you can assume that each lower-level drawing set consists of only one drawing.

Figure 2-34
equtest



In **slogic** (Figure 2–35), only one lower-level drawing set is referenced. Again, it is a drawing set that consists of only one drawing.

Figure 2-35
slogic



There are several signal names in **equtest** and **slogic** that are the same. For example, notice that -EQL is an output of **slogic** and an input to **equtest**. Conversely, EQ, SR, and CK are outputs of **equtest** and inputs to **slogic**. This interconnection of common signals makes the two drawings a drawing set.

Still greater degrees of detail can be reached by moving to lower-level drawings to show the discrete logic of the flip-flops used by the counter, the shift register and the logic circuit.

To examine the hierarchical capabilities of FutureNet, refer to the tutorials for more information.

3 *Using FutureNet*

Starting FutureNet

FutureNet must first be installed as described in the *Installation Guide*. You can then start FutureNet from the system prompt by entering

fn [[*path*]*filename*[.ext]]

If you have installed both the standard and extended memory versions, the standard memory version is in the **dataio\fn** directory, and the extended memory version is in the **dataio\fnx** directory (DOS standard installation).

You may include the name of a command file you wish FutureNet to run or a drawing file you wish FutureNet to load. Path information is not needed if the file is in the current directory. The default extensions for command and drawing files are **.cmd** and **.dwg**, respectively. Other extensions must be included in the path information.

Exiting FutureNet

Exit FutureNet by entering the Q command, or selecting QUIT from the main menu. If there is any unsaved work on the drawing screen, FutureNet displays:

```
OK to discard changes that have not already been saved (Y/N)?
```

Press Y to exit without saving the changes. Press N to cancel the command. You may then save the drawing and exit FutureNet.

When you exit FutureNet, the system prompt appears.

The Mouse

The mouse moves the design cursors around the FutureNet window, and the mouse buttons allow you to perform frequently used functions without typing the actual commands on the command line. How the mouse and its buttons function depends on which mode FutureNet is in.

Complete information on FutureNet modes and how the mouse buttons are configured in each mode is contained in Chapter 5, "Mouse, Modes and Cursors."

Window Management

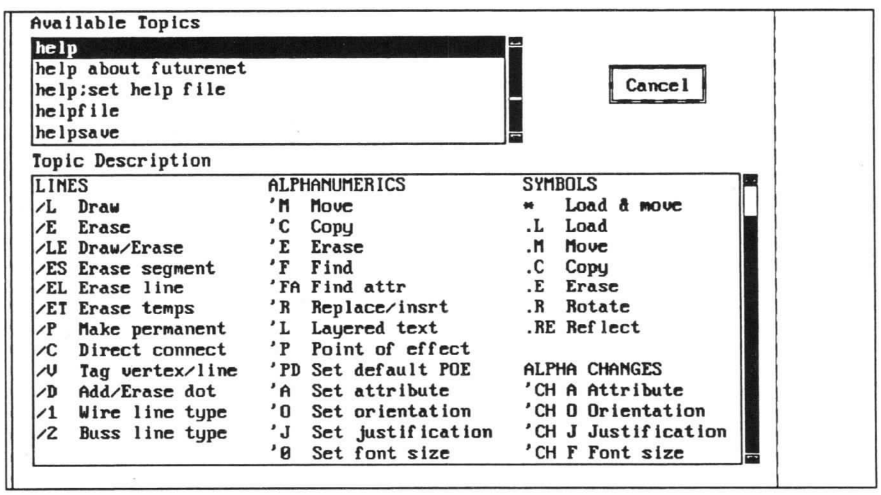
To use FutureNet on Sun workstations, you need to know how to move, resize, open and close Sun windows. See your Sun documentation for this information. In addition, note the following

- When you enter a system command from FutureNet with the ! symbol, the data appears in the window from which FutureNet was started. To see the window, you'll need to make it active by placing the cursor in the header of the window and pressing the left mouse button. When you go back to FutureNet, make the FutureNet window active the same way.
- The FutureNet command line accepts generic Suntools arguments such as window type and placement. Consult the Suntools manuals for more information.

Help

The Help index is available at any time by pressing F1, or by selecting Help from the main menu. The **Help Index** dialog box is shown in Figure 3–1.

Figure 3–1
Help Index Dialog Box



Use the scroll bars to scroll through the index and through the help data box. Select a new item in the index by clicking on it with the mouse.

The Menus

Entering the Menus

FutureNet has an easy-to-use menu system that is accessible through the mouse and/or keyboard. The menus are summarized below:

Menu	Command	Description
Help	help	Indexed help on FutureNet commands.
Command	menu	Used to select commands in the drawing session.
Attribute	'A, 'S, 'CH A	Used to select text attributes.
Symbol Editing	.s, menu	Used to select command in Symbol Definition Mode.

The Command and Symbol editing menus can also be accessed by pressing the right mouse button (in all except FAST drawing mode). Note that in FAST mode, the right mouse button is used to rapidly delete elements of the drawing, so the menus must be accessed by entering MENU on the command line. See Chapter 5 for more information on the MENU and FAST modes.

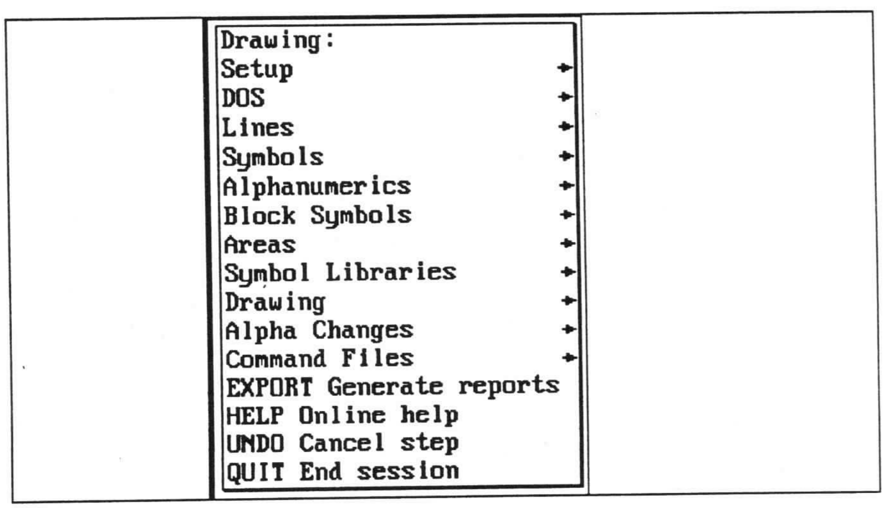
Exiting the Menus

Exit the menus by pressing Esc, or by clicking on an OK or Cancel button in a dialog box. Full operation of the menus is discussed in Chapter 4.

The Command Menu

The command menu contains most of the FutureNet commands organized by type, and includes setup commands, help, and access to the Export menus (described in the *FutureNet Post User Manual*). The Command menu is shown below.

Figure 3–2
Command Menus

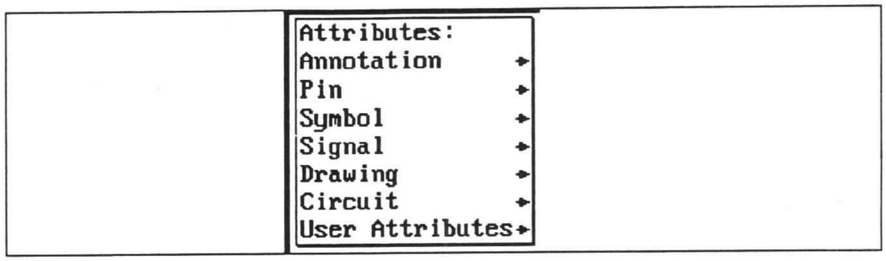


The Attribute Menu

The text attribute menu is accessed by entering an attribute command ('A, 'S and 'CH A) without a parameter. The attribute selected from the menu is then entered as the parameter for the attribute command used to call the menus.

The top-level attribute menu lists attribute classes (see Figure 3–3), and submenus list attributes for the selected attribute class.

Figure 3–3
The Attribute Menu

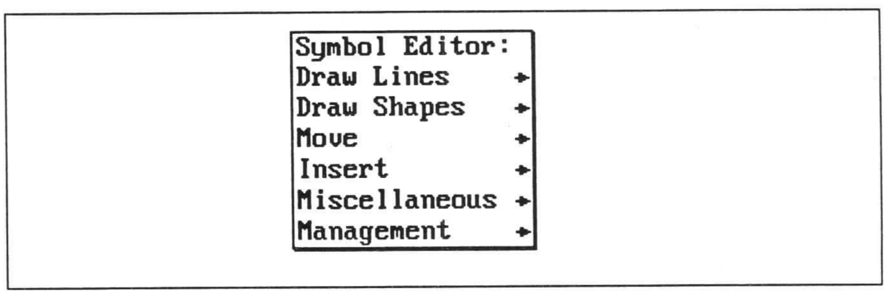


The User Attributes submenu allows you to define your own attributes. The user-definable attributes can be scanned using the scroll bars.

The Symbol Editor Menu

The Symbol Definition menu is accessed in Symbol Definition Mode only (enter **.S**) by pressing the right mouse button, or entering **menu** on the command line. Figure 3–4 shows the Symbol Definition menu.

Figure 3–4
The Symbol Definition Menu



The top-level menu lists overall categories of symbol definition commands, and the submenus list the commands for the selected category.

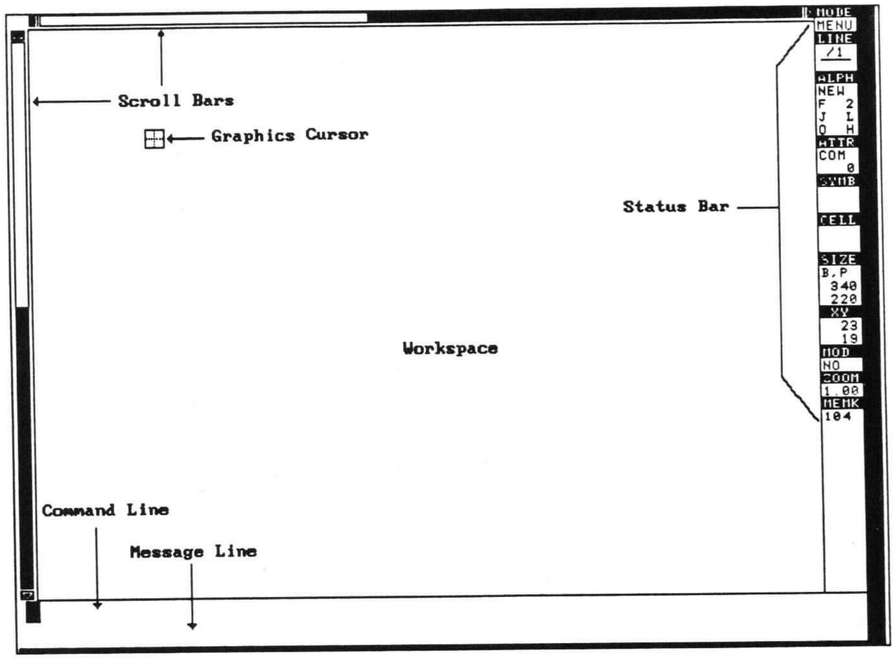
The Drawing Screen

The drawing screen is the design workspace, and is the primary screen used (see Figure 3–5). Its major parts are:

Workspace	Where the drawing appears.
Scroll bars	Used to move the window over the drawing
Command line	Used to run commands from the keyboard
Message line	Displays messages and prompts
Status bar	Operating information

These major parts are discussed here, with the exception of the command line, which is covered in Chapter 4, "The Menus and The Command Line."

Figure 3–5
The Drawing Screen



Workspace

The workspace is the area used for creating the drawing.

Scroll Bars

The scroll bars can be used to move the drawing through the workspace when the drawing does not fit in the workspace at its current zoom level (magnification).

Message Line

Directly below the command line is the message line. Prompts, system messages, and error messages appear here. Messages disappear once a key is pressed.

The Status Bar

The status bar at the right of the screen contains fields showing the current status for each of the functions listed. The fields described below

MODE

The MODE field indicates the current mode. The possible modes are summarized below, and explained in further detail in Chapter 5, "Mouse, Modes and Cursors."

MENU	Menu Mode
FAST	Fast Drawing Mode
LINE	Line Drawing Mode
MOVE	Move Mode (Tag and Drag)
COPY	Copy Mode
ERASE	Erase Mode
AREA	Area Definition Mode
ALPH	Alphanumeric Field Entry Mode
PTR	Point of Effect Tracking Mode
SYMD	Symbol Definition Mode
ZOOM	Dynamic Zoom Mode

LINE

The line field displays the current line type, and a sample of the line type.

The /1 through /10 commands select the line type. Unless redefined by the user, the function key F10 (or F1 on a Sun) cycles through the line types by entering the /0 command, which may also be entered from the command line. The default value is /1.

In Symbol Definition Mode, the current line setting is defined by the SL command. The line type ranges from SL 0 to SL 14. Only the BX, BY, BXD and BYD commands look at this setting when drawing a line.

ALPH
printability
F *font*
O *orientation*
J *justification*

The ALPH field displays the default settings for new alphanumeric fields when the cursor is not in an alphanumeric field, and displays the settings for the current fields when the cursor is within an alphanumeric field. This field is not applicable in Symbol Definition mode.

- **Printability** — The 'CH P command selects the printability of alphanumeric fields. Any alphanumeric field can be made printable or non-printable. The printability of an existing alphanumeric field is displayed as OLD for printable or ONP for non-printable. NEW is displayed if the cursor is not on an alphanumeric field.

- **Font** — The commands '1, '2, '3, '4, '5, '6 and '7 select the character size. The function key F9 (default) cycles through the font types ('0 command). '0 can also be entered from the command line. The default font value is '2. The font type is displayed as

  `F` *`font_type`*

  'CH F # is used to change the font size of an existing alphanumeric field.

- **Orientation** — The orientation is displayed as

  `O` *`orientation`*

  The command 'O [H | V] changes the orientation for new fields; 'CH O [H | V] sets the orientation of existing fields. The default value is horizontal.

- **Justification** — The justification is displayed as

  `J` *`justification`*

  The command 'J [L | C | R] sets the default justification for new fields. The command 'CH J [L | C | R] changes the justification of an existing field. Entering 'CH J without a parameter cycles through the options (L, R and C). The default value is left.

ATTR

This field displays the attribute assigned to the alphanumeric field at the cursor position. This field is not applicable in Symbol Definition mode.

When the cursor is not in an alphanumeric field, the attribute name and number assigned to new fields are displayed. When the cursor is in an existing alphanumeric field, the attribute mnemonic and number for the field are displayed.

- The command 'A *name* or *number* sets the default attribute for new fields. If no parameter is supplied, the attributes menu appears at the current cursor location.
- The command 'CH A *name* or *number* changes the attribute for an existing field. If no parameter is supplied, the attributes menu appears at the current cursor location.

Attributes are discussed in depth in Chapter 2, "Understanding FutureNet."

SYMB
rotation
reflection

This field indicates whether the symbol at the graphics cursor location has been rotated and reflected. In Symbol Definition mode, note that the symbol is always edited in its non-rotated and non-reflected state.

This field is blank when the graphics cursor is not in a symbol cell.

- The upper field shows the symbol rotation, noted as an R followed by the degree of rotation in 90-degree increments (using a standard 360-degree base). If the symbol is not rotated, this field is blank.
- The lower field shows if the symbol has been reflected, noted with REFL. This field is blank if the symbols has not been reflected using the reflect command (.RE).

CELL

This field gives the width and height of the symbol in display units.

This field is blank when the graphics cursor is not in a symbol cell.

The cell size is set by the Define Symbol Cell Size (*.w,h*) command in Symbol Definition mode, or by the Create Block Symbol (.B) or Create Functional Block Symbol (.F) commands in drawing mode (MENU or FAST) . The cell size for existing symbols can also be modified in drawing mode using the Define Symbol Cell Size command.

SIZE
sheet, printer_comp
width_in_du
height_in_du

This field displays the size of the drawing and the printer it has been adjusted for. This field is not applicable in Symbol Definition mode.

- **Sheet, printer_comp** — The first line of this field contains two entries. The first entry gives the standard size of the drawing in sheet size, such as B. If a SIZE command is given using inches, millimeters or display units to define the drawing width and height, then the first entry will be blank. The second entry gives the printer the drawing size has been adjusted for.
- **Width and Height** — The second and third lines of this field give the width and height of the drawing in display units. The default is 340 wide by 220 display units high, which equals 17 by 11 inches when printed (the dimensions of a standard B-size drawing).

The SIZE command sets the drawing width and height. (See the *FutureNet Command Reference* for more information.)

XY
x,y

This field gives the x,y coordinates for the graphics cursor in display units. In Symbol Definition mode, the x,y coordinates are in pixels, and this field shows the coordinates of each instruction as it is highlighted by the cursor.

An *x* value of 0 corresponds to the left edge of the drawing and increases as the cursor is moved right across the screen. A *y* value of 0 corresponds to the top edge of the drawing, and increases as the cursor is moved down.

MOD

This field indicates whether unsaved modifications exist for the current drawing.

YES indicates the SAVE command should be entered to save changes to the drawing before exiting. Only changes to the graphic content of the drawing affect this field.

ZOOM

This field displays the zoom level of the drawing display. This field is not applicable in Symbol Definition mode.

There are four zoom levels in FutureNet: fit, intermediate, half and full. The zoom level determines the scale of the display. Three commands — ZIN (zoom in), ZOUT (zoom out) and ZOOM (dynamic zoom) — affect the zoom level. See ZIN/ZOUT and ZOOM in the *FutureNet Command Reference* for more information.

MEMK

This field is displayed only in the DOS Version of FutureNet, and displays the amount of free drawing space left in kilobytes.

Moving Around in Drawings

You can move around in a drawing in the following ways.

HOME

Enter the HOME command to center the workspace around the current cursor location (*x,y*).

Scroll Bars

Use the scroll bars on the left and top of the screen to move around in the drawing workspace.

- To move the drawing incrementally through the workspace, place the cursor on the empty (black) portion of a scroll bar and press the left mouse button.
- To move to a specific area of the workspace, place the cursor on the highlighted portion of the scroll bar, press and hold the left mouse button, move the gray scroll bar, and release the left button.

Note: If you are dragging the scroll bar, moving the cursor out of the FutureNet window cancels the operation.

Mouse

If AUTOPAN is ON, display new areas of the drawing by using the mouse to move the cursor past the edge of the drawing workspace. As the cursor passes the edge of the workspace, the drawing shifts in that direction to center the workspace around the cursor. See the *FutureNet Command Reference* for more information on AUTOPAN.

Cursor

If AUTOPAN is ON, display new areas of the drawing by using the arrow keys to move the cursor past the edge of the workspace. As the cursor passes the edge of the workspace, the drawing shifts to center the workspace around the cursor.

Function Keys

You can set the **F** function keys to frequently-used functions. The default values are

F1	For DOS versions, runs Help. For Sun versions, runs the /0 command, which cycles through the line types, setting a new default line type.
F2	Exits FutureNet.
F3	Runs the 'P command, which repositions the point of effect for the alphanumeric text field.
F4	Runs the 'L command, which edits layered text.
F5	Runs the /D command, which toggles a connect dot at the graphics cursor when on a type /1 or /2 line.
F6	Runs the /ES command, which erases the line segment at the graphics cursor.
F7	Runs the /P command, which makes the temporary line at the cursor permanent.
F8	Runs the /ET command, which erases all temporary lines on the drawing.
F9	Runs the '0 command, which cycles through the font sizes.
F10	Runs the /0 command which cycles through the line types, setting a new default line type. (Use **F1** on the SUN.)

Setting Function Keys

Function keys are set in the Profile screen, or with the KEY command. Function keys can be assigned any FutureNet commands up to 127 characters. Although the default configuration has one command per key, you can assign a set of commands to create command macros. The section, "Using the FutureNet Command Language" discusses editing and creating custom command macros.

Viewing F Key Assignments

You can view and change **F** key assignments in the "Function Keys" section of the Profile screen. See also the section, "Using the FutureNet Command Language."

The Profile Screen

A number of important FutureNet functions have been arranged on a screen called the Profile screen, where you view and edit certain environment parameters.

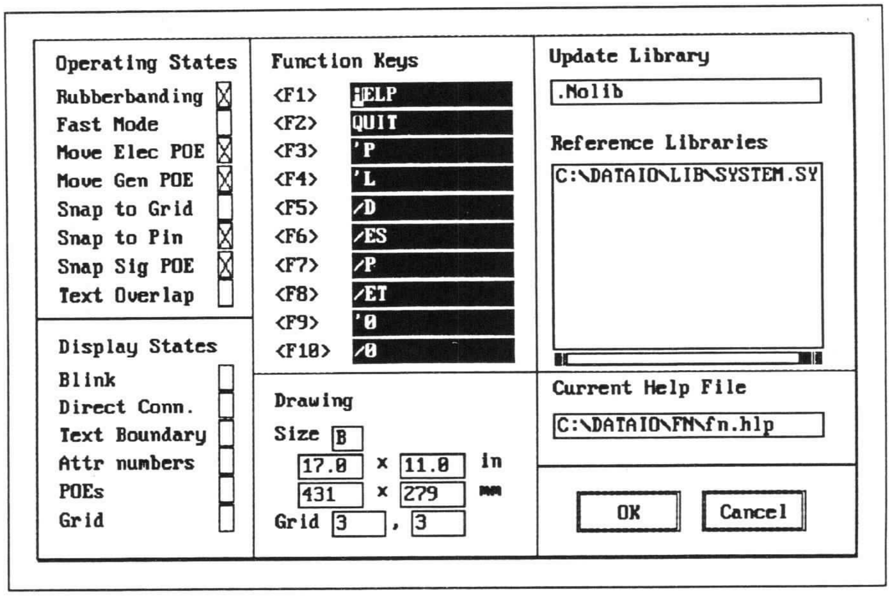
Accessing

The Profile screen is accessed from the command line by entering

profile

on the command line, or from the menu by selecting **Setup: Profile.**

Figure 3–6
The Profile Screen



Screen Areas

Each area of the Profile screen is described below. The related commands are given in parentheses following the description.

Operating States

☐ **Rubberbanding** — Default: ON. Specifies if connections are maintained when you move areas or symbols by rubberbanding the signal lines. (CONNECT)

☐ **Fast Mode** — Default: OFF. Specifies if FAST is the default mode. If OFF, MENU is the default mode. (FAST)

☐ **Move Elec POE** — Default: ON. Specifies if points of effect for electrical attributes (that is, signals and pins) move with the alphanumeric field to which they belong when the fields are moved. (MOVEPOEE)

☐ **Move Gen POE** — Default: ON. Specifies if points of effect for general attributes (not signals or pins) move with the alphanumeric field to which they belong when the fields are moved. (MOVEPOEG)

	☐ **Snap to Grid**	Default: OFF. Specifies if the drawing element being added to a grid snaps to the closest grid point in the right and downward direction. When on, drawing elements can only be added at grid points. (GRID SNAP)
	☐ **Snap to Pin**	Default: ON. Specifies if a line snaps to a pin's point of effect when it is within 2 display units. When ON, lines outside of symbols snap to a pin's point of effect. (PINSNAP)
	☐ **Snap Sig POE**	Default: ON. Specifies if the point of effect range is extended. When ON, points of effect for alphanumeric fields with signal attributes snap to the closest type /1 or /2 line if the alphanumeric field is within 10 display units. (POER)
	☐ **Text Overlap**	Default: OFF. Specifies if alphanumeric fields can overlap symbol boundaries. (OVERLAP)
Display States	☐ **Blink**	Default: ON (monochrome), OFF (color). Specifies if the zoom window, highlighted alphanumeric fields, and tagged graphics blink. (BLINK)
	☐ **Direct Conn.**	Default: OFF. Specifies if direct connections are displayed. When ON, only direct connections and symbol cell boundaries are visible. (.DCON)
	☐ **Text Boundary**	Default: OFF. Specifies if the boundaries of all alphanumeric fields are displayed. When ON, the dotted boundaries and justification points for all alphanumeric fields are displayed. ('B)
	☐ **Attr Numbers**	Default: OFF. Specifies if the attributes of all alphanumeric fields are displayed. When ON, the attribute number of each alphanumeric field, rather than the text itself, is displayed, and is highlighted. ('D)
	☐ **POEs**	Default: OFF. Specifies if all points of effect are displayed. When ON, all points of effect for all alphanumeric fields are displayed, and points of effect which do not reside with their text field are linked with a "rubberband." (POEDISP)
	☐ **Grid**	Default: OFF. Specifies if the grid defined in the workspace is displayed. (GRID)

Function Keys

This area on the Profile screen shows the function assigned to each F key. For complete information on using the function keys, see the section "Function Keys." These fields can be edited in the Profile dialog box, or entered with the KEY command on the FutureNet command line.

Drawing

Size	Indicates the size of the current drawing. You cannot edit the Size fields directly. They can be changed with the SIZE, SIZED and SIZEM commands on the FutureNet command line. (SIZE, SIZED, SIZEM)
Grid	Indicates the current grid setting. You cannot edit this field directly. It can be changed with the GRID command. (GRID)

Libraries

Update Library	When an update library has been opened, this field displays the file specification of the update library. You cannot edit these fields directly. They can be changed with the .LIB and .NOLIB commands. (.LIB, .NOLIB)
Reference Libraries	Displays the file specifications for the reference libraries. Use the scroll bars to reveal long names, or to view additional reference libraries. You cannot directly edit these fields. They can be changed with the LIB and NOLIB commands. (LIB, NOLIB)

Current Help File

Displays the current command reference file used by the HELP command. It can be changed with the HELPFILE command. (HELPFILE)

OK and Cancel

Selecting **OK** exits the Profile screen and saves any changes you have made. Selecting **Cancel** exits the Profile screen without saving changes.

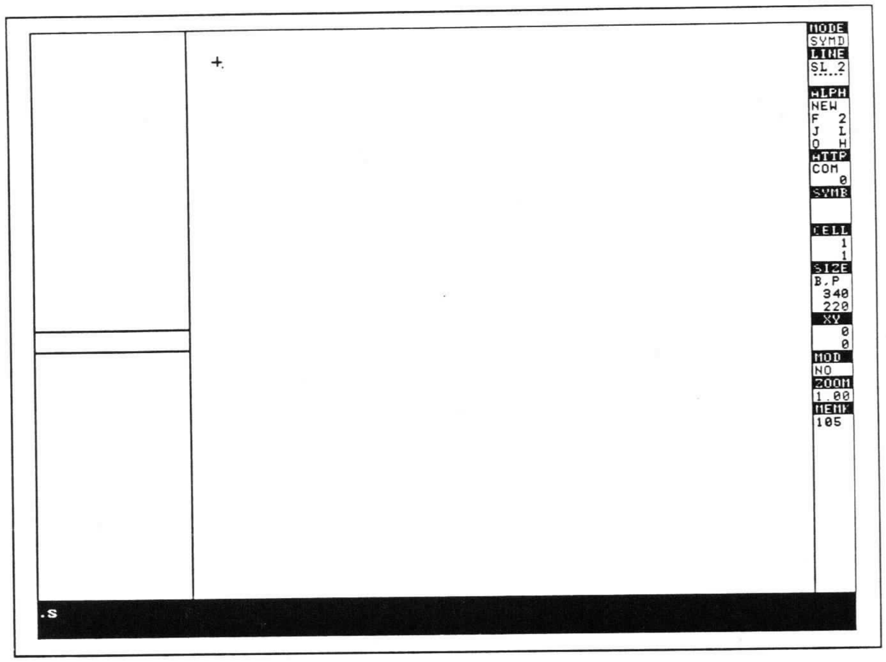
Note: You can also press **Esc** *to exit the Profile screen without saving changes.*

The Symbol Definition Screen

The Symbol Definition screen is used to build symbols that cannot be built within a drawing. In addition to the workspace, status bar and command line found on the drawing screen, this screen has an area on the left that displays the symbol definition instructions you enter. The target line in the center shows the current instruction.

The symbol definition screen appears when the .S command is entered from the drawing screen command line. Figure 3–7 shows the symbol definition screen with the top-level menu displayed. The symbol definition menu is discussed later in this section.

Figure 3–7
The Symbol Definition Screen



The biggest portion of the screen is the workspace, which displays the symbol being created. The cursor position corresponds to the ending coordinates of the last instruction in the instruction list.

Use the command line to enter

- Instructions that go on the target line
- Commands that manipulate the target line
- .Q to exit Symbol Definition Mode

See the Command Reference Card at the end of the *FutureNet Command Reference* for a complete list of symbol definition commands.

The status bar shows the current status for each of the functions listed, as it does for the drawing workspace. In the Symbol Editor, the following fields are used:

- **MODE** — SYMD for symbol definition mode.
- **LINE** — Displays the SL line type.
- **SYMB** — Indicates reflection or rotation.
- **CELL** — Width and height of the symbol in display units.
- **XY** — Displays the XY coordinates in pixels.

See "The Status Bar" earlier in this chapter for a full description of the status bar fields.

The Symbol Definition Menu

The Symbol Definition menu is accessed by pressing the right mouse button, or entering **menu** on the command line. The top-level menu lists overall categories of symbol definition commands, and the submenus list the commands for the selected category.

Using the FutureNet Command Language

The FutureNet command language is used to automate FutureNet functions. The command language consists primarily of familiar FutureNet commands. It is used in three ways:

Command Lists — A number of FutureNet commands may be combined on the command line. These are called command lists.

Macros — Command lists may be assigned to **F** keys, making customized macros always available. Command lists assigned to function keys are called macros.

Command Files — Command files contain sequences of commands used during the drawing process. Command files (.cmd) are ASCII text files, which can be written by hand, but are more often either the output of programs such as ABEL, or the result of automatic logging of FutureNet work sessions (AUTOLOG).

These features are often combined in various forms; one of the most common uses is that of the **fnpro.cmd** file. Because the commands in this file are run each time FutureNet is started, the file can be used to customize FutureNet. Things that might be included are: **F** key assignments, colors, and libraries to be opened. Information on the **fnpro.cmd** file is contained in a later section of this chapter.

Before using the command language, you will need to become familiar with its syntax, and with the methods used to translate key presses into corresponding commands.

Command Syntax

Commands may be entered in either upper or lowercase. Command files that result from logging work sessions or are written by other programs are written in uppercase.

In all cases, the syntax of the commands is identical to that used on the command line. See the *FutureNet Command Reference* for the syntax of all commands.

The Command List

A command list consists of two or more commands separated by semicolons and typed on the command line. The rules for writing command lists are given below:

- The command line can be up to 256 characters (the length of the command line). For function key macros, the limit is 127 characters.
- Commands must contain all required parameters.
- Commands are run from left to right.
- The only syntactical device used is the semicolon (;) which separates and enters commands.

For example, to add all of the pin stubs to one side of a symbol, with each pin stub 3 display units below the previous pin stub, the following command list would be typed on the command line:

.- ; DOWN 3

Each time [↵] or the middle mouse button is pressed, a pin stub is drawn at the current cursor location and the cursor moves down 3 display units, where the command list can add the next pin stub.

Key and Mouse Press Translations

Keys and all mouse button presses cannot be entered in the command lists or files you write—they must be entered in a command equivalent. These are summarized in Table 3–1.

Table 3–1
Key Press Translations for Command Files and AUTOLOG

Key	Command Equivalent
Arrow Keys	LEFT, RIGHT, UP and DOWN. These commands move the cursor one display unit or the specified number of display units, for example, RIGHT 30 moves the cursor right 30 display units.
Mouse Buttons	!LB (left button), !MB (middle button) and !RB (right button).
[PgUp] or [R9]	ZIN. Zooms in, cycling through the available zooms (0, 1, 2, n).
[PgDn] or [R15]	ZOUT. Zooms out, cycling through the available zooms (0, 1, 2, n).
[Esc]	'R *text*. Used to enter alphanumeric text.
[Home] or [R7]	HOME. Centers the display around the current cursor location.
[Ctrl] – [U]	'UNDER. Adds an underscore to the current alphanumeric field.
[Ctrl] – [O]	'OVER. Adds an overscore to the current alphanumeric field.

Event Sequences

When writing a command list, it is important to understand the sequence of events that make up a command, because actions taken at the keyboard and with the mouse must be translated into equivalent commands.

For example, drawing a straight line using the mouse is done in this manner:

Action	Function
Press left mouse button	Begin line drawing.
Move mouse to line end point.	Draw line left 50 display units
Press left mouse button	Set routing segment in place.
Press left mouse button	Exit line drawing.

The same event sequence translated for use in a command list would be

/L;LEFT 50;/L;/L

Command	Function
/L;	Enter the line drawing command to begin line drawing (the semicolon is the command separator).
LEFT 50;	Move the line drawing cursor 50 display units to the left.
/L;	Enter the line drawing command to set the routing segment in place.
/L	Enter the line drawing command to exit line drawing (the final command does not require a semicolon).

When writing a command list describing an event sequence, you re-enter the command in the places where you would have pressed the left mouse button.

Note: An alternative is to use the command equivalent to the mouse button presses (!LB, !MB and !RB) if the result is unambiguous. If the result might be ambiguous (for example, in a place where you have to press the left mouse button repeatedly until the desired item is tagged), using the specific command is recommended.

Only two exceptions exist to this rule, and they are ROTATE and REFLECT operations. Sequences involving these operations are completed by entering the related MOVE command. For example, if you enter [R 180 you must then enter [M to complete the operation; if you enter .RE you must then enter .M to complete the operation.

Command Macros and Function Keys

You can program often-used command macros into the function keys using command lists.

Assigning Command Macros

To assign a macro to a function key, enter the KEY command followed by a key assignment and a command list. The syntax is:

KEY <F*n*> *commandlist*

where *n* is the number of the **F** key to which the macro is assigned, and *commandlist* is a command or command list to assign to the function key. The brackets on the <F*n*> statement are required, and semicolons must separate commands in the command list. The macro can be up to 127 characters in length.

The easiest way to program a function key is to write the command list and test it, debugging where necessary. Once it is correct, you insert the KEY <F*n*> statement at the beginning of the line and press ↵ .
Function keys can also be assigned in the Profile screen.

Verifying Command Macros

Once you have assigned an **F** key, you may verify its contents by viewing the Profile screen.

If you have assigned a large macro, you can scroll through the entry field to see it all. You can edit it in the Profile screen.

To view the macro in the workspace, enter:

KEY <F*n*>

The complete macro is displayed on the command line. The cursor is placed at the beginning of the macro, and you can edit it. You are prompted:

`Enter command(s)`

Once you have finished edit the command list, press ↵

to assign the revised command list to the function key.

Saving Macros

When you exit FutureNet, any macros you have assigned during the session are lost unless you write them to a command file (see below). The command file can then be run when you start FutureNet, or any time during the editing session.

You may also include them in a file called **fnpro.cmd**. Refer to the section "Using Startup Files," later in this chapter.

Command Files

You can use command files to automatically build drawings or symbols. They contain FutureNet commands and may be as large as required.

When you run a command file, FutureNet performs each command in order. There are several commands used only in controlling and manipulating command files.

Running Command Files

Four commands are used on the FutureNet command line when working with command files: AUTO, EXEC, PAUSE and STOP.

The AUTO Command

The AUTO command runs the entire command file with no user action required. The command is in the following format:

AUTO [*path*] *filename* [*.ext*]

where *filename* is the name of the file you want to run, *path* is the directory containing the file, and *.ext* is the optional filename extension. The default extension is **.cmd**.

When you first run AUTO, you must supply the filename of the command file you want to run. If you resume after pausing, you don't have to respecify the filename.

AUTOEX is a synonym for AUTO.

The EXEC Command

The EXEC command runs the commands in a command file in single-step mode. Each command in the file is displayed on the command line and you must press ↵ to run it, then the next command is displayed. The command is in the following format:

EXEC [*path*] *filename* [*.ext*]

where *filename* is the name of the file you want to run, *path* is the directory containing the file, and *.ext* is the optional name extension. The default extension is **.cmd**.

When you first run EXEC, you must supply the *filename* of the command file you want to run.

To change from EXEC to AUTO, enter the AUTO command.

Note: The command displayed on the command line is the one that will be run the next time **Enter** *is pressed, not the one just run.*

The PAUSE Command

The PAUSE command changes command file execution initiated with the AUTO command to single-step execution, as if it had been started using the EXEC command. Each line in the file appears on the command line for you to run by pressing [↵]. The **PAUSE** command is entered when a command file is running.

The STOP Command

The STOP command ends the command file run started by AUTO or EXEC. The **STOP** command is entered when a command file is running.

Automatically Logging Drawing Sessions

A schematic editing session can be recorded in a named file and played back at a later time to reconstruct the changes made during the session. This feature is called AutoLog.

File Format

Since a drawing is generated when the AUTOLOG command file is played back, the drawing can be altered by editing the command file. The file is created in ASCII format and can be edited with any text editor that maintains ASCII format. You can examine and alter the file as a means of altering the drawing that results from running the AUTOLOG file. However, for the sake of brevity the AUTOLOG feature examines strings of keystrokes and in several instances substitutes a single command string. Entering alphanumeric fields in a drawing is a good example.

Alphanumeric Field Entries

When an alphanumeric entries is made by pressing the [Esc] key, AUTOLOG does not record the [Esc] keystroke, nor the subsequent text entry keystrokes. Instead, the keystrokes are evaluated and replaced with a 'R *string* command. For example, if you press the following keys to enter the text HELLO:

[Esc] **H E L L O** [Esc]

the single command string

'R HELLO

is placed in the command file.

Layered Text Entries

During the schematic editing session, text entries that are made using the 'L command (that is, layered text entries) appear in the command file under a GETLAYERS command. The GETLAYERS command is valid only in a command file and is followed by a decimal number that indicates the number of text lines that make up the layered text entry.

Key Press Translations

Key and mouse button presses corresponding to FutureNet functions cannot be written to autologged command files. Instead, they are translated into a command equivalent. These are summarized in Table 3–1 earlier in this chapter.

An example AUTOLOG command file for copying a symbol using the mouse is shown below:

```
CURSOR 0 0
!LB
!LB
!MB
CURSOR 15,67
!LB
```

AutoLog Entry	Result
CURSOR 0 0	The location of the original symbol to be copied.
!LB	Line drawing cursor appears.
!LB	The symbol is tagged.
!MB	The mode is changed to COPY.
CURSOR 15,67	The location of the copy destination is given.
!LB	The copy operation is confirmed.

Reporting Event Locations

Actions performed with FutureNet can be generally separated into two categories: those that change the graphic content of the drawing and those that do not.

Commands change the drawing change it at a given point, and the location of the cursor when the command is entered is important. For this reason, all commands of this type automatically report the cursor position before they run. For instance, if you look at the example above, you'll notice that the first and fifth commands are CURSOR commands, giving an absolute location within the drawing. The system adds the cursor commands before it enters the requested commands.

Commands that do not change the graphic content of the drawing do not require the cursor to be in a specific place, so the command file contains only the requested command. Examples of commands that do not change the drawing are changing the default attribute or printing the drawing.

Enabling AUTOLOG

The section on environment variables in your *Installation Guide* gives complete details on activating and controlling the AUTOLOG feature.

Using a Startup File

Each time FutureNet is started, it automatically looks for a file called **fnpro.cmd**. This file of commands is run as part of the FutureNet initialization process, before any other command or command file is run. This file makes sure that FutureNet initializes each time in a known state, and can be customized for your needs. Any FutureNet command may be included in this file. For instance, the **fnpro.cmd** file shown on the left sets the conditions shown on the right each time FutureNet is started.

Blink OFF	Disable blinking.
PrintOpt NoFit	Do not scale drawings to fit the printer paper
PrintOpt Legal	Set printer paper size to LEGAL.
Profile	Display the Profile screen (to verify your Profile each time you start FutureNet).

If you specify a command file on startup, it runs after the **fnpro.cmd** file.

The section on environment variables in your *Installation Guide* gives complete details on installing the **fnpro** feature.

Customizing the Startup File

You can customize the startup file as follows:

- You can rename the file (it doesn't have to be called **fnpro.cmd**).
- If you include a full path name, you can access the same **fnpro.cmd** file regardless of the directory from which you start FutureNet.

Custom Menus and Help Screens

The following sections describe how to create custom menus and help screens.

Note: The HELP command runs the online Command Reference. Use the "?" command to run the Custom Menu facility.

How the Custom Menu and Help Screen Mode Works

In order to create custom menus and help screens, you need to understand how the Custom Menu and Help Screen feature works.

When the ? command is entered and the Custom Menu or Help Screen is loaded, the cursor only moves between a few special fields. These fields are called menu selections and are defined when the help screen is created. How they are defined is explained later in this section.

Each menu selection should have an associated command that can be run by highlighting the menu selection, and pressing the left mouse button or pressing the ↵ key. The command associated with a menu selections can be the ? command to load a new menu, or any other FutureNet command. If the command is not the ? command, then the Custom Menu mode is closed and the selected command is run in drawing mode. How a command is associated with a menu selection is discussed later in this section.

Custom menus can have other text and graphics in addition to the menu selections. The additional text and graphics can be used to convey more clearly the use of the menu or to provide help. How to enter text and graphics in a custom menu is explained later in this document.

How to Create Custom Menus and Help Screens

Follow the steps below to create custom menus and help screens.

1. Using the guidelines listed below, create custom menus and help screens in the drawing mode just like any other drawing.
    - The custom menu or help screen must be composed in the upper left corner of the drawing space.
    - The size of the menu must not exceed one screen at zoom level 1 (Full Zoom) for the lowest resolution monitor that the menu will be used on. For example, a single screen at zoom level 1 on an EGA monitor will display 146 display units in the X direction and 75 display units in the Y direction.
2. Text fields in your custom menu or help screen have the following characteristics:
    - Attribute (either attribute 1 or attribute 2)
    - Visible or invisible
    - Menu selection and/or associated command

    Follow the steps below to create a text field and determine if it is a menu selection and/or associated command.

    a. To create a text field that is both a visible menu selection and the associated command

    1) Enter a FutureNet command;
    2) Place the cursor on the text field and set the attribute to 1 using the 'CH A 1 command.

    b. To create two text fields, one that is an invisible associated command and one that is the visible menu selection

    1) Enter the FutureNet command that is to be the associated command;
    2) Place the cursor on the text field and set the attribute to 1 using the 'CH A 1 command;
    3) Make the field invisible using the 'CH V OFF command.

4) If the next field (directly to the right or directly below) is attribute 2, then it becomes the menu selection for the associated command. To create a different menu selection for the associated command continue with steps 5 and 6.

5) Enter the desired text directly to the right or below the associated command (no other text fields can be between this menu selection and the associated command);

6) Place the cursor on the text field and assign attribute 2 to the field using the 'CH A 2 command.

Note: *The 'B (Display Alphanumeric Boundaries) command is useful in laying out the menu selections since the boundaries of invisible fields are displayed.*

Cursor Bar Movement

The movement of the cursor bar between the menu selections is controlled by the location of the command fields (text fields with attribute 1) to allow flexibility in laying out menu selections. Since the command fields can be invisible, the location of the command field does not affect the appearance of the menu.

The cursor bar can move in four directions: left, right, up and down. The movement of the cursor with the mouse or the arrow keys is controlled by the following rules:

- For a given arrow key or corresponding mouse movement the cursor bar moves in the direction of the arrow, if possible to the menu selection of a command field that lies in the direction of the arrow or mouse movement.

  If several command fields lie in that direction, then the following criteria are used to determine the next menu selection:

  1. The cursor bar moves to the adjacent field in the direction of movement. A field is adjacent if either both ends of the candidate field overlap the current field or if both ends of the candidate field are overlapped by the current field. Note that for up and down movement, the left and right ends are used for comparison; for left and right, the top and bottom are used for comparison. If several fields satisfy this criteria, then choose the field which is closest to the current field in the direction of movement.

  2. If no fields are adjacent, then the cursor bar moves to the field which is closest to the current field in the direction of movement.

Other Features

- **FutureNet Symbols** — You can load a FutureNet symbol into a custom menu or help screen, and change the attribute of all text fields with attribute 1 or 2 (so the text fields are not seen as menu selections and commands). Use the 'FA (Find Attribute) command to find fields of a particular attribute.
- You can make text fields display in reverse video by using the 'CH R command. This displays the text field in reverse video at all times (most fields are displayed in reverse video when selected). This can be used to accentuate important labels, keywords or other special text.

Saving the Custom Menus

Save custom menus using the HELPSAVE command, for example,

HELPSAVE "menu1"

The custom menu is saved in the current update library as a symbol with the name **"?menu1"**.

Entering and Exiting Custom Menus

You can enter a custom menu with the ? command, for example,

? "menu1"

See the *FutureNet Command Reference* for more information on the ? command.

You can exit a custom menu by pressing the right or middle mouse button or the [Esc] key.

4 *Menus and The Command Line*

The Menus

Entering the Menus

FutureNet has an easy-to-use menu system that is accessible through the mouse and/or keyboard. The menus are summarized below:

Menu	Command	Description
Help	help	Indexed help on FutureNet commands.
Command	menu	Used to select commands in the drawing session.
Attribute	'A, 'S, 'CH A	Used to select text attributes.
Symbol Editing	.s, menu	Used to select command in Symbol Definition Mode.

The Command and Symbol editing menus can also be accessed by pressing the right mouse button (in all except FAST drawing mode).

Submenus and Executable Selections

If an arrow appears to the right of a menu selection, it is not directly executable, but instead calls up a submenu. If there is no arrow, then the menu selection takes you into a new menu set, or places the selected command on the FutureNet command line. If parameters are required for the command, then a message on the status line prompts for parameters. If no parameters are required, the command runs. To select from the menu, place the menu selection bar over the command and press the left mouse button or ↵ . The command operates on the cursor position when the menus were accessed.

Moving Through the Menus

To move up or down in the menus, highlight the desired menu selection and press the right mouse button.

Selecting from the Menus

To select an item from the menus, highlight an executable command or the desired option and press the left mouse button. If an executable command is selected, the menus disappear and the command is entered on the command line.

Exiting a Menu

To exit a menu, move the cursor clear of the menus and press any mouse button. Your cursor snaps back to its location when the menus were accessed. Pressing Esc , or selecting an executable command also exits a menu.

Mouse Operation

The mouse is used to traverse the menus and run commands. Highlight an item by placing the mouse cursor (a small arrow) on the item to be selected. In general,

- The **right mouse button** enters and exits the menu system and moves up or down in the menus). See Table 4–1.
- The **left mouse button** runs commands and selects options in dialog boxes.

There is some overlap in the function of the left and right buttons when there is only one possible action. For example, the left button traverses the menus if it is pressed on an item that is not an executable command. See Table 4–1.

Table 4–1
Mouse Button Summary

Mouse Cursor	Left Button	Right Button
On menu command:	Run command	No effect
On menu item with right arrow:	Enter submenu or dialog box	Enter submenu or dialog box
Outside any menu or dialog box:	Return to application	Return to application
On parent of submenu or dialog box:	Return to parent	Return to parent
In dialog box:	Select or manipulate dialog box items	No effect

Keyboard Operation

The keyboard can also be used to operate the menu interface. Table 4–2 gives the keyboard commands for SUN and for PC operating systems.

Table 4–2
Keyboard Operation Summary

SUN Key	PC Key	Function
Arrow Keys	Arrow Keys	Move the mouse cursor around the screen. Shift-Arrow moves the cursor 10% of the screen.
Esc	Esc	Exits the menu system and returns to the application program or returns to the parent menu.
R9	PgUp	Moves up one level in the menus.
R15	PgDn	Moves down one level in the menus.
On the numeric key pad:		
N/A	+	Mimics the left button click.
N/A	-	Mimics the right button click.

Menus and Dialog Boxes

Menus

Menus and submenus present lists of options to choose from. An arrow at the right of a menu or submenu item indicates there is additional information at a lower level in the menus. Highlight menu items by placing the mouse cursor on them.

The menu system contains the following elements: menus, submenus, dialog boxes, and commands.

Menu References

The following key shows how the different menu elements are referenced in the FutureNet documentation.

Menu Selection

Action Buttons

☐ **Check Boxes**

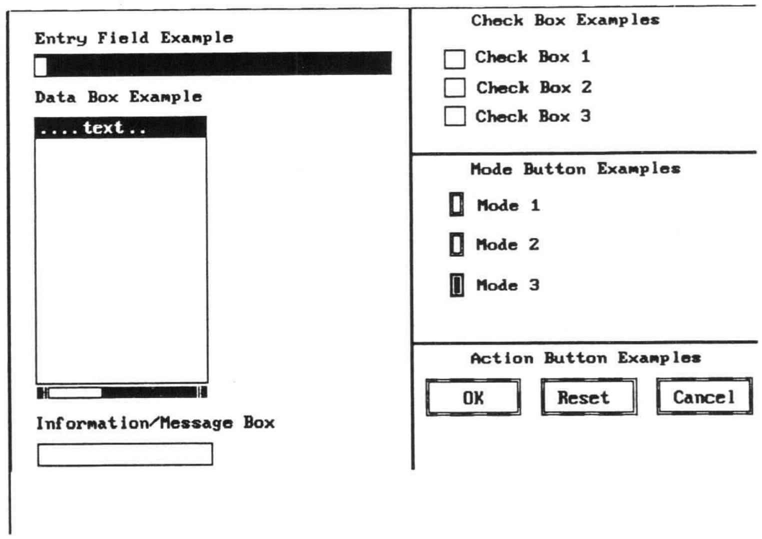
Mode Buttons

Entry Fields

Dialog Boxes

Dialog boxes are used for selecting and running programs. They use the following features to interact with the application program: mode buttons, check boxes, action buttons, entry fields, and data boxes. Figure 4–1 shows an example of a dialog box that contains all the listed features. The Help dialog box, shown in Figure 4–1 later in this chapter shows an example of a dialog box. Other dialog boxes accessed through the EXPORT selection on the main menu are discussed in the *Post User Manual.*

Figure 4–1
Example Dialog Box



Action Buttons

Action buttons run the command specified on the button.

☐ Check Boxes

Check boxes toggle an option on or off. A selection is on (enabled) when it is checked. Toggle a setting by positioning the tag cursor within the box and pressing the left mouse button.

Mode Buttons

Mode buttons select an option from a list of mutually exclusive options.

Entry Fields ▮_

Entry fields are used to enter text.

More than one entry field may be present in a particular dialog box, with the current entry field being the one with the text cursor. Select the desired entry field with the mouse.

When a field is selected, the text cursor moves to the location of the mouse cursor or the end of the entry field data, whichever is farther to the left.

If the text entered is longer than the entry field, the text scrolls to the left. To edit text in the entry field, use Ctrl plus the arrow keys.

The location of the mouse cursor is irrelevant when entering data in an entry field. Text entered at the keyboard is entered in the current entry field at the text cursor.

Data Boxes

Data boxes display lists of data. Dark gray data boxes can be edited. To edit a field, position the tag cursor within the field and press the left mouse button. The editing cursor appears at the cursor location, and the field can be edited the same way as the FutureNet command line.

Light gray data boxes can be scrolled through, sometimes selected from or to, but not changed. Some fields are changed through entry fields, or by entering commands on the command line. Refer to the discussion on specific dialog boxes for information about a particular data box.

Scroll Bars

Scroll bars are used to move the data vertically or horizontally within a data box if all the data does not fit. The scroll bar at the bottom moves the data horizontally; the scroll bar to the right moves it vertically.

The small arrows at the ends of scroll bars move the data one line vertically or one character horizontally in the direction of the arrow. The highlighted portion of the scroll bar can be tagged and dragged using the mouse cursor. Clicking the mouse on an unhighlighted portion of a scroll bar moves the data incrementally through the data box.

Note: Most FutureNet mouse functions require only the click of the appropriate button. Only the scroll bars use the "tag and drag" function.

Editing Functions

Tables 4–3 and 4–4 summarize commands for editing, inserting, or deleting data in entry fields.

Note: *For cursor and editing commands that use both the control key and a letter, hold down the control key and then press the desired letter.*

Table 4–3
Cursor Commands

Key	Function
Ctrl-[←] or Ctrl-[→]	Moves the cursor within the current entry field.
Ctrl-[Home]	Moves the cursor to the beginning of the line.
Ctrl-E or Ctrl-[End]	Moves the cursor to the end of the line.
Ctrl-N	Moves the cursor down through a group of connected entry fields, one field at a time.
Ctrl-P	Moves the cursor up through a group of connected entry fields, one field at a time.

Table 4–4
Editing Commands

Key	Function
Backspace	Deletes the character to the left of the cursor.
Del	Deletes the character under the cursor.
R11 (Sun) Ins (PC)	Toggles the insert or overwrite text mode (for insert on, the cursor is a solid block; for insert off, the cursor is a half block).
Ctrl-D	Deletes from the cursor to the end of the line.
Ctrl-K	Clears the line.
Ctrl-L	Refreshes the current entry field.
Ctrl-T	Marks a block of text in entry fields (see Cut Text function below).
Ctrl-W	Deletes from the cursor to the beginning of the line.
Ctrl-Y	Pastes the last text cut at the cursor (see Cut Text function below).

Cut Text Function

With the Mouse

Place the mouse cursor over the first letter of the text to be cut. Press and hold down the left mouse button while dragging the mouse cursor across the text you want to cut. Marked text is highlighted. Release the left button and press the backspace key once to cut the marked text.

With the Keyboard

Place the cursor under the first letter of the text to be cut and press [Ctrl] - [T]. Using the right or left arrow keys, move the cursor to the end of the block of text and press [Ctrl] - [T] again to mark the block of text. Press the Backspace key once to cut the marked text.

Paste Text Function

Use [Ctrl] - [Y] to paste the last text cut. Text is pasted beginning at the cursor.

The FutureNet Command Line

The first line below the drawing workspace is the command line. All commands, whether selected from the menu with the mouse or typed from the keyboard, appear on the command line. The command line cursor is either a full block, indicating insert mode, or a half block, indicating overtype mode. [Ins] or [R11] toggles alphanumeric insert mode.

This section discusses entering commands on the command line only. Information on entering commands from the menus is given earlier in this chapter. For a complete listing of FutureNet commands, see the Command Reference Card at the end of the *Command Reference*.

Entering Commands

- All FutureNet commands can be typed on the command line and entered by pressing [↵]. The command remains on the command line after it is entered, and may be reentered by pressing [↵] or the middle mouse button. This feature provides a convenient way to do repetitive operations.
- Commands written to the command line as the result of being chosen from the menus are identical to those typed in.
- After a command is entered, the first keystroke of a new command clears the command line and begins a new command.
- More than one command can be entered to form a command macro — see the discussion in the section "Using the FutureNet Command Language," in Chapter 3.

Command Behavior

Commands behave in three general ways:

- Certain commands are run immediately; entering the command is all that is required for the system to act.
- Some commands support optional command parameters. If a parameter is optional, the command and the parameter must be typed on the command line before ↵ is pressed. For instance, the symbol reflection command (.RE) entered by itself performs horizontal reflection; including the **V** parameter performs vertical reflection.
- Other commands require a parameter. The parameter can be entered with the command, or FutureNet prompts for parameter if it is missing.

Command Line Editing

A number of editing functions can be performed on the command line. These functions are summarized in the section "Command and Function Keys" below.

Command and Function Keys

A number of commonly used FutureNet commands and functions have been assigned to various keys and control key combinations. These keys and key combinations enter commands without affecting the command line.

Table 4–5 summarizes the operation of command and function keys for these cursors. A hollow circle indicates the key has no function for that cursor; a solid circle indicates the key is functional.

Table 4–5
Cursor Control and Editing Keys

Key	Description	Design (Cursors)	Command Line (Cursors)	Alpha-numeric (Cursors)
Backspace	Deletes the character to the left of the cursor and moves the cursor left.	○	●	●
Del	Deletes the character at the cursor position and shifts characters on the right of the cursor to the left [1].	○	●	●
Spacebar	Replaces the current character with a space and moves the cursor to the right [2].	○	●	●
← or →	Moves left or right 1 display unit or character.	●	○	●
↑ or ↓	Moves up or down 1 display unit.	●	○	○
Ctrl – ← or →	Moves cursor left or right.	○	●	○
Ctrl – Backspace	Erases the line and moves the cursor to the beginning of the line.	○	●	●
Home or R7	Centers the display around the design cursor; moves the cursor to the beginning of the line.	●	○	●
Ctrl – Home or R7	Moves the cursor to the beginning of the line.	○	●	○
End or R13	Moves the cursor to the end of the line.	○	○	●
Ctrl – End or R13	Moves the cursor to the end of the line.	○	●	○
Ins or R11	Toggles command line and alphanumeric field insert mode.	○	●	●
Tab	Moves the cursor to the next alphanumeric field [3].	○	○	●
Shift – Tab	Moves the cursor back to the previous alphanumeric field [3].	○	○	●
↵	Runs commands on the command line. In alphanumeric mode, moves the cursor down one line for the current font size.	●	○	○
PgUp or PgDn	Moves to beginning or end of symbol instructions.	●	○	○

[1] *This key acts slightly different for alphanumeric cursors in fields with center or right justification. See the 'J command in the* FutureNet Command Reference.

[2] *Assumes overtype mode. In insert mode, characters are inserted at the cursor position. No characters are overtyped.*

[3] *Only true when the cursor is located within a symbol in alphanumeric mode.*

Command Line Options

The following common options have been defined for the command line interface.

Option Summary

Table 4-6 summarizes the command line options.

Table 4–6
Command Line Options

Option	Definition
-i*filename*	Specifies input file.
+i	Resets option to no input files included.
-o*filename*	Specifies output filename.
+o	Resets output filename to default.
-e*filename*	Specifies output file for errors.
+e	Resets error filename to default.
-s	Sets program to silent mode.
+s	Resets program to non-silent mode.
-q	Sets program to query mode.
+q	Resets program to non-query mode.

Specifying Input Files (*filename* or -i*filename*)

Any string on the command line that is not preceded by a -, +, or @ symbol is interpreted to be an input file. The **- i*filename*** option was included for completeness and also as a means of adding clarity to command files, where

-i can be used to identify input files.

+i resets the option so that no input files are included.

Specifying Output Files (-o*filename*)

The **-o*filename*** option is used to specify names and dot extensions (optional) for output files. If the -o option is not used to specify a name for an output file, the output filename will default to the name of the input file with an application-specific dot extension supplied by the application program.

The **+o** option resets the output filename to the default, the name of the first input file.

Specifying Error Files (-e*filename* and +e)

The **-e*filename*** option is used to specify names and dot extensions (optional) for error files. If the -e option is not used to specify a name for the error file, the error filename will default to the name of the first input file with an application-specific dot extension supplied by the application program.

The **+e** option resets the system so that the error filename will default to the name of the first input file with the dot extension supplied by the application program.

Silent Mode (-s and +s)

The **-s** option is used to set the program to silent mode, so that it will run without sending error and status messages to the display device.

The **+s** option resets the program to non-silent mode.

Query Mode (-q and +q)

The **-q** option sets the program to query mode, so that the program will interactively request all options and, where appropriate, their values. Query mode will be entered automatically when the minimum required options for a given application are not specified. Default answers can be selected by pressing ↵. Query mode will continue until the required information has been provided.

The **+q** option resets the program to non-query mode, so that it no longer requests all unspecified or multiple value options, provided the minimum set of options has been specified on the command line.

The System Command Line

The following options are used with the FutureNet Post Processing Tools:

-i	Input file
-e	Error file (default extension is .der)
-s	Silent mode
-q	Query mode

5 *Mouse, Modes and Cursors*

In FutureNet, the functions of the mouse buttons change, depending on the activity you are performing. These activity-specific configurations are known as modes. Some modes have a cursor associated with them that gives a visual clue to the mode, and aids in performing tasks within the mode. This chapter discusses each mode, how the mouse performs in each, and the mode cursors. The modes are summarized below:

Mode	Description
MENU	Menu Mode
FAST	Fast Drawing Mode
LINE	Line Drawing Mode
MOVE	Move Mode (Tag and Drag)
COPY	Copy Mode
ERASE	Erase Mode
AREA	Area Definition Mode
ALPH	Alphanumeric Field Entry Mode
PTR	Point of Effect Tracking Mode
ZOOM	Zoom Mode
SYMD	Symbol Definition Mode

Mouse Button Basics

Note: For DOS Users — If you have a Logitech mouse and the Logitech mouse driver v5.00d, your mouse movements may be extremely sensitive and ballistic. To slow the mouse down, enter the following line in your **autoexec.bat** *file:*

mouse */sxx /b(on | off)*

where **xx** *is the sensitivity between a low of 01 and a high of 09, and /b controls the ballistics.*

Although the mouse buttons perform different functions in each mode, the buttons have common characteristics:

Left — **Action and Pointing** — The basic function of the left mouse button is to initiate and complete functions within a mode. For example, using the left mouse button alone, you can initiate, draw and complete a multi-segment routing line.

This button answers Yes to any (Y/N)? prompts, and is used to select items in menus and dialog boxes.

Middle — **Modification** — The middle button modifies actions taken in the current mode. For example, the middle button reverses routing priority of horizontal and vertical routing segments.

The middle button can be used to cycle through the MOVE, COPY and ERASE modes after a symbol or other object has been tagged, and to enter commands that remain on the command line.

Note: If you are using a two-button mouse, you can simulate pressing the middle button by pressing the left and right buttons simultaneously.

Right — **Cancel** — The right button's main purpose is to cancel in-process actions and return you to the previous mode, and to answer No to any (Y/N)? prompts.

This button also is used to bring up and move down in the menus.

Using the Buttons

Most FutureNet mouse functions require only a single press of the appropriate button (no double clicking).

- Except for the scroll bars, no action uses "tag and drag," where you press and hold a button while moving the mouse.
- No functions use double-clicking (two quick, consecutive presses of a button).

Tagging

To tag an object, position the graphics cursor on the object and either press the left mouse button, or enter an appropriate command on the command line. Most often, an object is tagged with the mouse. A tagged object has the following characteristics:

- Its outline changes. On color systems, it changes color and may blink, if BLINK is on. On monochrome systems, it blinks.
- An alternate cursor appears. Most often this is the tag cursor.
- A "ghost" image of the object is attached to the cursor. This ghost may be moved to a new location.

Tagging is canceled automatically when the action is completed, or by pressing the right mouse button.

What Can Be Tagged

In general, anything within a drawing can be tagged.

- Symbols (to move, copy, erase, rotate, reflect)
- Lines (to move a vertex)
- Areas (to move, copy, erase, rotate, reflect)
- Alphanumeric fields (to move, copy, erase, rotate)

Tagging with the Mouse

When the mouse is being used to select graphic items, the cursor can be in a location where it's not possible to determine which item is being targeted.

For instance, one of the most common actions in FutureNet is that of moving a symbol. To move a symbol using the mouse, the graphics cursor is positioned on the symbol, and then the left mouse button is pressed to tag the symbol and enter MOVE mode. What actually happens when the left mouse button is pressed, however, depends on what other elements are present and what other possible operations could be performed.

When more than one item can be tagged, pressing the mouse button cycles through the items based on the tagging hierarchy.

Tagging Hierarchy

Tagging hierarchy refers to the way FutureNet processes the available options in locations where more than one exists. Once an item is tagged, pressing the left mouse button cycles through the possible selections, with the tagged object or its outline either blinking or changing color. Also, depending upon what is tagged or which operation is possible, the cursor and mode changes.

Assuming a location where all things are possible, each press of the left mouse button cycles through these options:

Object Tagged	Mode	Cursor	Command Equivalent
Line start point	LINE	Line drawing	/L
Line Vertex	MOVE	Vertex	/V
Symbol	MOVE	Tag	.M
Alphanumeric field	MOVE	Tag	'M
Area	MOVE	Tag	[M

You can use the command equivalent to remove the ambiguity, or continue pressing the left mouse button until the desired function is indicated.

Cursors

The position and appearance of the various FutureNet cursors allows you to quickly determine which mode FutureNet is in and which functions are available to you.

Design Cursors

Design cursors are those used to manipulate and control FutureNet design elements. The design cursors are summarized below:

- Graphics
- Point of effect
- Line drawing
- Tag
- Hourglass
- Vertex
- Area
- Stop sign
- Zoom

Controlling Design Cursors

All of these cursors are controlled in the same manner:

With the Mouse

Moving the mouse moves the design cursors around the workspace. On the SUN, when you move a design cursor outside of the FutureNet window it becomes the Sun environment cursor. When the cursor returns to the FutureNet window, it becomes the same FutureNet design cursor it was before it left the window. The mouse moves design cursors in pixel increments unless GRID SNAP is on, in which case it moves in grid units.

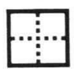
With the Arrow Keys

The four **Arrow** keys can also be used to control the design cursors. (These keys cannot cause a design cursor to leave the FutureNet window. Instead, the FutureNet display pans if AUTOPAN is ON. The **Arrow** keys move design cursors in whole display unit increments.

Combination Command Line and Arrow Keys

Typing a number on the command line and pressing an arrow key moves a design cursor the number of display units indicated in the direction of the arrow. As an alternative, you may also enter the name and value of a direction. For instance, DOWN 3, UP 5, or LEFT 50.

The Graphics Cursor



Description	Small crosshairs surrounded by a box
Cursor Point	The intersection of the crosshairs.
MODE	MENU or FAST
Function	Indicates the starting location for graphics commands. The graphics cursor is the primary FutureNet cursor, from which all other modes and commands are available.

The Point of Effect Cursor

Description	A checkered diamond
Cursor Point	The center of the diamond
Command	'P entered in an alphanumeric field
MODE	PTR (point of effect tracking)
Function	Move the location of a alphanumeric field's point of effect.

The Line Drawing Cursor

Description	Crosshairs which span the workspace
Cursor Point	The intersection of the crosshairs
Command	/L or the left mouse button
MODE	LINE
Function	Draw lines in the style indicated in the LINE status field.

The Tag Cursor

Description	A small arrow pointing up and to the left
Cursor Point	The tip of the arrow.
Command	[M, [C, .M, .C, left mouse button or other command that tags an design element (other than a line).
MODE	MOVE, COPY or ERASE
Function	Move, copy, rotate or reflect a symbol, area or alphanumeric field. If the left mouse button is used to tag the item, you can cycle the COPY and ERASE modes by pressing the middle mouse button. The tag cursor is also used in menus and dialog boxes.

Hourglass Cursor

Description	An hourglass
Function	The hourglass cursor indicates that the system is working and needs you to wait. Once the system work completes and control returns to FutureNet, the previous FutureNet design cursor reappears.

Vertex Cursor

Description	A small box
Function	Move a vertex or line
Command	/V or the left mouse button
Cursor Point	The center of the box
MODE Status Field	MOVE

Area Cursor

Description	Small crosshairs that expand or contract into a rectangle when the mouse or arrow keys are used.
Command	[D

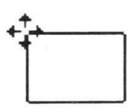
Stop Sign Cursor

Description	A solid circle sitting on a pedestal, resembling a stop sign.

Function

Indicates an unsupported action has been attempted in a dialog box (for example, the PALETTE dialog box).

Zoom Cursor



Description	A rectangle (representing the smallest zoom box) and four-directional pointer. The rectangle expands and contracts with mouse movements.
Function	Define an area to zoom in to.

Note: The Zoom window maintains the aspect ratio of the FutureNet drawing screen, so the specified zoom area fills the display.

Alphanumeric Cursors

Alphanumeric cursors appear in two areas: on the command line and in alphanumeric fields within a drawing. In both areas, two cursors are possible:

- **Half cursor** — This is the default cursor in alphanumeric mode, and indicates that the cursor is in overtype mode.
- **Full cursor** — This is the default cursor from the command line, indicates that the cursor is in insert mode. Insert mode is toggled by pressing the Ins or R11 key.

Controlling Alphanumeric Cursors

For detailed information on moving the alphanumeric cursors, and on editing alphanumeric fields, refer to the "ALPH Mode" section later in this chapter, and to the "Keyboard" section in Chapter 3.

Symbol Definition Cursor

The symbol editor cursor is controlled only when in Symbol Definition Mode. For more information, refer to the "SYMD Mode" section later in this chapter, and to the symbol definition commands in the *FutureNet Command Reference.*

Modes

MENU Mode

MENU mode is FutureNet's default mode. MENU mode allows access to the menus and to all other FutureNet modes. The MENU mode cursors are the graphics cursor (in the drawing workspace), and the tag cursor (in the menus).

Reaching MENU Mode

FutureNet starts in MENU mode. You can return to MENU mode by cancelling all other modes or by entering the FAST OFF command.

The Mouse In MENU Mode

In MENU mode, the functions of the mouse buttons depend upon whether a menus is displayed.

With the drawing displayed:

Left — Draws lines (LINE mode) or tags graphic items (MOVE/COPY/ERASE mode), depending on the location of the graphics cursor. (See discussion above on tagging hierarchy.)

Answers Yes to (Y/N)? prompts.

Middle — Enters the command on the command line.

Right — Displays the top-level menu.

Answers No to (Y/N)? prompts.

With a menu displayed:

Left — Enters the highlighted command (and exits the menus).

Middle — No function.

Right — When the tag cursor is on the menu, the right mouse button selects the highlighted field's submenu (if any). When the tag cursor is clear of the menu, the right mouse button exits the menus.

Commands can be entered directly from the menu. Commands that require additional parameters prompt for necessary information.

FAST Mode

The FAST mode has the most frequently-used drawing functions available with the mouse. All other modes are accessible from FAST mode, and you can enter commands on the command line. The FAST mode cursor is the graphics cursor. The command menu can be accessed in FAST mode by entering the MENU command.

Entering and Exiting FAST Mode

The FAST command toggles the mode between MENU and FAST. You can specifically set the mode by including ON or OFF with the FAST command.

The Mouse in FAST Mode

The mouse in FAST mode differs from MENU mode only in the following ways:

Left No difference from MENU mode.

Middle Add/Remove Interconnect Dot. When the graphics cursor is located on a /1- or /2-type line, the middle button toggles an interconnect dot. (/D or F5)

Right Erase Line Segment. The right button erases the line segment at the graphics cursor location. (/ES or F6)

LINE Mode

LINE mode is used for drawing all types of lines: interconnection lines, bus lines, and graphic lines. The LINE mode cursor is the line drawing cursor.

Entering LINE Mode

With the Mouse

Press the left mouse button in a valid location (in a symbol where a pin point of effect is visible, on a symbol cell boundary, or outside of any drawing element (in empty space)).

From the Keyboard

Enter the /L command in a valid location. The line cursor appears, and the MODE status field displays LINE.

Exiting LINE Mode

Either complete the line segment in process, or press the right mouse button to cancel LINE mode. The graphics cursor reappears and you return to the previous mode.

The Mouse in LINE Mode

Left If one routing segment is on the screen, makes it permanent. If two are visible, makes permanent the first, and allows a new routing segment to be drawn, or exits LINE mode if all routing segments have been made permanent.

Middle Switches the vertex of two routing segments. (/R)

Right Cancels LINE mode. All routing segments disappear. You are returned to either MENU or FAST mode.

MOVE, COPY and ERASE Modes

The MOVE, COPY and ERASE (ERAS) modes are accessed and work similarly. These modes allow you to perform basic editing functions without entering commands on the command line. The tag cursor is used in the MOVE, COPY and ERASE modes.

Entering MOVE, COPY or ERASE Mode

Mouse Press the left button when the cursor is in a symbol, alphanumeric field or area. The MODE status field says MOVE, the tag cursor appears, and the symbol, alphanumeric field, or area is tagged (its boundary changes color and/or blinks).

Keyboard Enter a symbol (.), alphanumeric (') or area ([) MOVE, COPY or ERASE command.

Exiting MOVE, COPY or ERASE Mode

There are two ways to exit these modes:

- Complete the operation by pressing the left mouse button or reentering the MOVE, COPY or ERASE command.
- Cancel the operation by pressing the right mouse button. The graphics cursor reappears and you return to the previous mode.

The Mouse in MOVE, COPY and ERASE Modes

Left Performs the operation (MOVE, COPY or ERASE) shown in the MODE status field. When the mode is either MOVE or COPY, a new location for the item is confirmed. When it is ERASE, the item is erased.

Middle Cycles through MOVE, COPY, and ERASE when an object has been tagged by the left mouse button (has no effect when an object has been tagged by a command initiated from the menu or command line).

Right Cancels MOVE, COPY or ERASE mode and returns to either MENU or FAST mode.

AREA Mode

AREA mode is used to define a specific area within a drawing. In addition to the tag cursor, the area definition, which begins as small crosshairs and expands into a rectangle, is manipulated in AREA mode as the mouse is moved. All symbols, alphanumeric fields and lines completely within the boundary of the definition are affected by AREA commands.

Note: *Connections between symbols made with type /1 and /2 lines are not rubberbanded when an area is rotated or reflected in relation to symbols outside the area definition!*

Entering AREA Mode

Mouse

Enter the [D command from the menu.

Command Line

Enter the [D command.

In either case, the tag cursor appears and:

- If there is no area definition currently on the screen, or if the graphics cursor was located outside an existing definition, area definition begins at the cursor location; or
- If the graphics cursor was located inside a current area definition, the corner nearest the cursor snaps to the cursor point of the tag cursor and you are able to resize the definition.

Exiting AREA Mode

Complete the area definition by entering the [D command again or by pressing the left mouse button. The area definition is confirmed (becomes permanent), the graphics cursor reappears, and FutureNet returns to the previous mode.

Cancel the area definition by pressing the right mouse button. The in-process area definition disppears, the graphics cursor reappears, and FutureNet returns to the previous mode.

The Mouse in AREA Mode

Left	Completes area definiton, returning you to MENU or FAST mode.
Middle	No function.
Right	Cancels area definition and returns you to MENU or FAST mode.

PTR Mode

Point of Effect Tracking (PTR) mode is used to override the default point of effect placement for alphanumeric fields, allowing you to position the point of effect in the location of your choosing. The PTR mode cursor is the point of effect cursor.

Entering PTR Mode

Mouse

Position the cursor in an alphanumeric field and enter the 'P command from the menu, or with the 'P command on the command line, press the middle button.

Keyboard

Enter the 'P command.

In both cases, the point of effect cursor appears and PTR is displayed in the MODE status.

Exiting PTR Mode

Position the point of effect cursor in the new location, then either press the left mouse button or enter the 'P command to confirm. The graphics cursor reappears and FutureNet returns to the previous mode.

Cancel the point of effect move operation by pressing the right mouse button. The point of effect is unmoved, the graphics cursor reappears, and FutureNet returns to the previous mode.

The Mouse in PTR Mode

Left Comfirms the move operation for the point of effect.

Middle Identical to the left mouse button.

Right Cancels the point of effect placement operation. FutureNet returns to either MENU or FAST mode.

ZOOM Mode

The ZOOM mode is activated by entering the ZOOM command. This mode allows exact definition of the portion of a schematic to be displayed on the screen.

Entering ZOOM Mode

Menus

Place the design cursor at the corner of the required zoom area and select the ZOOM command from the menus.

Keyboard

Place the design cursor at the corner of the required zoom area and type the ZOOM command on the command line.

In both cases, the zoom cursor (a small rectangle) appears. The rectangle is expanded or contracted with the mouse, similar to the area definition cursor.

Exiting ZOOM Mode

To complete the ZOOM command, place the cursor at the opposite corner of the desired zoom area and press the left mouse button. The display adjusts to include only those elements defined by the zoom area.

To cancel ZOOM mode without changing the display, press the right mouse button.

Either operation returns FutureNet to MENU or FAST mode.

The Mouse in ZOOM Mode

Left Confirms the zoom area, modifying the display and returns to MENU or FAST mode.

Middle No function.

Right Cancels ZOOM mode and returns to MENU or FAST mode. The display is unchanged.

ALPH Mode

Alphanumeric (ALPH) mode is used to enter alphanumeric text into the drawing. You cannot enter or exit ALPH mode using the mouse, and the only function the mouse has in the mode is that of positioning the alphanumeric cursor within the alphanumeric field.

For information on using ALPH mode, refer to Chapter 3, ''Using FutureNet.''

SYMD Mode

Symbol Definition (SYMD) mode indicates you are in the FutureNet symbol editor. For information on SYMD Mode, refer to .S in the *FutureNet Command Reference*, and to "The Symbol Definition Screen" in Chapter 3.

6 *Introduction to the Tutorial*

The following suggestions will help you get the most out of the tutorial sessions:

- Do the sessions in order. Each session builds on what you learned in the session that precedes it. The drawing you create in one is used by the next. Do not erase any session files until all sessions have been completed.
- Each session is designed to be completed in one sitting. Therefore, try to complete each session without interruption.
- Before beginning these sessions, take time to look at Chapters 1 through 4). These chapters contain important information about FutureNet.
- After finishing the sessions, explore the system on your own. With the aid of the *FutureNet Command Reference* and reference card, you can learn much more about FutureNet, and about how to deal with your own design challenges.

Conventions

The following conventions are used during the tutorials.

1. Paragraphs beginning with a number are steps you should perform, or which contain information crucial to the task you are doing. Read and perform them carefully.
2. Actions you take at the keyboard are generally prefaced with one of three words:
   - ***Type*** means you should type the characters mentioned (do not press [↵]).
   - ***Press*** means you should press a key bearing the name mentioned. Keys are represented by their names in boxes. For instance, [↵] and [Tab] .
   - ***Enter*** means you should type the FutureNet command named and press [↵] . You may either enter the command, if you know it or if it is mentioned, or you may select it from a menu.

3. Note the differences in the appearance of an uppercase alphabetic O and a zero, as shown on the following table.

	Alphabetic	Numeric
Text	O	0
Screen Example	o	0

4. Pay special attention to the sections titled "Before You Begin." They tell you what you need to do first, so the session moves smoothly.

5. Although we recommend that you follow the sessions and produce the drawings yourself, you can use the Data I/O-supplied drawings for Sessions 1 through 5 (Chapters 7 through 11). The following files are supplied with the FutureNet software:

   clkres.ara
   sr2bit.dwg
   session1.dwg
   session2.dwg
   session3.dwg
   session4.dwg
   session5.dwg

6. If you are going to do the structured design sessions (Chapters 12 and 13), you need the following files supplied with the FutureNet software:

   scontrol.dwg
   count8.dwg
   dff1.dwg
   dff2.dwg
   equality.dwg
   equtest.dwg
   slogic.dwg

7. Before starting the tutorials, review the information on using FNLIB in Chapter 3, "Using FutureNet." FutureNet must be able to find the library **system.sym** in order for the sessions to work correctly. Your system administrator may also be able to help you with this.

7 *Session 1: Libraries, Symbols and Areas*

About Session

This session introduces you to a broad range of FutureNet editing and drawing commands that deal with manipulating symbols.

In addition, you begin to build the schematic used throughout the sessions. The schematic represents a simple circuit composed of a CPU, clock, various buses and other parts. A schematic reflecting all the steps in a session appears at the session. Refer to those illustrations when necessary.

These topics are discussed:

- Symbol commands that let you load, move, copy, erase, rotate and reflect symbols
- Area commands that let you load, move, copy, erase, rotate, reflect and save areas
- Drawing commands that let you save, load, erase and clear drawings
- Quitting FutureNet

Step 1. Before You Begin

1. Review at least these portions in Chapters 2 and 3:
   - Using the Mouse
   - Entering Commands
   - Using the Menus
   - Starting FutureNet
2. Start FutureNet (enter **fn**).

Step 2. About Libraries

All symbols used by FutureNet are stored in libraries which may have the filename extension **.sym**. A FutureNet library may be opened in one of two modes: reference or update.

Opened as reference, libraries are used exclusively as a source of symbols loaded into your drawing.

Opened as update, libraries may be modified. The symbols within them may be changed or deleted. You may also add new symbols you create.

Any library may be loaded in either mode, with the exception of the default system library **system.sym** which may only be used as a reference library. Be aware that **system.sym** is always loaded as the default reference library (if it is found on startup), and that you may have only one update library open at a time.

The Library Commands

Library commands can be used to open and close update and reference libraries. Library commands can also be used to display the contents of a library.

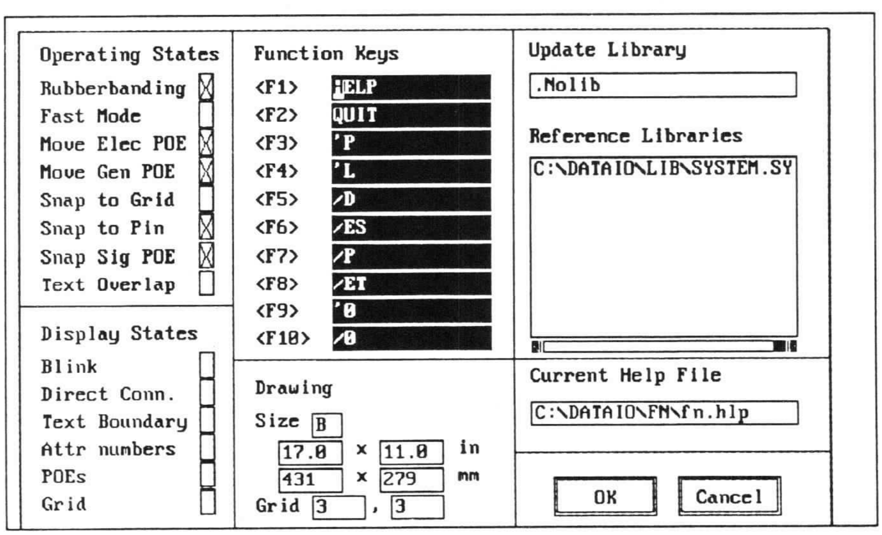
.LIB	Opens the update library
.NOLIB	Closes the update library
LIB	Opens a reference library
NOLIB	Cancels all reference libraries except SYSTEM.SYM
.DIR	Displays Symbol Library Directory

Create/Open an Update Library

Using the .LIB command, you can create and open an update library.

1. Enter the **PROFILE** command. The PROFILE screen appears. The upper right-hand corner (see Figure 7–1) shows which libraries are currently in use. Because no update library has been specified the UPDATE field reads **.Nolib**. Exit the PROFILE screen.

Figure 7-1
The PROFILE Screen



2. Create an update library (this automatically opens it, also). Enter the .LIB command. You are prompted:

   ```
   Enter symbol library filename
   ```

   The name you enter can be any legal 8-character filename. The system automatically adds the filename extension **.sym** for you.

3. Enter something easy to remember. It can even be your own name. You are prompted:

   ```
   Library doesn't exist or is in use. Create new one?
   (Y/N)?
   ```

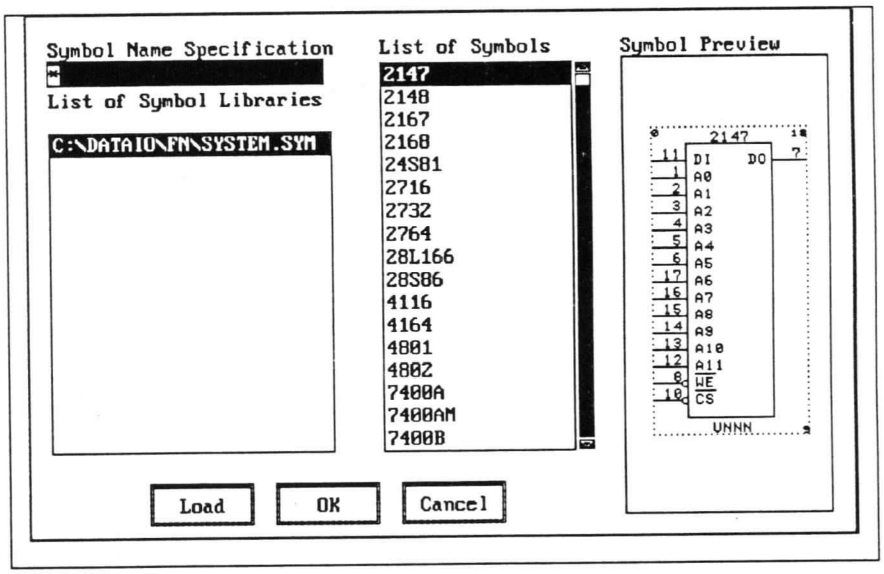
4. When you answer yes, the library is created. You can now check the PROFILE screen to see that the library has been opened.

List Library Contents

Using the .DIR command, it's possible to look at what is inside the libraries.

5. Enter the .DIR command. Your screen will look similar to Figure 7–2.

Figure 7-2
The DIR Screen



The Symbol Name Specification selects which symbols are displayed in the List of Symbols window. The List of Symbols window displays the contents of the selected library. If necessary, use the scroll bar to view the portion of the library you are interested in. To select another library, click on the library name.

On Your Own

Feel free to experiment with library commands. Try opening and closing various libraries, noting the following:

- You can open or close a library only if the drawing area is clear (SAVE your drawing, if necessary, then use the CLEAR command).
- All update or reference libraries, other than the default reference library (**system.sym**), must be opened each time you start FutureNet unless automatically opened using a command file. For information on command files, see Chapter 3, "Using FutureNet."

- The environmental variable FNLIB provides library path information. Refer to Chapter 2, "Installation Options," in the *FutureNet Installation Guide* for information on using the FNLIB environment variable.

Step 3. About Symbol Commands

Now that you know where symbols come from, you can begin working with them. The symbol commands allow you to manipulate symbols in a number of ways. Referring to the list below, take a look at some of the most useful symbol commands. All symbol commands, with the exception of the * command, begin with a period (.).

.L	Load a symbol
*	Load a symbol already tagged for move
.M	Tag a symbol for moving
.C	Tag a symbol for copying
.E	Erase a symbol
.R	Rotate a symbol
.RE	Reflect a symbol
.K	Cancel symbol tag

Step 4. Symbol: Loading

1. Position the graphic cursor anywhere on the screen and enter the .L command by either entering it on the command line or by choosing it from the menu.

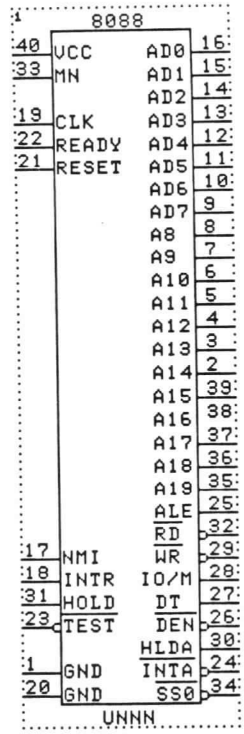
   The .L command appears on the command line, and you are prompted for the name of the symbol to load:

   ```
   Enter symbol name
   ```

2. Enter on the command line the name of one of the symbols that you saw previously, for example, enter:

   8088

   The symbol appears with its upper left corner on the cursor. The dotted line is called the symbol cell boundary. Everything within this line, such as pins, alphanumeric fields, etc., is part of the symbol. The solid line is the actual symbol cell. Everything within the symbol cell boundary prints, but the symbol cell boundary itself does not.



3. Try loading a symbol on top of another symbol by positioning the cursor inside the symbol you just loaded. You will see the message:

   ```
   Symbol boundary conflict
   ```

   This message means that, at the very least, two dotted symbol cell boundaries are overlapping. Move the cursor into another location and try again.

Note: *You may also load a symbol by entering the .L command and the symbol name on the command line at the same time.*

Step 5. Symbol: Move, Copy, Erase

As you can tell from the symbol commands, once a symbol is on the drawing, it may be moved, copied, rotated, reflected and erased.

Method #1—By Command

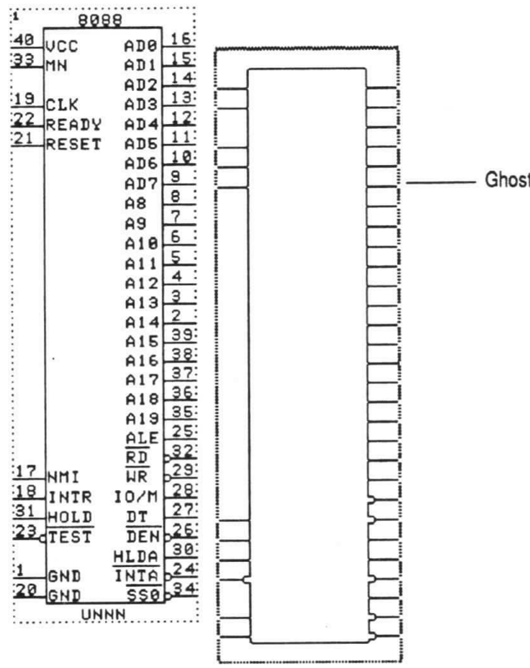
You can MOVE, COPY and ERASE a symbol by using the mouse or by entering commands on the command line. The command line is shown first.

1. First, try a move operation. Place the cursor anywhere inside the dotted-line boundary of a symbol.

2. Enter the .M command.

   The symbol's boundary blinks (on monochrome systems) or changes color (on color systems), indicating it is tagged. Note that MOVE appears in the MODE status field.

3. Begin to move the cursor to the new location. Notice the "ghost" attached to it.



4. Press the left mouse button to complete the move operation. The symbol snaps from its original position to the new cursor location.

Note: *The * (asterisk) command is a simple combination of the .L and .M commands; the symbol loaded is already tagged for an operation. Try it!*

5. The .C (copy) command works in the same fashion. The .E (erase) command erases the symbol. Experiment with these commands.

Method #2—By Mouse

You can also MOVE, COPY and ERASE a symbol by using only the mouse.

6. Choose a symbol. Position the graphic cursor inside of it, in a location where the cursor touches nothing else. Press the left mouse button. The dotted symbol cell outline either changes color (on color systems) or begins to blink (on monochrome systems). The symbol is tagged.

   The MODE field says MOVE.

Note: *If you positioned the cursor in the wrong location, the "line cursor"—a large crosshairs—might appear. If this happens, press the left button again to actually tag the symbol. (More on the line cursor later.)*

7. Press the middle mouse button a few times, slowly. Notice that the MODE field cycles through MOVE, COPY and ERAS (erase) operations. Cycle through to COPY.

Note: *If you are using a two-button mouse, you can simulate pressing the middle button by pressing the left and right buttons simultaneously.*

8. Move the mouse to the right. Notice that the ghost detaches from your symbol. Move the ghost clear of the symbol.

   Press the left button to confirm the COPY operation. The MODE field returns to MENU and the copied symbol appears at the cursor position, replacing the ghost.

Note: If you cycle to ERAS and press the left mouse button, the symbol is erased.

On Your Own

Experiment with these commands, using both the command line and the mouse. It is important that you understand both methods because each will be used in different ways during your work. Most often you'll use the mouse to MOVE, COPY and ERAS. Later, you'll be introduced to command lists which can be used to create macros. For those, you'll need to understand how the commands work.

Step 6. Load Session Symbols

Using the commands you have learned so far, do the following:

1. Delete all symbols from your drawing.
2. Load the following symbols. They are the symbols necessary to complete the session.
   - Refer to the illustration at the end of this chapter, if necessary.
   - To get to a coordinate location, such as 57,96, enter CURSOR 57,96 on the command line to move the cursor to that location.
   - The symbols only need to be close to the locations mentioned, not exactly positioned.
   - All of these symbols are loaded by entering the symbol name as it appears in the left-hand column. Use either the .L or * command.

8088	At 57,96
8755	At 133,96
8185	AT 134,33
GND	Two GND symbols are needed: one at the lower left corner of the 8088 and one at the lower left corner of the 8755
RES	At 16,88

Step 7. Symbol: Reflect and Rotate

This step introduces two more symbol commands, .RE and .R. Once a symbol has been reflected and rotated it is often difficult to see the change. The SYMB status field indicates if the symbol the cursor is on has been rotated or reflected. Each time you perform a step in this section, notice the change to the SYMB status field.

Reflection

Symbols may be reflected vertically or horizontally. The default setting is horizontal reflection. In horizontal reflection all symbol elements move horizontally around a vertical axis. In vertical reflection, just the opposite happens: all symbol elements move vertically around a horizontal axis.

Horizontal Reflection

To reflect a symbol horizontally, use the .RE command.

1. Position the cursor on a symbol, and enter the .RE command. The symbol's ghost is tagged and reflected, and the MODE status field says MOVE. You could move the symbol now, in addition to reflecting it.
2. Press the left button to confirm the reflection. The symbol itself flips around an imaginary vertical axis through the center of the symbol and the SYMB field is updated to say REFL.

   At first glance it is difficult to see what happened because the text remains oriented face out so it can be read. But if you look carefully, you'll see that the symbol graphics have actually reflected, as if we were looking at the symbol from behind.
3. Experiment with this symbol, flipping it back and forth, observing the changes to its appearance.

Vertical Reflection

To reflect a symbol vertically, you must add a **v** to the .RE command.

4. Position the cursor on the symbol and enter the command:

   .RE V

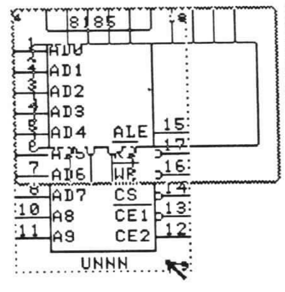
   The symbol flips vertically around a horizontal axis (notice that the part number and symbol reference designator appear to have exchanged positions). Even though you've turned the symbol "upside-down," the text remains oriented right-side up.

   The SYMB field says R 180/REFL, indicating that reflection has taken place, and that the effect of the reflection was also to rotate the symbol 180 degrees.

Rotation

You can rotate symbols clockwise or counterclockwise in 90-degree increments around their geographical center. The default is 90 degrees clockwise. Counterclockwise rotation is indicated by using a minus (–) sign.

5. Position the cursor on a symbol (it can be one of the ones you have reflected, if you prefer). Enter the .R command.



6. Press the left mouse button to confirm the rotation.

   As with reflection, at first it is difficult to pick out exactly what has happened, because the text always adjusts to read left-to-right or down-to-up, even though the symbol graphics have turned upside-down.

7. Put the symbol back in its non-rotated position by entering the command:

   .R –

8. You can also specify a specific amount to rotate, in 90-degree increments, for example,

   .R 270

   Experiment by rotating, in a number of ways, one of your symbols.

Step 8. About Area Commands

The area commands allow you to do everything you have done with symbols, but with groups of symbols. You decide the physical dimensions and contents of the area. As long as you can completely include symbols within the area definition, you can perform the same type of operations on an area as on symbols.

In addition, an area can be saved and used to build a library of standard designs.

All area commands begin with a left square bracket ([ ).

[D	Define an area
[C	Copy an area
[M	Move an area
[ERASE	Erase area
[R	Rotate an area
[RE	Reflect an area
[K	Cancel area definition or area tag
[SAVE	Save Area
[LOAD	Load an area

Defining an Area

Before an area command can be used, an area must be defined. Follow these steps to define an area:

1. Choose two symbols. Imagine them enclosed within a rectangle. Position the cursor in the upper left-hand corner of the imaginary rectangle. Enter the [D command. A small crosshairs appears where the graphic cursor was.
2. Move the mouse a very short distance down and to the right. The crosshairs becomes a rectangle which expands as you move the mouse. This is the area definition. Expand it to include the symbols you want.
3. When the area definition is where you want it, press the left mouse button to confirm.

Resizing an Area

Once an area is defined, you can resize it.

4. Position the cursor near a corner and enter the [D command (or press the middle mouse button if the command is still on the command line). The tag cursor appears. You can now enlarge or shrink the definition by dragging the corner attached to the tag cursor to a new location. Confirm the new dimensions by pressing the left mouse button.

Tagging an Area

You may now tag the area just as if it were a symbol, anytime the cursor is inside it. Be aware, however, that there is an order involved in tagging.

5. Position the cursor in empty space within the area definition boundary and press the left mouse button: the line drawing cursor appears. Press the left mouse button again to tag the area. (Untag the area by pressing the right button.)
6. Now position the cursor within a symbol within the area. Press the left button to tag the symbol. Press the left button again to tag the area.
7. Press the right mouse button to untag the area.

On Your Own

The [K command is used to remove an area definition. This command is not used often because you are only allowed to have one area per drawing—each time you define a new area, the former one is cancelled automatically. Use [K to remove an area definition when the definition is no longer needed if it's in the way. The cursor does not have to be within the area for the [K command to operate.

Step 9. Using Area Commands

Once an area has been tagged, you can use the mouse shortcuts you learned earlier for MOVE, COPY and ERASE operations. All other commands must be enterd from the command line.

Manipulating an Area

As with individual symbols, you can MOVE and COPY an area from one location to another.

1. Define and tag an area.
2. Using the mouse shortcuts, either MOVE or COPY it to a new location. As with symbols, once the operation is complete, the area tag is canceled; but, notice that the area definition remains.

Note: Take some time to use the command versions of MOVE ([M), COPY ([C) and ERASE ([ERASE).

3. Define another area. Note that you can only have one area defined per drawing. Enter a command, such as rotation or reflection, on this area.

Saving and Loading an Area

One of the most practical uses of an area is that any area can be saved under a unique name. This means that symbol groups that are used in a number of different drawings can be standardized, and then called in to any drawing.

4. Define an area.
5. Enter the [SAVE command. You are prompted:

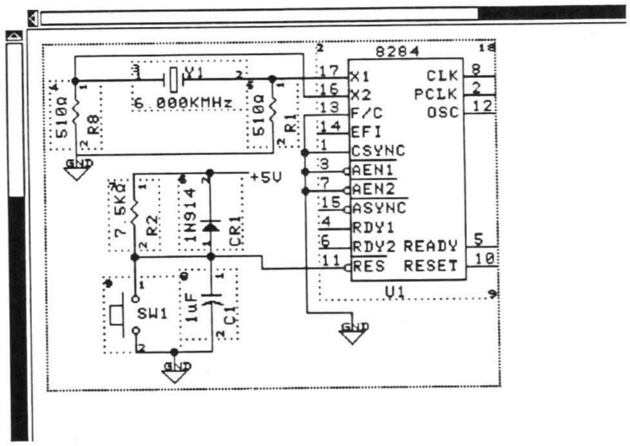
   ```
   Enter filename
   ```

   Enter any legal 8-character filename. The extension .ARA is automatically added by the system.

Note: You do not need to tag an area in order to save it—it only needs to be defined. Additionally, because you may only have one area defined at a time, the cursor does not need to be inside the area when the [SAVE command is enterd.

You can now load this area back into your drawing.

6. Position the cursor in a blank area of the screen. The location must be big enough to contain the area you just defined. Be aware that the "origin" (the point attached to the cursor) of an area is the upper left-hand corner.
7. Enter the [LOAD command with an 8-character filename (you do not need to specify the **.ara** extension). The area is loaded into your drawing.
8. The area **clkres** is needed for the session drawing. Load it into the drawing in the upper left-hand corner (CURSOR 0,0).



On Your Own

Take some time to experiment with the area commands until you feel comfortable.

- Check for your area in your sessions directory. (You can get a listing of files by entering the **DIR** command, or selecting **DIR** from the menus. Do not confuse it with the **.DIR** command, which is used to select symbols.)
- Keep track of the areas you create and delete the ones you do not need once you have finished the sessions.

Step 10. Cursor Movement

This step introduces the cursor commands and functions that help you move around a drawing quickly.

Move Cursor to Absolute Coordinates

Look at the XY status field in the status bar. The numbers indicate the XY coordinates of the cursor in the drawing. You can move the cursor to specific XY coordinates using the CURSOR command.

1. Enter the following on the command line:

   cursor 50,100

   As you can see by the XY status field, the cursor location is now at 50,100.

Move Cursor to Symbol Reference

Each symbol you load into your drawing has a small number in the upper left-hand corner. This is called the symbol reference number, and refers to the order in which symbols were loaded. This number is significant only to FutureNet—it is distinct from the reference designator which is significant to other programs, such as printed circuit board layout applications.

2. Look at the symbols in your drawing. Decide which one you would like to send the cursor to. In this example we'll use 7. Enter:

   n 7

   The cursor moves to the seventh symbol loaded.

Fast Cursor Movement

Normally, the cursor moves in 1-display unit increments. You can temporarily change this unit to anything you wish. Doing so makes it easy for you to move around your drawing using only the cursor keys, and to move small increments accurately.

3. Type on the command line (do not press ⏎):

   20

4. Press any cursor key. The cursor moves in the direction of the key you press, and in the number of units you typed.

On Your Own

Explore these cursor movement commands until you feel comfortable with them. Learning them can make moving around a drawing very easy.

Step 11. Drawing Management Commands

The following commands relate to managing the drawings you create. They are:

LOAD	Load a drawing
SAVE	Save a drawing
FILE	Display the current filename
CONTEXT	Restore editing context from last SAVE
CLEAR	Erase the drawing screen and reinitialize FutureNet
ERASE	Erase drawing without reinitializing filename.

Using SAVE

The SAVE command is used to save a drawing to disk.

1. Enter the SAVE command. SAVE appears on the command line, and the system displays:

   ```
   Enter file name
   ```

2. Enter the filename of your choice on the command line. If you do not include an extension, **.dwg** is automatically added.

   The drawing on your screen is saved to the filename you give.

Notes: If you are using the command line, include the filename the first time you enter the SAVE command. Once a drawing has a filename, you can enter the SAVE command without a filename to write over the older version. FutureNet verifies the overwrite.

Using ERASE, FILE and CLEAR

Now that you have saved your drawing, you can clear the screen using either ERASE or CLEAR, depending upon what you want to happen.

3. Enter the ERASE command. The drawing screen clears, but you are still editing the same drawing.

If you were to SAVE now, the contents of the drawing screen—nothing—would be saved to the file named here.

4. Enter the FILE command. This command shows you which file you are editing. The **Current File** is the one you named above.
5. To begin a new drawing, enter the CLEAR command. If you enter the FILE command again, FutureNet will display

```
No specified file name
```

indicating that you are working on a new drawing.

Using LOAD

The LOAD command is used to bring drawings to the screen for editing.

6. Load the drawing you just cleared by entering the LOAD command (you may include the filename if you are entering from the command line). The drawing appears.

Using CONTEXT

A drawing just loaded displays at the greatest zoom level, with the zoom window in the upper left-hand corner. The CONTEXT command returns you to the zoom level and the location as it was when the drawing was last saved.

7. Enter the CONTEXT command.

Step 12. Clean Up the Drawing and End the Session

Only one major task remains in this session—exiting FutureNet. When you exit FutureNet, the program checks to see if all changes have been saved. If you have unsaved changes, the program asks if you want to discard the changes, or cancel the command.

To see how this works, do the following:

1. Referring to the illustration at the end of this session, remove all unnecessary symbols. If you don't have any extra symbols, load any symbol and remove it. To make this step work, the drawing must have been modified.
2. Enter the Q command. You are prompted:

   ```
   OK to discard changes that have not already been
   saved? (Y/N)?
   ```

3. Answer N or press the right mouse button (you want to save these changes!). Answering no to the message, however, does not automatically SAVE the changes to the drawing.
4. Enter the SAVE command.
5. Now quit FutureNet (Q). You are returned to the operating system.

Looking Ahead

In the next session, you'll learn about creating symbols similar to the ones you've been working with here. In addition, these topics are covered:

- Attributes and points of effect
- Alphanumeric text entries
- Command lists

Summary

This is a summary of the commands presented in session 1.

Library Commands

Command	Function
.LIB	Open the update library
.NOLIB	Close the update library
LIB	Open a reference library
NOLIB	Cancel all reference libraries except SYSTEM.SYM
.DIR	Display Symbol Library Directory

Symbol Commands

.L	Load a symbol from a library
*	Load a symbol already tagged for move
.M	Move a symbol
.C	Copy a symbol
.E	Erase a symbol
.R	Rotate a symbol
.RE	Reflect a symbol
.K	Cancel symbol tag

Area Commands

[D	Define an area
[C	Copy an area
[M	Move an area
[ERASE	Erase an area
[R	Rotate an area

[RE	Reflect an area
[K	Cancel area definition or area tag
[SAVE	Save an area
[LOAD	Load an area

Cursor Movement Commands

CURSOR *x,y*	Move cursor to absolute coordinates given
N *reference*	Move cursor to symbol reference
units Arrow-Key	Fast cursor movement the number of units specified and in the direction indicated

Drawing Commands

LOAD	Load a drawing
SAVE	Save a drawing
FILE	Display the current filename
CONTEXT	Restore editing context from last SAVE
CLEAR	Erase the drawing screen and reinitialize FutureNet
ERASE	Erase the drawing screen without resetting the drawing filename.
Q	Quit FutureNet and return to the operating system

The Mouse

Left Button

In MENU mode, tags symbols or areas when the cursor is in a valid location. Once a symbol or area is tagged, you enter MOVE/COPY/ERASE mode. Once inside of MOVE/COPY/ERASE mode, pressing the left button enters the function shown in the MODE status field.

Middle button

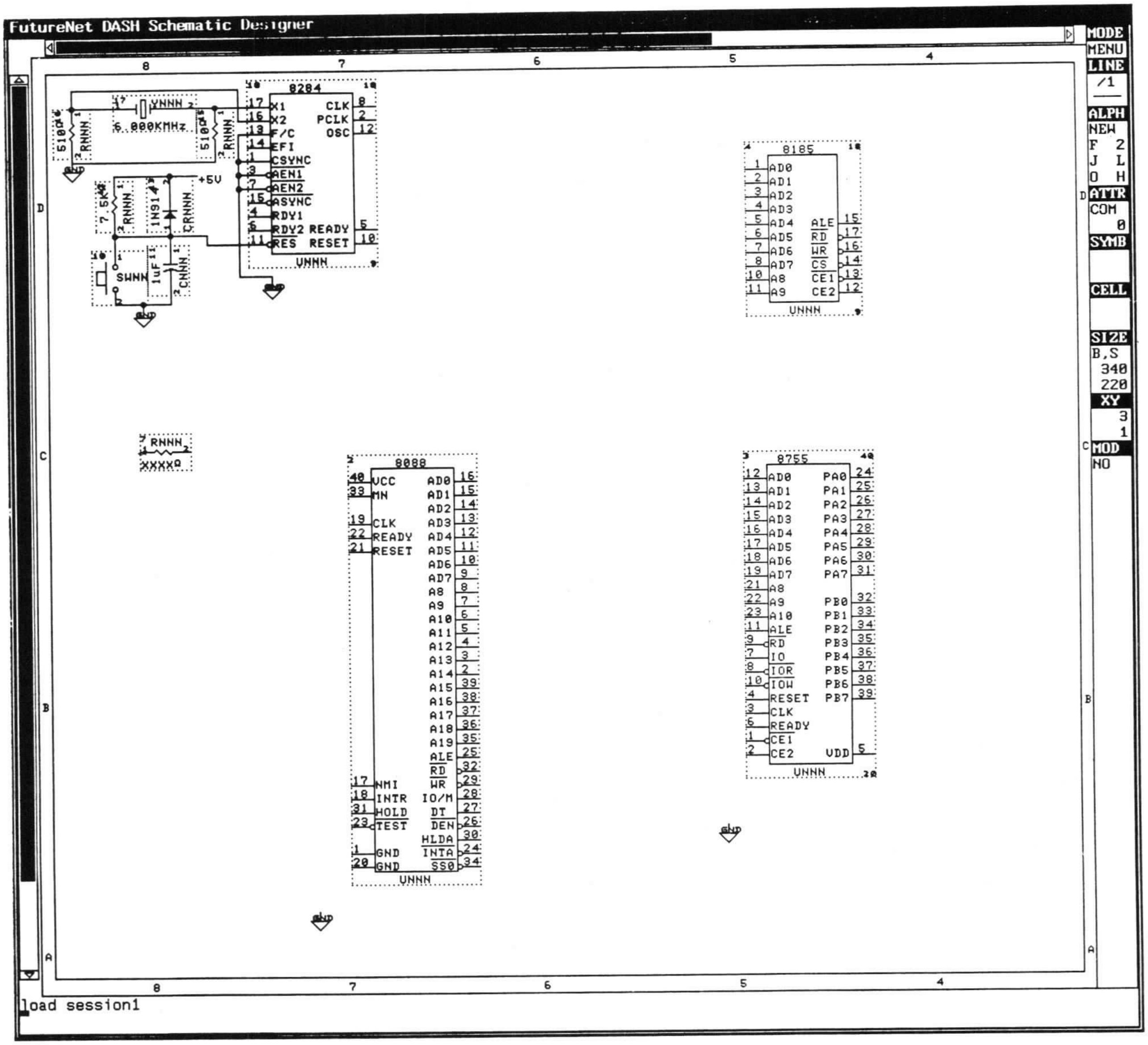
In MENU mode, reenters the command on the command line. In MOVE/COPY/ERASE mode, cycles through those functions.

If you are using a two-button mouse, you can simulate the middle button by pressing the left and right buttons simultaneously.

Right Button

In MENU mode, calls up the menus. In all other modes, cancels any function in progress and all in-progress actions, such as moving or copying.

session1.dwg



8088	At 57,96
8755	At 133,96
CLKRES.ARA	In extreme upper left
corner	
8185	At 134,33
GNDs	One at the lower left corner of
the 8088 and	
	one at the lower left corner of the 8755
RES	At 16,88

8 Session 2: Building Symbols, Alphanumerics

In this session you will build an 8185 symbol. This symbol is already in the library, but symbol building illustrates a number of FutureNet features which are crucial to producing accurate schematics. These FutureNet functions are introduced:

- Creating a block symbol
- Adding pins to a block symbol
- Using alphanumeric mode and setting field characteristics
- Setting and displaying attributes
- Using a command list (macro) to enter block symbol data
- The ATTR, CELL and ALPH status fields

About Symbol Editing

You can create symbols in FutureNet in two ways: using symbol definition mode, which is discussed in Sessions 8 and 9, or by using the .B command which is presented here. The same symbol built by either method would be identical, but symbol definition mode is generally reserved for creating intricate graphic shapes, such as gates.

The steps illustrated here can be used to build any symbol consisting of a square or rectangular symbol with pins—which describes almost all ICs.

Step 1. Before You Begin

1. Start FutureNet
2. Do one of the following:

If you do not want to use your own drawings, or if you have not completed Session 1:

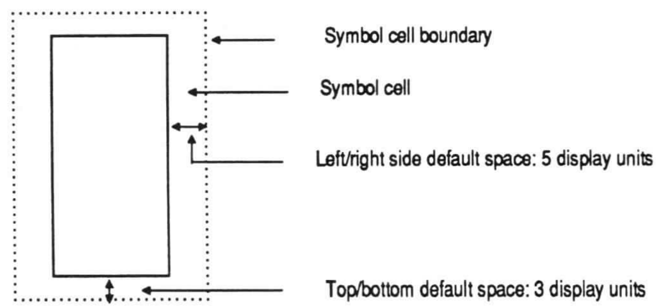
- LOAD the Data I/O-supplied drawing **session2**.

If you are beginning the sessions again and have completed Session 1:

- LOAD the drawing you created in the last session.
- SAVE the drawing you just loaded as **gs2**. This creates a copy of the drawing you created in the last session.
- LOAD the drawing **gs2**. This loads the copy of your drawing into your machine. Thus, any changes you make to the drawing will be made to the copy and not to the original.

Step 2. Build the Symbol Cell

The first step builds the symbol cell. The two commands that build symbol cells are .B and .F. The .F command builds a functional block symbol which is discussed in later chapters. The .B command builds a standard symbol, and is the command we'll use.



The illustration above shows the two components of a symbol. The solid line is the actual symbol cell, which prints. The larger, dotted line is called the symbol cell boundary and does not print. Its purpose is to define the boundary of the symbol, so that no two can overlap.

Into the area between the symbol cell and the boundary are placed the pin stubs, pin numbers, and part and reference numbers, as well as other pieces of information which will be covered later.

1. Position the cursor to the right of the 8185 in your drawing. It will be used as a reference for the symbol you build.
2. Enter the .B command. You are prompted:

   ```
   Enter block symbol width, height
   ```

3. Enter the dimensions of the symbol cell in display units:

   15,32

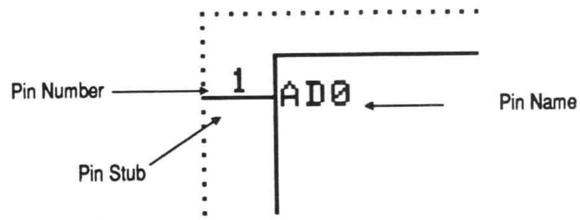
   The block symbol appears in the drawing. Notice the following:

   - The symbol loads in with the top left corner on the cursor.
   - The CELL field is updated and shows the dimensions of the symbol cell boundary (dotted line), 25,38.
   - The actual symbol, shown as a solid-line rectangle, is 15 units wide and 32 units high, as you specified in the .B command. The dotted symbol cell boundary is 3 display units larger on the top and bottom of the symbol and 5 display units larger on the left and right. These are the default values which you can change if you wish. For more information, refer to .B in the "Command Reference."

   This symbol can be tagged, moved, copied, erased, and named and saved just like any other symbol.

Step 3. About Pins

The next step in building the symbol is to add the pins. Before you do this, however, it is important to understand just what makes up a FutureNet pin. Pins are composed of three components: the pin number, the pin stub and the pin name.



The Pin Number

The pin number, in addition to its significance as the pin number, carries electrical information about the pin to post processing packages in the form of an attribute. Attributes are attached to alphanumeric text and are required elements in post processing a FutureNet drawing.

Later sections of this chapter explain more about attributes and how to use them.

The Pin Stub

The pin stub is the graphic element that looks like a pin in the space between the symbol cell and the boundary. Pin stubs come in all standard variations: regular, inverted and clock. In addition, IEC/ANSI pins are available.

Pin stubs are not electrically significant—they are graphic elements which serve as a location for the pin number which carries the attribute. Although you wouldn't want to, you could build an accurate schematic without pin stubs.

The Pin Name

The pin name is a short bit of alphanumeric text which helps you to determine the function of the pin. As with pin stubs, pin names carry no electrical significance. Without them, you might have a difficult time understanding the functions in a schematic, but if other elements were present, the drawing could be successfully processed.

Pin names generally appear inside the symbol cell.

Step 4. Working with Pin Stubs

The Pin Stub Commands

The pin stub commands are:

Command	Description
.-	Add pin stub
.-O	Add inverted pin stub
.>	Add clock pin stub
.>O	Add inverted clock pin stub
.-AI	Add IEC/ANSI pin stub with arrow pointing into the symbol
.-AO	Add IEC/ANSI pin stub with arrow pointing out of the symbol
.>A	Add IEC/ANSI pin stub with arrow pointing into clock input
.<A	Add IEC/ANSI pin stub with arrow pointing out of clock input
.D	Delete pin stub

When you add a new pin stub on top of an existing pin stub, the old is automatically replaced by the new. Use .D to delete a pin stub.

Placing Pin Stubs

Generally, pin stubs are placed anywhere on either side of the symbol, a 3-display unit margin is usually left clear on the top and bottom edges of the symbol cell.

Adding Pin Stubs

When you add a pin stub, it automatically spans the space from the symbol cell to the boundary. Thus, the cursor needs only to be located at the correct vertical location.

1. Position the cursor at the top left corner of the solid symbol cell.
2. Press [↓] three times to move down 3 display units.
3. Enter the .- command. The pin appears, drawn from the symbol cell to the boundary.
4. Move down another 3 display units. Using the .-O command, add an inverted pin stub.
5. Try adding different types of pin stubs over these pin stubs. Note that each new pin stub replaces the old pin stub.

On Your Own

Experiment with all the pin stub commands. When you are finished, delete all of them using the .D command.

Step 5. Using a Command List to Insert Pins

Refer to the symbol 8185 just to the left of the 8185 you are building. All of the pins on the left side of the symbol are the same type. You could add all of them by moving the cursor down 3 display units and pressing either ↵ or middle button after you add the first. There is a simpler way to add the data: use a command list ("macro").

A command list is made up of two or more commands, each separated by a semicolon, typed on the command line. Command lists allow you to enter cursor movement and data in one operation. You can enter alphanumeric data by typing an insert command on the command line and specifying what goes in the field.

Adding information piece by piece is easiest when working with a single field, or when working with data where every entry is very different. Command lists come in handy when you have essentially repetitive data to add.

1. From the top left edge of the solid symbol cell, cursor down 3 display units to the location of the first pin.
2. On the command line, type the following:

   .– ; DOWN 3

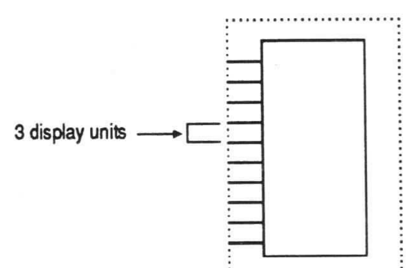
   What you have entered is:

.–	The ADD/REPLACE PIN command
;	Command separator
DOWN 3	The DOWN command indicating 3 display units of movement

---

Note: Commands in a command list can appear in either upper- or lower-case. Insert a space between the command and the numeric value.

3. Press ↵ or the middle mouse button. A pin is drawn, the cursor moves down 3 display units, and the command list remains on the command line.



4. Repeat Step 3 until this side of the symbol looks like the left side of the reference 8185 (9 more pins). When complete, your symbol looks like the symbol shown above.
5. Use .D to delete extra pin stubs, if any.

Step 6. About Alphanumeric Text

At this point in the symbol, it's appropriate to label the pin stubs along the left side of the symbol. Before we do that, take a look at the alphanumeric commands.

Alphanumeric Text Editing Commands

	[Esc]	Toggle alphanumeric mode
Text Appearance	'J *justification*	Set new field (default) justification
	'CH J *justification*	Change existing field justification
	'O *orientation*	Set new field (default) orientation
	'CH O *orientation*	Change existing field orientation
	'0 *through* '7	Set new field (default) font size
	'CH 0 *through* 7	Change existing field font size
	[Ctrl] – [O] 'OVER	Toggle inversion bar
	[Ctrl] – [U] 'UNDER	Toggle underscoring
Field Manipulations	'M	Move alphanumeric field
	'C	Copy alphanumeric field
	'E	Erase alphanumeric field
	'I	Insert string and increment numbering (repetitive operations)
	'F	Find alphanumeric string
	'R	Replace alphanumeric string
	'B	Toggle alphanumeric field boundary display

All alphanumeric commands begin with an apostrophe ('). As you can see, in addition to selecting font size, justification and orientation of text, you can also tag, move and copy alphanumeric text, just like symbols. There are other alphanumeric field commands which do not form part of this discussion. Refer to the reference card at the end of the *Command Reference* for more information.

Controlling the Cursor

Controlling the cursor in alphanumeric mode is identical to controlling it on the command line, with the following exceptions:

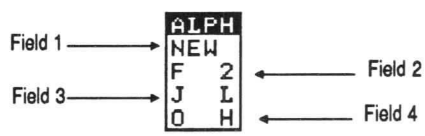
- To move left and right nondestructively, just use the [←] and [→] keys. (You do not need to use them in combination with the [Ctrl] key.)
- Within a symbol, press [Tab] to move sequentially left-to-right, top-to-bottom through text fields, press [Shift] – [Tab] to reverse the direction, and [↵] to move straight down through fields.

The ALPH Status Field

You have several options available when entering alphanumeric data:

- Font–The size of the text.
- Justification–Whether the text lines up left-, right- or center justified.
- Orientation–Whether the text appears vertically or horizontally.
- Printing–Whether the text prints with the drawing.

The ALPH status fields indicate to you all of these things.



Field 1 **OLD, NEW or ONP**–When the cursor is outside of alphanumeric data, the field says NEW, and the parameters listed govern any new text added. When the cursor is on alphanumeric data, the field says OLD or ONP and gives the parameters of the current text. ONP means "old/non-printing."

Field 2 **F (Font)**–You have 7 fonts available to you. Most symbol text is in font 2. The command '*font#* sets the default font. The command 'CH F *font#* changes the font of an existing alphanumeric field. The command '0, which cycles through font types, is assigned by default to function key [F9] . Pressing [F9] is similar to issuing a '*font#* command for the next font.

Field 3 **J (Justification)**–You have three justifications available: right, left and center. The command 'J *justification* sets the default justification. The command 'CH J *justification* changes the justification of an existing alphanumeric field.

Field 4 **O (Orientation)**–Text can be oriented either horizontally or vertically. The command 'O *orientation* sets the default orientation. The command 'CH O *orientation* changes the orientation of an existing alphanumeric field. Vertical text appears as horizontal text rotated 90 degrees counter-clockwise.

1. Position the cursor in empty space within the drawing. The ALPH status field looks like this:

```
ALPH
NEW
F  2
J  L
O  H
```

"NEW" tells you that there is currently no alphanumeric field. The remaining fields indicate the characteristics of any new text added (font 2, left justification, horizontal orientation). The first entry in the ALPH status field, in this case "NEW," is displayed in the default font, font 2.

2. Position the cursor on a pin name, such as AD1, in the reference 8185. When the name highlights, you know that the cursor is within the text field. The ALPH status field looks like this:

```
ALPH
OLD
F  2
J  L
O  H
```

"OLD" indicates that the remaining fields give the characteristics of the highlighted text.

3. Position the cursor on the small number 18 in the upper right-hand corner of the reference 8185. The ALPH status field looks like this:

```
ALPH
ONP
F  1
J  L
O  H
```

"ONP" indicates that the remaining fields give the characteristics of the highlighted text.

Step 7. Alphanumerics

Enter Text

To add text to a drawing, use [Esc] to enter and exit alphanumeric mode.

1. Position the cursor in empty space. Enter alphanumeric mode by pressing [Esc]. The alphanumeric cursor appears. If you want, you can move the graphic cursor out of your way using the mouse.

2. Type a few characters. The cursor moves right as you type. This is because you are typing with left justification.

3. Exit alphanumeric mode by pressing [Esc] again. If you moved the graphic cursor, it returns to the position it had before you entered alphanumeric mode.

Change Alphanumeric Defaults

When new text is entered, it is given the default characteristics for an alphanumeric field. You can use the commands below to change the default characteristics.

4. Change the default justification to right and the font size to 3 by entering the commands:

   'J R
   '3

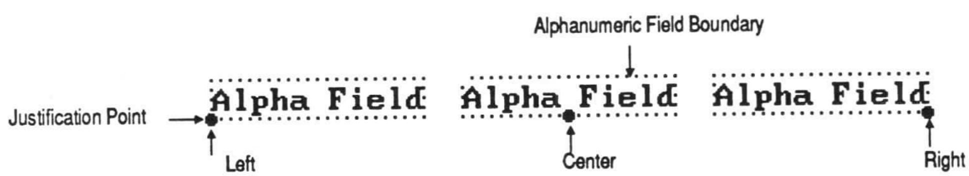
   (Or, you could have pressed [F9] until font '3 displayed in the ALPH field.) Notice the changes to the ALPH field. Enter alphanumeric mode and enter a line of text. Note that the text is larger, and issues from the cursor and moves to the left.

5. Exit alphanumeric mode.

View Alphanumeric Field Boundaries

As with symbols, each block of alphanumeric text is enclosed in a field boundary.

6. Enter the 'B command to toggle the display of alphanumeric field boundaries. The dotted line around a line of text is the boundary, and the small dot is the justification point. If you had entered a field with center justification, the dot would be in the center of the line.



7. Reenter the 'B command to disable the boundary displays.

Step 8. Manipulate Alphanumeric Fields

Move/Copy/Erase

You have seen that each line of text has its own boundary. Because of this, a text field can be tagged and operated on much like a symbol or an area, except that you cannot reflect it. In addition to the specific commands, you are able to use the mouse shortcuts for MOVE, COPY and ERASE.

Be aware of the following:

- Alphanumeric mode is used for entering and editing text and adding over- or underscores. Commands such as MOVE, COPY, and making changes in the size or appearance of the text are done outside alphanumeric mode.
- Each line of alphanumeric text is its own field.

1. Position the cursor inside an alphanumeric field you created in the last step, (you can tell you are in one when the text highlights). Press the left mouse button. The alphanumeric field is tagged. Experiment with the mouse shortcuts to move, copy and erase the text. (If necessary, review Session 1, Step 5.)

Note: *Alphanumeric fields inside of symbols are lower in the tagging hierarchy, meaning that the symbol will be tagged before an alphanumeric field. Keep pressing the left mouse button until the field is tagged.*

Overscore/Underscore

In addition to changing font size and justification, you can also change other alphanumeric attributes, including overscore and underscore.

2. Any alphanumeric field can be overscored or underscored. Try these commands:
   - The commands 'OVER and 'UNDER toggle an overscore or underscore for the current alphanumeric field when you are not in alphanumeric mode.
   - The key presses [Ctrl] – [O] and [Ctrl] – [U] toggle an overscore or underscore for the current alphanumeric field when you are in alphanumeric mode.

Editing Alphanumeric Fields

To edit an existing alphanumeric field, you use some of the cursor control commands you use when creating an alphanumeric field.

3. To make sure that you edit the correct alphanumeric field, position the cursor in the middle of a field to be edited, then enter alphanumeric mode. Be aware of the following:
   - The cursor moves to the left-most edge of the text, regardless of justification.
   - When you first enter alphanumeric mode, new text types over old text. Press [Ins] if you want to insert rather than type over text.
   - When overtyping text in right-justified fields, the text types straight over the old text until the right edge of the field is reached, then the text begins to move to the left.

On Your Own

Now that you know how to add new text, edit existing text and change default alphanumeric attributes, take some time to experiment with the alphanumeric commands. Pay special attention to using different justifications.

4. Try changing the font of an existing field using the 'CH F *font#* command.
5. When you are finished, delete all pieces of text in empty space in your drawing. SAVE the drawing.

Step 9. Insert Pin Names Using a Command List

You have just seen how to enter individual bits of text, which might be used to name isolated pins. But, looking at the reference 8185, you can see that the pins running down the left side have very similar names.

The insert text command, 'I*text*, used in conjunction with cursor movement commands, can be used to make a command list that offers a nice advantage: it inserts the text for you, and numeric data automatically increments.

1. Verify, and set if necessary, the following:
   - Font 2 ('2)
   - Left Justification ('J L)
2. Position the graphic cursor to the top left corner of the solid symbol boundary and then down 4 display units—1 display unit lower than the pin. (Aligning the cursor exactly with the pin places the text too high.)
3. Type the following command list:

   'I AD0;DOWN 3

   The 'I command inserts the alpha field (AD0) at the graphic cursor location and the DOWN 3 instruction moves the cursor down the specified number of display units to the beginning of the next alphanumeric field.

   Note that the "0" is a zero.
4. Press [↵] or middle mouse button. Three things happen:
   - The pin name AD0 appears.
   - The cursor moves down to the next location.
   - The pin name increments to AD1 on the command line.

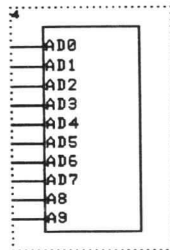
Note: Not all numeric data entered via a command list increments. Only the right-most numeric field increments. For instance, if you were naming pins beginning with A2DX1, only the "1" would increment.

5. Continue naming the pins through AD7.

6. When you complete pin AD7, you need to change the pin naming convention slightly. Edit the pin name on the command line to look like the one below (remove the "D") which will make the next two names A8 and A9:

   'I A8;DOWN 3

7. Add those names. The 8185 should look like this:



Step 10. About Attributes

The pins are now ready to have their numbers added. Because the pin number carries the electrically significant information, we need to explain FutureNet attributes.

Attributes

An attribute controls the interpretation of the information in an alphanumeric field. Attributes can either be electrical or graphic.

Electrical attributes are signal attributes such as POWER, GND, +5VDC, and pin attributes such as PINI (input) PINO (output) and PNBT (tri-state). FutureNet has many electrical attributes which are crucial to producing schematics that carry the information you need. For instance, when FutureNet sees an alphanumeric field **CLK with an attribute of SIG, it knows that it is looking at a signal line named CLK**. When this line connects to an alphanumeric field of 8 with an attribute of PINI, it knows that it is looking at input pin number 8.

Graphic attributes are used to define non-electrical elements of a drawing, such as PART, LOC (the reference designator) and TITL (the drawing title).

Both electrical and graphic attributes are significant for drawing post processing. For instance, most PCB packages must have PART and LOC information for packaging and placement routines.

How Are Attributes Displayed?

An attribute is assigned to every piece of text (name or number) in a drawing, except for the symbol reference number. The attribute for each item is displayed in the ATTR field when the cursor moves onto the item.

1. Move the cursor into a blank area of the screen. Note that the ATTR field says COM 0. This means comment or common, and is an attribute of 0. It is not electrically significant (it is a synonym for GROUND).

   The attribute which is displayed when the cursor is in a blank area of the screen is the default attribute.

2. Move the cursor over a pin number (such as 1) of any pin on the reference 8185. The number highlights to let you know that the cursor is actually on the field. The ATTR field is updated with the name of the attribute and the attribute number.

3. Move the cursor over various features of the reference 8185. You will see values such as:

   - LOC 2 (the reference designator or U number)
   - PART 3 (the part number)

Working with Attributes

Each alphanumeric field is assigned an attribute, whether it's the default attribute COM 0 or one that gives more information about the field. The easiest way to assign an attribute for an alphanumeric field is to assign the attribute before you begin typing.

One of the attribute values for a pin is PNBT 22 (bidirectional tri-state). It can be assigned to any field that numbers a pin. There are two ways to assign attributes to alphanumeric fields:

- To set the default for new fields, use the 'A *attribute* command. The default attribute is automatically assigned to each new field as it is entered.
- For an existing field, 'CH A *attribute* changes the attribute currently assigned to the field.
- Note that you can enter either the name or the number of the attribute. For instance, if you want the attribute PNBT 22, enter either PNBT or 22.

4. You can change the default pin attribute two different ways: either from the command line or by using menus.
5. If you want to use the command line to change the default pin attribute, move the cursor to a blank area of the screen and enter the 'A *attribute* command. Enter either

   'A PNBT

   or

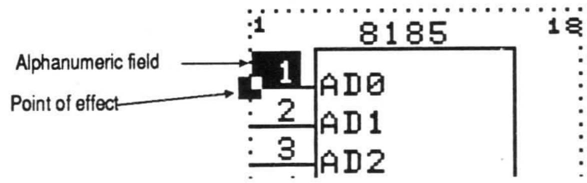
   'A 22

   The ATTR status field now shows PNBT 22. This is the attribute that will automatically be assigned to any new alphanumeric data you enter.
6. If you want to use the menus to change the default pin attribute, move the cursor into a blank area of the screen and enter the 'A command. The Attributes menu appears.

   Select Pin from the Attributes menu and then select PNBT from the Pin Attributes menu. The ATTR status field now shows PNBT 22. This is the attribute that will automatically be assigned to any new alphanumeric data you enter.

Point of Effect

As you move the cursor around looking at attributes, you may notice that when the cursor moves into a field and the text highlights, a small square also appears. The square is called the point of effect, and it lets you know which item the alphanumeric field is linked to.



7. Move the cursor over a pin number (such as 1) of any pin on the reference 8185. When the pin number highlights, the point of effect also appears. The point of effect lies on the dotted symbol cell boundary.

8. Move the cursor over a number of items, this time noting the location of the point of effect.

Chapter 1, "Understanding FutureNet," contains detailed descriptions of attributes and their applications to your designs. Turn there for information about your specific design challenges.

Step 11. Enter Pin Numbers Using a Command List

You are now ready to add the pin numbers.

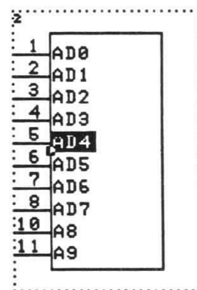
1. Position the graphic cursor at the top pin (AD0), at the left edge of the dotted symbol cell boundary.
2. On the command line, type command list:

   'I 1;DOWN 3

3. Press ⏎ or the middle mouse button until the pins for AD0 through AD7 have pin numbers.
4. After numbering pin 8, edit the command line to read:

   'I 10;DOWN 3

   Add pin numbers 10 and 11. When you're done, the screen looks like this:



Step 12. Complete the 8185 Pins

You are now ready to begin work on the other side of the symbol. Using all of the information, and using the reference 8185 as a guide, complete the right side of your 8185. Keep the following in mind:

- You won't be able to use a command list to add any of the data here because it is all different.
- Note that when adding alphanumeric fields, it's not necessary to leave alphanumeric mode to get to the next field. Just press [↵] for the next line.
- In all fonts, when [↵] is pressed, the cursor moves to the next text field. It is of special interest here that in Font 2, each text field is 3 display units lower than the last—similar to standard pin spacing.

1. Add the required regular (.-) and inverted (.-O) pin stubs, beginning with pin 15 (which is 15 display units down from the top right corner of the symbol cell, directly across from pin 5).
2. Verify, and set if necessary, the following:
   - Right justification ('J R)
   - Attribute COM 0 ('A 0)

   When we added the pin names to the left side of the symbol, we used left justification. For adding text on the right of a symbol, use right justification. In right justification, the cursor remains stationary, and the text moves to the right as it appears. This allows you to accurately place each bit of text.

   To add the first piece of text, position the cursor 1 display unit lower than the first right-side pin (directly across from AD4) and on the symbol cell boundary.

   While in alphanumeric mode, press [Ctrl] – [O] for each field you want inverted (overscored). You can press it with the cursor located anywhere in the field; either before you type anything or while you're typing the field. If you press the key and then realize you didn't want to invert that field, just press the key again to remove it.

   When using an 'I or 'R command list, you use the 'OVER command to overscore the field, as in 'R RD; 'OVER.
3. Add the pin numbers. Set the default attribute at PINI 23 (enter the 'A 23 command).

Step 13. Assign Reference Designator and Part Number

As you can see by the reference 8185, a prototype circuit designator UNNN and the part number 8185 are contained within the cell boundary. Enter those fields for your 8185 now.

1. Position the cursor centered on the symbol cell boundary at the bottom of the symbol.
2. Verify and change, if necessary, the following:
   - Attribute to LOC 2 ('A LOC)
   - Justification to center ('J C)
   - Font to 2 ('2)
3. In alphanumeric mode, enter the reference designator:

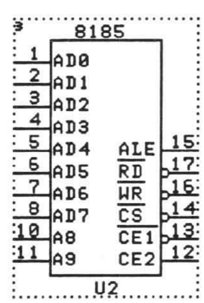
   U2

   Return to graphic mode.
4. Now add the symbol name. Position the cursor centered on the symbol cell at the top of the symbol.
5. Verify and change, if necessary, the following:
   - Attribute to PART 3 ('A 3)
   - Justification to CENTER ('J C)
   - Font to 2 ('2)
6. In alphanumeric mode, type the symbol name:

   8185

   Return to graphic mode.

Your 8185 should look like this:



Step 14. Enter Power and Ground Pin Numbers

You have probably wondered what the small numbers are in the corners of the reference 8185. Instead of cluttering the symbol with pins which are always present, the POWER and GROUND pins are merely noted in the corners of the symbol. These are referred to as "implicit" connections. It is possible to do this because, as we've seen so far, the actual electrical value of a pin lies not in the graphic of the pin, but in it's attribute. For instance:

1. Move your cursor over the small 18 in the upper-right corner of the reference 8185. This assigns the pin tied to the net POWER. Notice the following:
   - The attribute is +5V 101.
   - The ALPH status field says ONP. This means "old and non-printing." To make the schematics easily read, printing of implicit power and ground pin numbers is often suppressed.
   - Font size is 1 (smallest).

Now make the power and ground connections.

2. Prepare for adding the POWER pin number. Verify and change, if necessary, the following:
   - Attribute to +5V 101 ('A 101)
   - Justification to RIGHT ('J R)
   - Font to 1 ('1)
3. Move the cursor to the upper right corner of the symbol cell and then 2 display units down.
4. Enter alphanumeric mode and enter the pin number:

   18

   Exit alphanumeric mode.
5. Move the graphic cursor to the lower right corner of the symbol cell, right at the symbol cell boundary.
6. Change the attribute to GND 100 ('A 100).
7. In alphanumeric mode, type the pin number:

   9

   Exit alphanumeric mode.

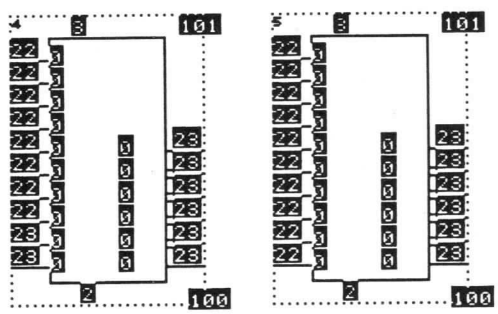
Step 15. Display Assigned Attributes; Change Attributes

Now that you have finished your 8185, you can check it against the reference 8185 to see that all attributes are correctly assigned.

Display Assigned Attributes

Normally, the attributes of a symbol are hidden. Use the 'D command to display the attributes assigned to elements in a symbol.

1. Enter the 'D command. Your symbol will look similar to this:



   The attribute types are displayed in reverse video in place of the alphanumeric fields.

2. Verify that your symbol is identical to the reference 8185.

Change Attributes

As you may notice, the attributes for pins 10 and 11 are wrong: they are PNBT 22 instead of PINI 23. To change them, do the following:

3. Reenter the 'D command to cancel the display of attributes.
4. Type on the command line:

   'CH A 23

5. Position the cursor on the pin number of pin 10. Press ⏎ or the middle mouse button. The ATTR field is updated with the new attribute of the field.
6. Repeat Step 5 for pin 11.

Step 16. Save a Symbol in a Library

Now that you have completed this symbol, save it in a symbol library. You must have an update library open for this step.

1. Enter the PROFILE command to check which libraries are in use. Note whether an update library is open.

   Exit the PROFILE screen by pressing Esc or clicking on the CANCEL action button. If you have an update library open, go to Step 4.

2. If you do not have an update library open, enter the .LIB command. You are prompted:

```
Enter symbol library file name
```

3. Enter an appropriate name, such as your own, or the name of the library you created in the last session. If the library you named does not exist, you are asked if you want to create a new update library.

   Answer yes to create and open it.

4. Position the graphic cursor within the symbol. Enter the .SAVE command from the menu. You are prompted:

   ```
   Enter symbol name
   ```

5. Enter the name of the symbol:

   8185

Symbol 8185 is now saved in the update library you opened.

Step 17. Check a Library

If you want to check that symbol 8185 did get saved into your update library, use the .DIR command to check.

1. Enter the .DIR command. A list of the symbols in your update library, a preview of symbols, and the reference libraries appears. Use the left mouse button to page through the list.

2. Once you find your 8185 listed, press Esc (or the right button on the mouse outside of the window) to return to the drawing. If you can't find it, repeat Step 16 to save the symbol.

Step 18. Clean Up and Save the Drawing

1. Delete the reference 8185 (use .E). You only need one in your drawing at this point.

2. Verify your drawing against the drawing at the end of this chapter.

3. Save the drawing.

Looking Ahead

In the next session, you complete placing all of the symbols in your drawing and begin connecting them together. Some of the functions you will learn are:

- Reflecting a symbol
- Drawing lines
- Rotating an area

If You Are Continuing

If you are continuing immediately with the next session, enter the SAVE command to save your drawing, then continue.

If You Are Stopping Here

If you are stopping here and plan on exiting FutureNet, enter the SAVE command, then the QUIT command.

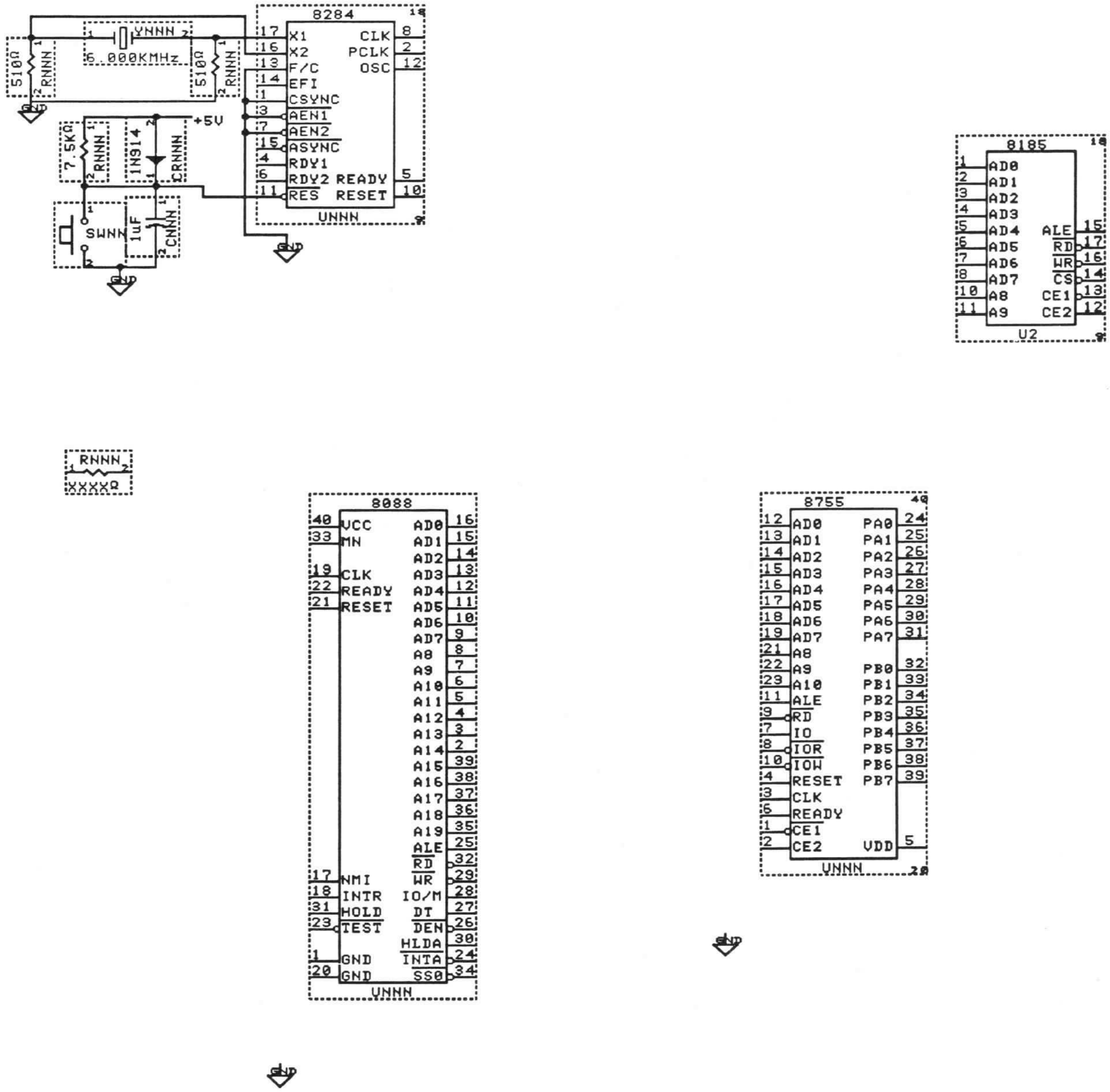
Summary

You have now completed Session 2. In summary, here are the commands you learned:

	Command	Function
Keys	DOWN ##	Moves the cursor down the indicated number of display units.
	Ctrl –**Arrow Key**	Moves the cursor on the command line.
	Del	Deletes a character on the command line.
	Ins	Inserts a character on the command line.
	Esc	Enters and escapes alphanumeric mode.
	Ctrl – O	In alphanumeric mode, adds and deletes an inversion bar.
	Ctrl – U	In alphanumeric mode, adds and deletes an underscore.
	Tab	Moves the cursor to the next alphanumeric field in the symbol.
	Shift – Tab	Moves the cursor to the previous alphanumeric field in the symbol.
Block Symbols	.B *w,h*	Creates block symbol.
	.-	Draws a pin stub.
	.-O	Draws an inverted stub.
	.D	Deletes a pin.

Alphanumerics	'I *text*	Inserts alphanumeric data (*text*) and increments a numeric field.
	'R	Replace alphanumeric text.
	'OVER	In a command list or from the command line, toggles an inversion bar.
	'UNDER	In a command list or from the command line, toggles an underscore.
	'D	Enables and disables attribute display.
	'B	Enables and disables alphanumeric field boundary display.
	'M	Moves an alphanumeric field.
	'A *attribute*	Sets default attribute for a new field.
	'CH A *attribute*	Sets an attribute for an existing field.
	'J *justification*	Changes the default justification.
	'CH J *justification*	Changes the justification of an existing field.
	'O *orientation*	Changes the default orientation.
	'CH O *orientation*	Changes the orientation of an existing field.
	'*font#*	Changes the default font.
	'CH F *font#*	Changes the font size of an existing field.
Libraries	.LIB	Specifies the update symbol library.
	.SAVE	Saves a symbol to a library.
	.DIR	Lists symbols in libraries.

session2.dwg



9 *Session 3: Drawing Lines and Making Connections*

The symbols in your design are connected with line drawing commands. This session covers:

- Drawing lines
- Editing lines
- Line types

Step 1. Before You Begin

1. Start FutureNet.
2. Do one of the following:

   If you do not want to use your own drawings, or if you have not completed Session 2:

   - LOAD the Data I/O-supplied drawing **session3**.

   If you are beginning the tutorial sessions again and have completed Session 2, do the three steps below. If you are continuing from the previous session, skip the first step:

   - LOAD the drawing you created in the last session.
   - SAVE the drawing as **gs3**. This creates a copy of the drawing you created in the last session.
   - LOAD the drawing **gs3**. This loads the copy of your drawing. Thus, any changes you make to the drawing will be made to the copy and not to the original.

Step 2. Introduction to Lines

The LINE Status Field

Lines in FutureNet can either indicate connectivity, or can simply be notation. Both use specific line types. The LINE status field lets you know the current line type.

Line Types

Wires and buses are lines that indicate connectivity. The wire line type is /1, the bus line type is /2. These lines are meant to connect one pin to another. It is their only purpose.

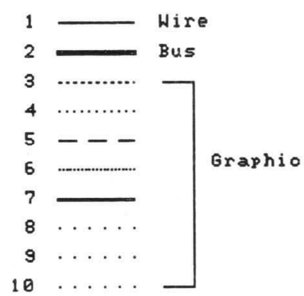
Notational line types are types /3 through /10. These lines can be placed anywhere in the drawing, for example, to separate a comment area or surround a detailed inset.

Changing Line Types

The line type can be changed in two ways:

- Enter the command that corresponds to the type of line you want to use. For instance, to change to a /5 line, you enter the /5 command.
- Enter the /0 command, which cycles to the next line type. The /0 command is also assigned by default to F10 .

In either case, the LINE status field is updated with your choice.



1. Enter one of the line type commands, for instance, /8. The LINE status field is updated.
2. Press F10 repeatedly, watching the LINE status field. The line types cycle through.

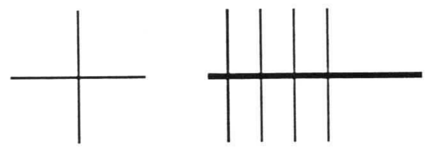
Note: Always check the LINE status field to make sure that the line type is set correctly before you start drawing. If you draw a line of the wrong type, just draw over it with a line of the right type.

Making Connections

Connected lines:

- When a line "dead ends" into another line, the two are considered to be connected.
- A line may make any number "jogs" and remain connected with itself.
- Lines that cross are connected when a connect dot is present.

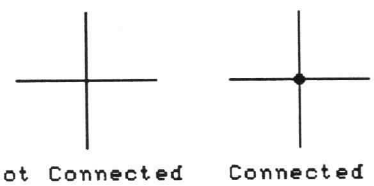
Non-connected lines:



- Lines may cross in any combination and frequency and are not connected.

The Connect Dot

As stated above, lines that cross are considered not connected. When it is necessary to indicate a connection, FutureNet provides a connect dot.



A connect dot can be placed anywhere on a line to indicate connection to a line that crosses it.

The /D command toggles connect dots. (The /D command has a default assignment to the F5 key.)

Step 3. Line Drawing Commands

Line Types	/0	Select next line type
	/1 through /10	Select line type
Draw/Erase	/L	Begin/end line segment
	/K	Escape line drawing
	/E	Erase line
	/LE	Draw/erase line
	/EL	Erase line segments
	/EN	Erase line network
	/ES	Erase line segment
	/ET	Erase temporary lines
Miscellaneous	/C	Draw direct connection through symbol
	/D	Toggle connect dot
	/P	Convert temporary line to permanent
	/R	Change line routing
	/V	Move vertex
	/J	Draw junction segment
	CONNECT	Enable/disable maintenance of line connections
	PINSNAP	Snap line to pin

Step 4. Line Drawing Basics

Line Drawing

Now that the line type has been selected, you are ready to begin drawing lines.

1. Position the cursor in a blank area of the screen. Press the left mouse button. The line drawing cursor—a large crosshair—appears. The MODE field says LINE.
2. Move the cursor to the right about an inch and then down an inch.

   These two segments, shown as dotted lines, are called routing segments. They are not permanent lines, but show you where the permanent lines would be drawn if they were set in place. A couple of things are important to understand here:

   - FutureNet chooses the routing of the second routing segment. You have the opportunity to reverse the routing by pressing the middle mouse button or entering the /R command.
   - FutureNet only draws straight lines and right angles. Diagonal line capability is limited to junction segments, which are actually a symbol and not a true line (these are discussed later).
3. Move the cursor around. Routing segments expand and contract to keep the starting point attached to the cursor.
4. Press the middle mouse button. The routing of the segments changes.
5. Suppose, however, that you decide against these lines altogether. Press the right button. Both lines disappear because they were merely routing segments, the MODE status field says MENU, and the graphic cursor reappears.
6. Re-create your two-line segment. Set the first segment by pressing the left mouse button. It is now a permanent line. You could now move in the other direction with a new routing segment, but for right now, press the left mouse button again. The second segment is set in place. Press the left mouse button to exit line drawing mode, and the graphic cursor reappears.

On Your Own

Practice drawing simple and complex lines.

- Draw lines using the command line and the cursor control keys by following the steps above but, instead of using the left mouse button in LINE mode, use the /L command, reentering it where necessary by pressing the middle mouse button.

Note: *Once all routing segments are set in place, FutureNet is still in LINE mode, ready to continue drawing. If you want to exit line drawing mode, you must press the left mouse button again or enter the /L command until the graphic cursor reappears.*

Step 5. About Editing Lines

A number of commands exist enabling you to edit the line segments you draw.

The /ES Command

The /ES command erases individual segments of a line.

1. Complete a 4-segment line. Position your cursor on any segment of the line and enter the /ES command. The segment at the cursor position disappears.

   With this command on the command line, you can repetitively erase segments by positioning the cursor and pressing the middle mouse button or by pressing [↵].

The /EL Command

The /EL command erases segments of a line until a symbol or some type of interconnection is reached.

2. Complete another line of any number of segments. Position your cursor on any segment of the line and enter the /EL command. The entire line disappears.

   With this command on the command line, you can completely remove lines that are incorrect. The /EL command works by erasing all segments of a line until a symbol or some type of interconnection is reached.

The /E Command

The /E command erases a portion of a line segment.

3. Complete another line of any number of segments. Position your cursor on any segment of the line and enter the /E command. You enter LINE mode.
4. Draw over a portion of the line, and the routing segment overlays the permanent line. Press the left mouse button. The portion of the permanent line drawn over disappears. Press the left mouse button again to exit the /E command.

   With this command on the command line, you can easily make minor corrections to the way lines have been routed: press the middle button to enter the /E command and draw over lines you don't want; press the left button to begin a line that you do want.

Move a Vertex

With the /V command you can relocate a corner, thereby stretching or shrinking the two lines connected to it, or a line, thereby stretching and shrinking any other lines connected to it. Connections are maintained wherever possible, allowing you to stretch and shrink lines that have already been set in place. The start and endpoints of any lines affected remain fixed. Connect dots are added where necessary. To see how this works, do the following:

5. Complete a 3-segment line.

6. Position the cursor on the middle segment. You can enter the /V command from the command line, or by using the mouse. To use the mouse, press the left button once—the line drawing cursor appears; press the left button again—a small box appears. The small box is the vertex marker.

1. Tag the vertex
2. Move it

3. Confirm the new location

7. Move the vertex marker to a new location and press the left mouse button to set the new vertex. The lines adjust to their new location.

On Your Own

Experiment with the line drawing commands. Pay special attention to:

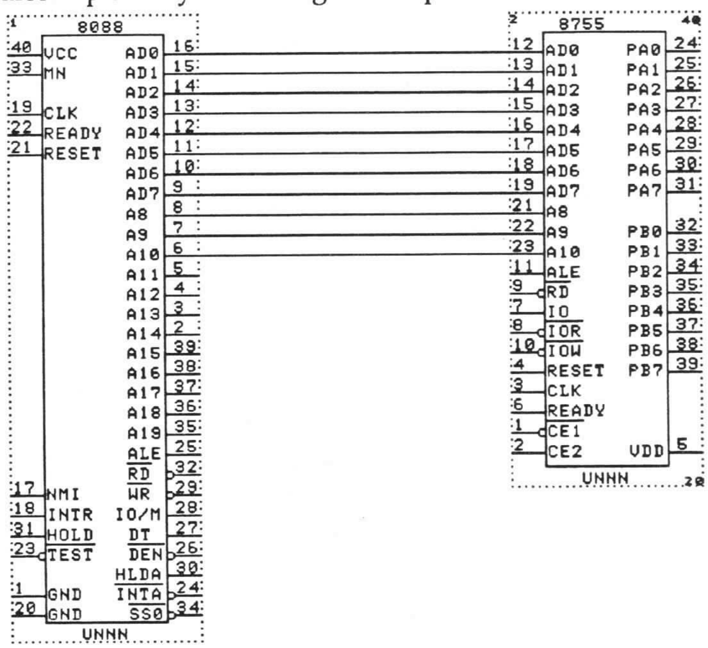
- The /EN command. This command erases entire networks of lines. To try this command, draw a series of interconnecting lines (use "Ts" or connect dots). When you enter the /EN command, all lines in the net are erased. Because this is a network command, it only works reliably for /1- and /2-type lines. Use /EN carefully!
- Work with the /ES command until you understand just exactly what it removes: portions of lines until a symbol or interconnect is reached.
- The /V command. Because the /V command only moves a segment of a line between connections, you can use connect dots (/D) to define the vertexes to be moved.

Note: Before going on, delete all of your experimental lines.

Step 6. Straight Line Connections

With this background in drawing lines, begin the first set of lines. The illustration notes the order in which the lines should be completed. Follow the steps below for the first line.

1. Place the cursor near pin 16 of the 8088, until the pin number's alphanumeric field and the point of effect is highlighted. Do not line the cursor up exactly on the edge of the pin stub.



2. Press the left mouse button to begin line drawing and move the cursor to the right until the point of effect is not longer highlighted. Notice that the routing segment is attached to the pin, even though the cursor was not directly on the pin. This is a function of the PINSNAP command, which automatically snaps line segments to the point of effect, allowing you to route quickly.

   Note also that the pin stub is a graphic element that is not needed for connectivity; it is the points of effect of the pin numbers that are being connected.

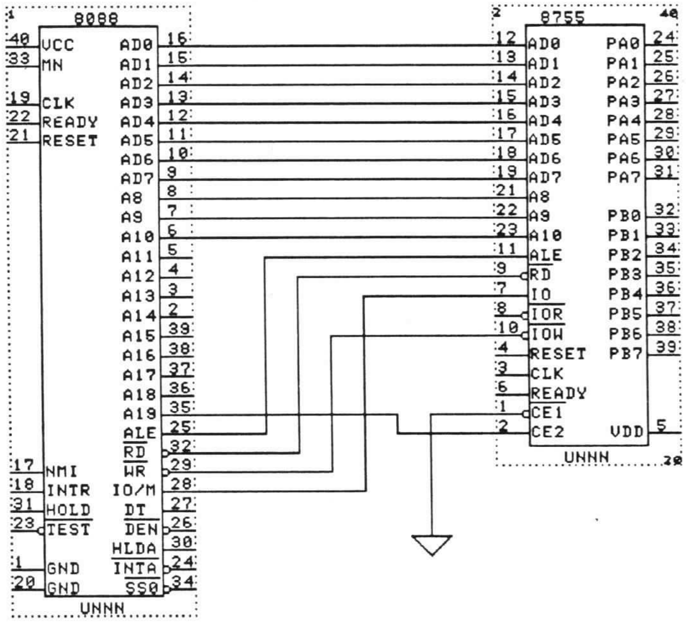
3. Bring the routing segment over to the 8755, pin 12, until the point of effect is highlighted. Press the left mouse button to set the routing segment, and again to exit LINE mode. PINSNAP attaches the routing segment to the pin stub at the point of effect.

Note: *If the pins do not line up, press the right mouse button to exit LINE mode, align the pins by moving one of the symbols, then begin drawing again.*

4. Complete the rest of the connections as noted in the drawing above. To make drawing the lines easier, type **3** on the command line. Then, once a line is completed, you can press ↓ to move immediately to the position of the next line and begin drawing.

Step 7. Drawing Multi-Segment Lines

Multi-segment lines are used to connect pins not directly opposite each other. Drawing this kind of line is an extension of a single-segment line. The following illustration shows the connections made in this step.



1. First, draw the line from 8755 pin 11 to 8088 pin 25. Make the first segment of the line 21 display units long. This is the least distance required for all of the connections to fit in.

2. Complete the remaining lines in this area in the order shown on the illustration, noting the following:

   - Lines usually maintain a 3 display-unit spacing, duplicating the pin spacing for most chips. Preserve this distance when making a jog in a line by hugging other lines.

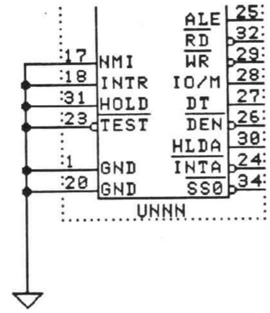
   - When connecting to GND, try to connect without making a jog in the line. You may need to move GND to the line.

On Your Own

As you go, experiment with changes in line routing by pressing the middle mouse button when you have two routing segments on the screen.

Step 8. Draw Using a Command List; Connect Dots

The following illustration shows where the next group of lines will be drawn—the GND for the 8088.



Although these connect dots are not absolutely necessary here (lines that dead-end into other lines are considered to be connected), the dots provide a way of not only making the connections clear, but preserving the connections if the GND must be moved.

Because this group of lines is essentially a repeated series, it gives us an opportunity to look at drawing using a command list.

1. Position the cursor at 8088 pin 17 on the left symbol cell boundary.
2. Enter the following commands (which appear in the left column) on the command line:

/L;	Begin line drawing; the semicolon separates the commands
LEFT 5;	Move left 5 display units; command separation
/L;	Set the routing segment in place; command separation
/L;	Exit line drawing; command separation
DOWN 3;	Move to the next line location; command separation
RIGHT 5	Move back to the pin (no command separator)

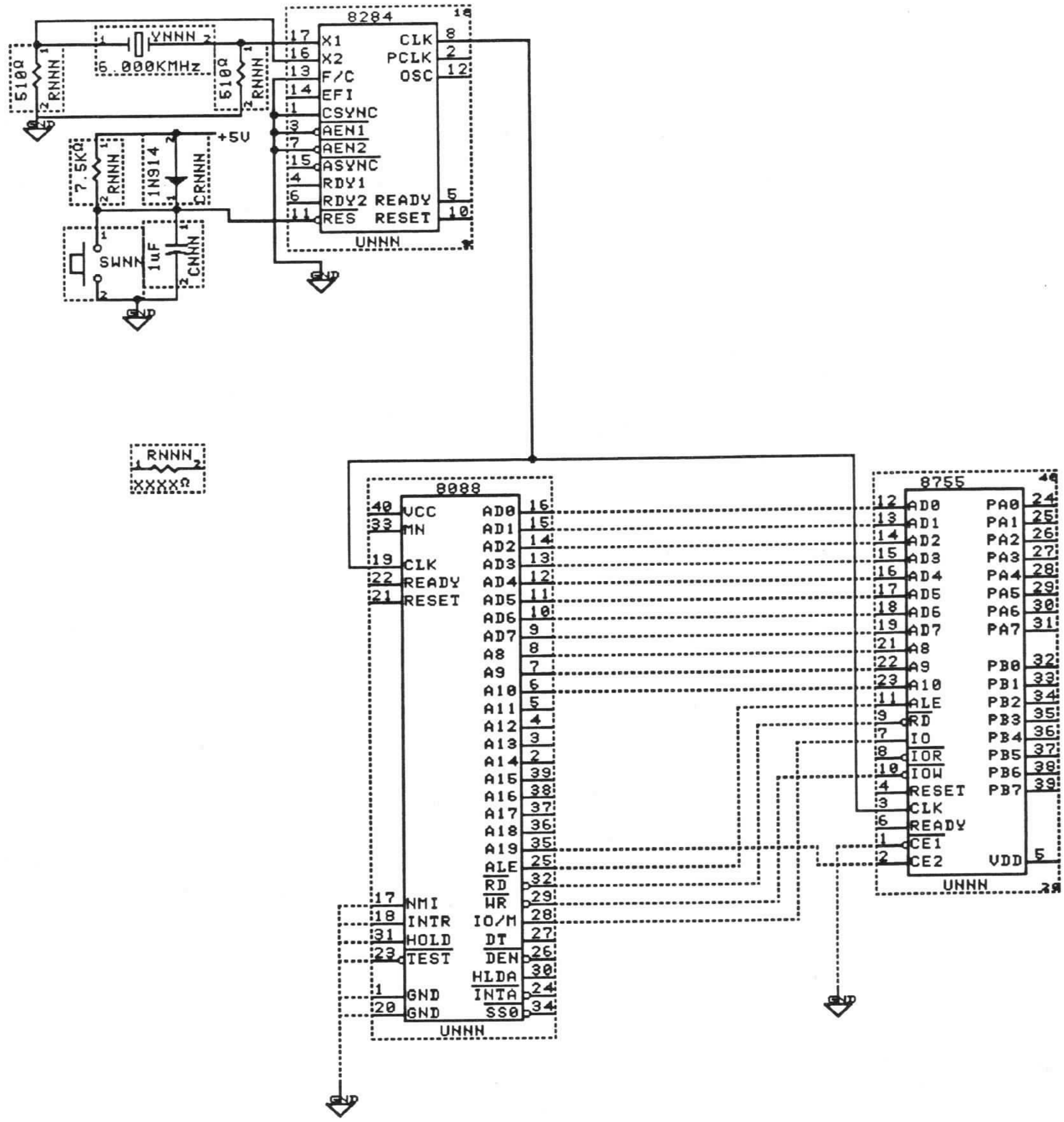
3. Press [↵]. The first line is drawn and the cursor positions itself correctly for the next one.
4. Complete the lines from pins 18, 31 and 23, then use the mouse or cursor control to position the cursor at 1, right on the symbol cell boundary, for the remaining two lines.
5. Draw the vertical segment linking all of the horizontals to the GND symbol (move the GND, if necessary).

6. And, just to show you how flexible FutureNet is, choose one of these methods for adding the connect dots:
   - Add each one using the /D command.
   - Add each one using the [F5] key, to which the /D command is assigned by default.
   - Write a command list, similar to the one above.
   - Use a combination of [F5] and 3 (on the command line) [↓] .
7. When complete, check your work against the illustration above.

Step 9. Route the CLK Network

The next series of lines creates the CLK network. Using the following illustration as a guide, complete the CLK network, keeping these things in mind:

- The line from 8088 pin 19 to 8755 pin 3 hugs both symbols at a 3-display unit distance.
- The horizontal segment from 8284 pin 3 should be 9 display units long, to leave room for other connections.
- Remember to add the connect dot.



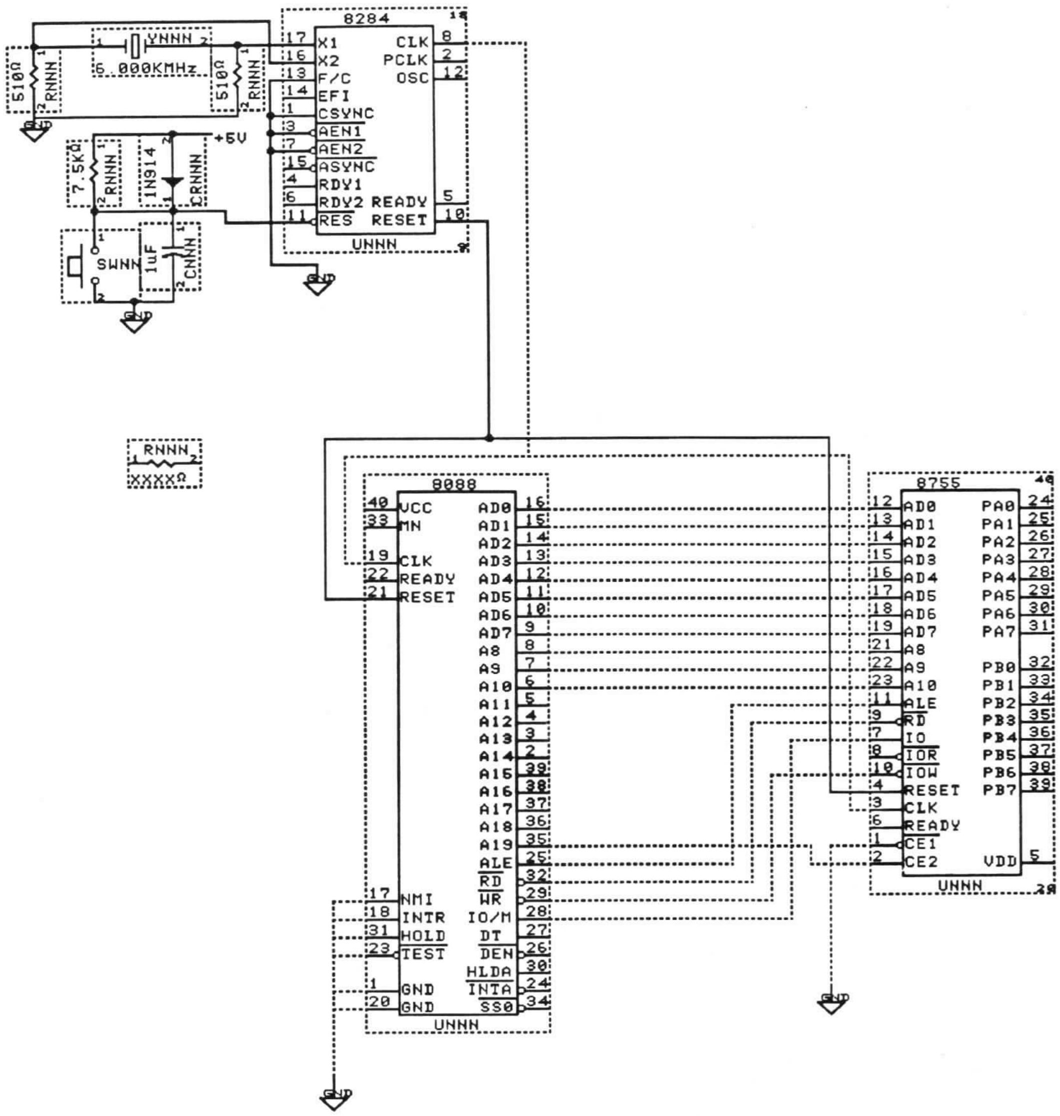
Step 10. Route the RESET Network

These lines create the RESET network. Using the following illustration as a guide, complete the RESET network, keeping these things in mind:

- The line from 8088 pin 21 to 8755 pin 4 hugs the CLK line.
- The horizontal portion of the line from 8284 pin 10 should be 3 display units long to leave room for other connections.
- Remember to add the connect dot.
- A 3 on the command line used in concert with the [↑] is a convenient way of helping you preserve a spacing of 3 display units between lines.

On Your Own

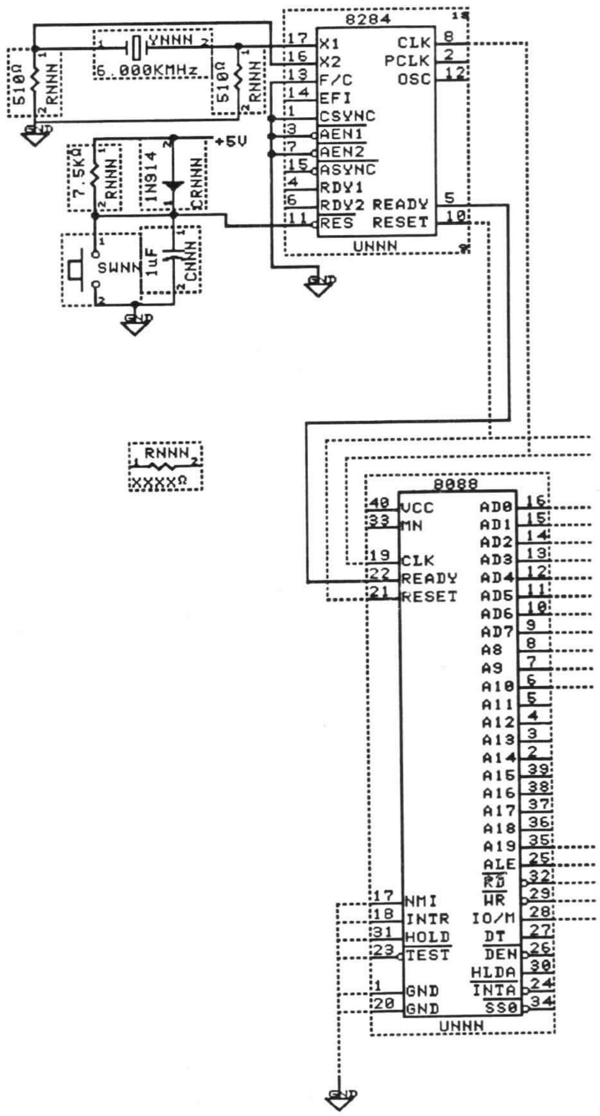
With these lines in place, experiment with the /V command some more. For instance, if you try to move the horizontal section of the CLK line between the 8088 and the 8755, you'll find that, because of the connect dot, you'll have to actually move two segments of the line.



Step 11. Route The READY Network

You can build the READY network in two steps. The first involves a connection between the 8284 and the 8088. The second creates a resistor network.

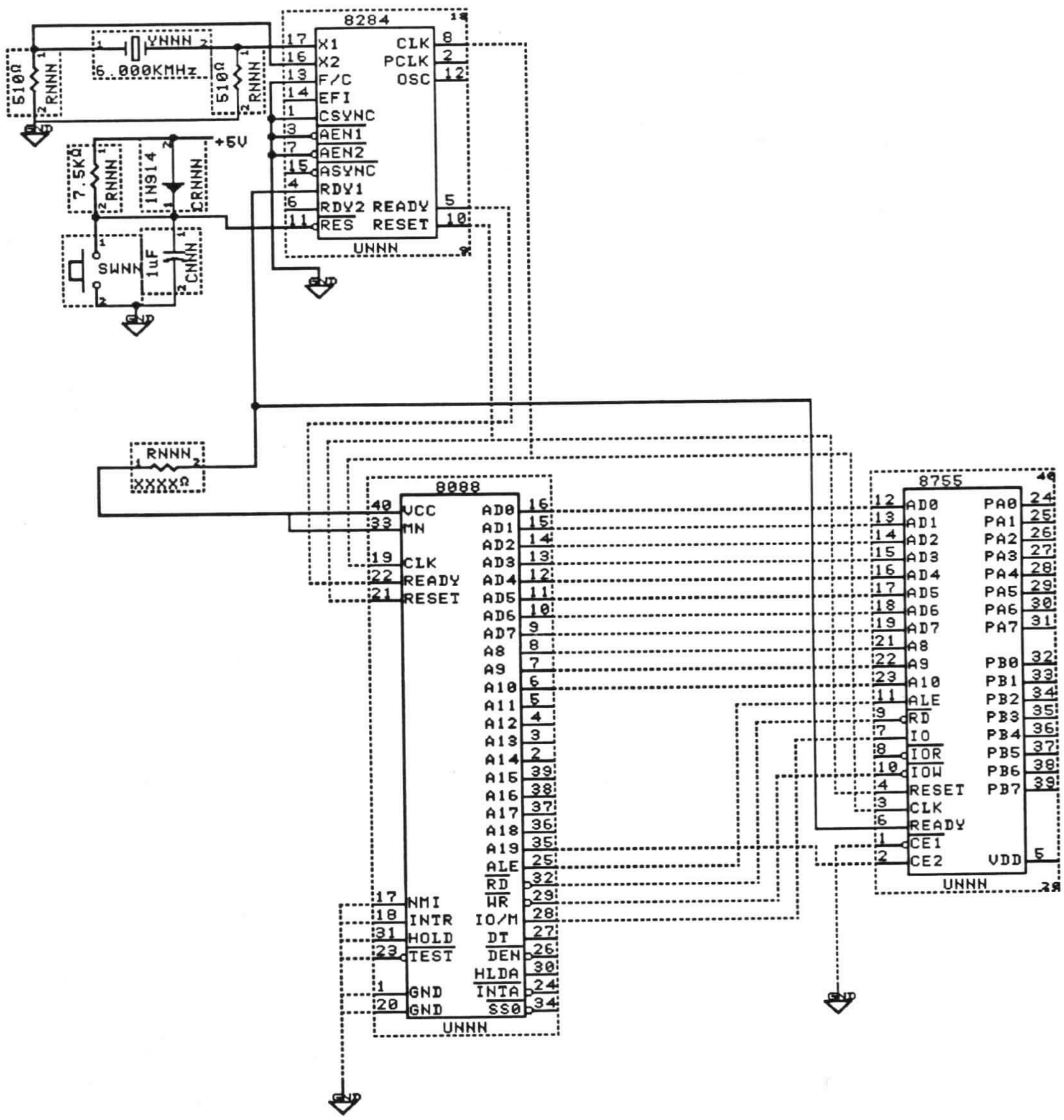
1. Referring to the following illustration, add the line from 8088 pin 22 to 8284 pin 5.



2. Referring to the following illustration, build the resistor network. Note the following:

   - You may have to move the resistor. Make sure that the line coming from 8088 pin 40 maintains a 3-display unit spacing around the left edge of the resistor.

   - Observe 3-display unit spacing on the horizontal segment going into 8284 pin 4.



Step 12. On Your Own

There are many things that you can experiment with at this point.

- Using the /V command (or tagging a vertex or line using the middle mouse button), reroute some line segments. Be aware, however, that if you inadvertently move a line segment on top of another, the two become the same line.
- Experiment with all the line editing commands.
- Hone your skills in writing command lists. A key to using a command list is to be aware of the commands that perform functions similar to the mouse buttons.
- Turn to the description of the FAST command in the *Command Reference* manual. Entering the FAST command places FutureNet in FAST drawing mode. Try FAST drawing.
- You can shorten and lengthen groups of lines by defining portions of them as areas, and then moving or deleting them, as required. Try this. (But SAVE the drawing first; if you need to return to your original drawing, reLOAD the drawing, abandoning your changes.)
- Depending upon what you do, FutureNet may display one or more dotted red lines. These are temporary lines and indicate that FutureNet could not adequately reroute the lines. Turn to Session 5 for a discussion on temporary lines, or to the *Command Reference* manual, /ET command.

Looking Ahead

In the next session, you learn about the following:

- Junction segments
- Creating and naming buses
- Manipulating points of effect
- More information on area manipulations

If You Are Continuing

If you are continuing immediately with the next session, enter the SAVE command to save your drawing, then continue.

If You Are Stopping Here

If you are stopping here and plan on exiting FutureNet, enter the SAVE command, then the QUIT command.

Summary

You have now completed Session 3. In summary, here are the commands and functions you learned:

Line Drawing Commands

Command	Function
/0	Select next line type
/1 through /10	Select line type
/L	Begin/end line segment
/K	Escape line drawing
/E	Erase line
/LE	Draw/erase line
/EL	Erase line segments
/EN	Erase line network
/ES	Erase line segment
/ET	Erase temporary lines
/D	Toggle connect dot
/R	Change line routing
/V	Move vertex

The Mouse

Left Button

In MENU mode, begins line drawing when the cursor is in a valid location. Once LINE mode has been entered, makes routing segments permanent, and ends line drawing mode.

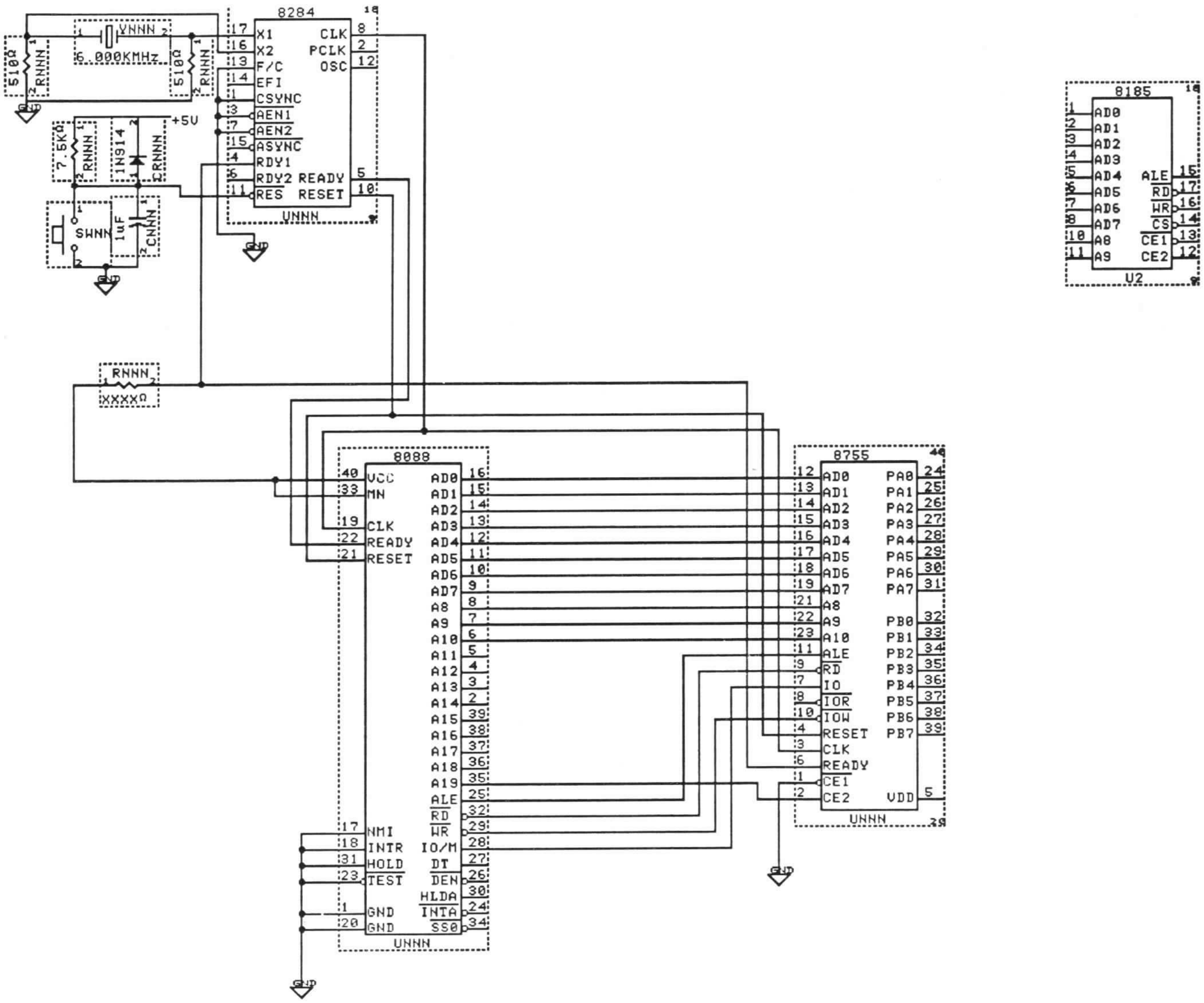
Middle Button

In MENU mode, reenters the command on the command line. In LINE mode, switches the vertex of two routing segments. If you are using a two-button mouse, press both buttons of the two-button mouse to simulate the middle button of a three-button mouse.

Right Button

In MENU mode, calls up the menus. In LINE mode, cancels all routing segments and exits LINE mode.

session3.dwg



10 *Session 4: Creating and Naming Buses*

This session continues working with the 8088 processor design that was begun in the previous sessions.

This session focuses on creating and naming buses and manipulating points of effect.

Step 1. Before You Begin

1. Start FutureNet.
2. Do one of the following:

   If you do not want to use your own drawings, or if you have not completed Session 3:

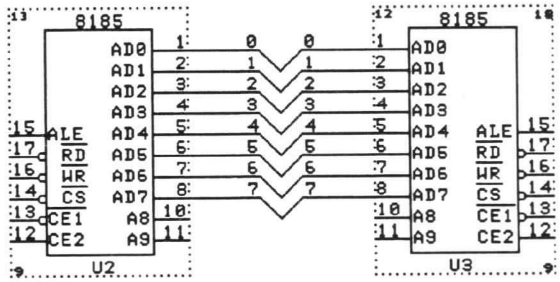
   - LOAD the Data I/O-supplied drawing **session4**.

   If you are beginning the tutorial sessions again and have completed Session 3, do the three steps below. If you are continuing from the previous session, do the second and third steps:

   - LOAD the drawing you created in the last session.
   - SAVE the drawing on your screen as **gs4**. This creates a new drawing but does not place it on your screen.
   - LOAD the drawing **gs4**.

Step 2. Add Junction Segments

As you can see in the following illustration, the 8185s hook up to two separate bus lines. Using a diagonal junction graphic, the connections are easy to draw.



Junction segments are actually a second type of symbol: graphic symbols. The symbols you've dealt with so far have been circuit symbols.

Circuit symbols are electrically significant; graphic symbols aren't. The symbol cell boundaries and reference numbers are displayed in the drawing for circuit symbols and are not displayed for graphic symbols. Both types of symbols are created and handled the same by FutureNet; it's in the Data I/O Drawing Preprocessor that they're processed differently.

Look at a junction segment.

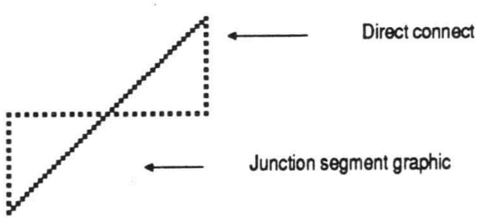
1. Position the cursor in a blank area of the screen. Enter the command:

   /J UR,5,D

   - UR is direction, in this case, Up Right from the cursor position. Other directions could be DL (down left) and RD (right down). As long as you specify a horizontal and vertical direction, the order of the two is unimportant: UL and LU are both equally valid.
   - 5 is the size, which actually denotes the boundary of the symbol, in this case, a 5-display unit square containing a diagonal line. The size can be from 1 to 10 display units.
   - D is the type, in this case, Diagonal. The other type is C (curved).

2. Enter the .DCON (direct connect) command to display direct connections.

   Your junction segment looks like this:



Notice the dotted line that connects the ends of the junction segment.

Junction segments are graphic symbols with a direct connection drawn through them. Consequently, no line drawing or erasing commands work on junction segments; you must use symbol commands on them. For instance, to delete a junction segment, you must use the .E command.

Also notice that, when the junction segment appears at the cursor position, the cursor does not move. This is because a junction segment is not a line—it's a symbol.

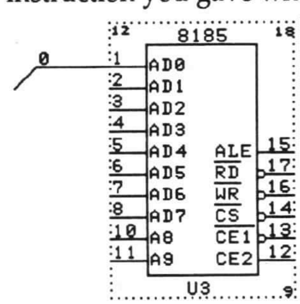
Note: *Each time you enter the /J command, what you specify becomes the new default. If you enter only the /J command, you will get the last type of segment you requested. The first time you use FutureNet, the default is* UL,3,D.

3. Reenter the **.DCON** command to cancel the display of direct connections.
4. Experiment with the different types of junction segments. Before proceeding, delete all of the ones you create.

Create Junction Segments

Now that you have learned how to draw junction segments, you should add them to the 8185.

5. The lines and junction segments you need to add to the 8185 can be created with a command list. The command list must do the following:
   - Draw a 10-display unit segment to the left of the pin stub.
   - Draw a junction segment down and to the left that is 3 display units long.
   - Return to the next pin. Remember, because the cursor doesn't move when you add a junction segment, you only have to reverse the LEFT instruction you gave when drawing the line.



   Experiment with the command list until it performs correctly.

6. Once it is debugged, add the line and junction segment to pins 1 through 8. (The command list is:
   /L;LEFT 10;/L;/L;/J DL,3,D;RIGHT 10;DOWN 3)

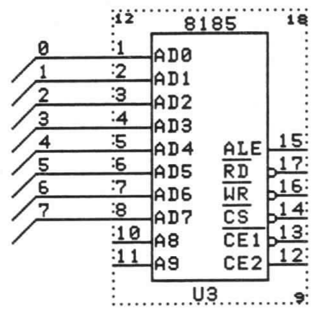
Note: *Because you are drawing lines of specific lengths, position the cursor exactly at the edge of the symbol cell boundary when entering the command list.*

Step 3. Add Junction Segment Signal Names

Add a signal name at the edge of each straight line segment where it attaches to the junction segment.

1. Verify and set, if necessary, the following defaults:
   - Attribute to signal SIG ('A SIG).
   - Font size '2 ('2)
   - Justification left ('J L)
   - Orientation horizontal ('O H)
2. Beginning with a signal name of 0 (zero), write a command list to place signal names on the wire, just before the junction segment. (Hint: **'I 0;DOWN 3)**

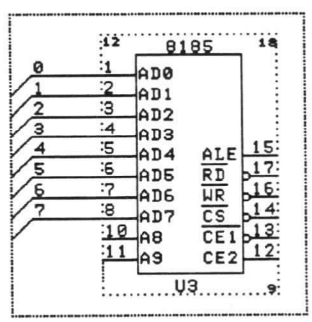
Your completed segments and names should look like this:



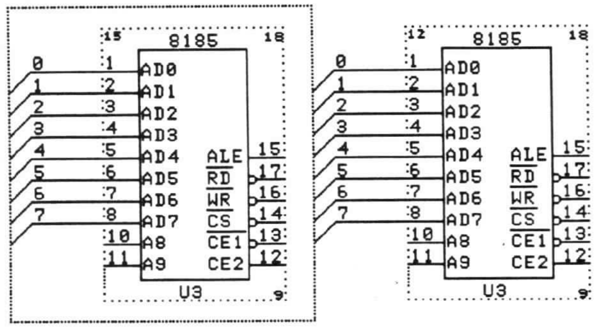
Step 4. Create the Second 8185; Join the Two 8185s

Now that you have one 8185 completed, you can create the second one. This is all possible using the area commands. The secret of this step is to make the left boundary of the area snug with the edges of the junction segments.

1. Using the [D command, define the 8185 and its junction segments as an area. Make sure that the left area boundary line joins the junction segments (in the right position, it seems to overlay the edges of the junction segments). Resize the area if necessary.



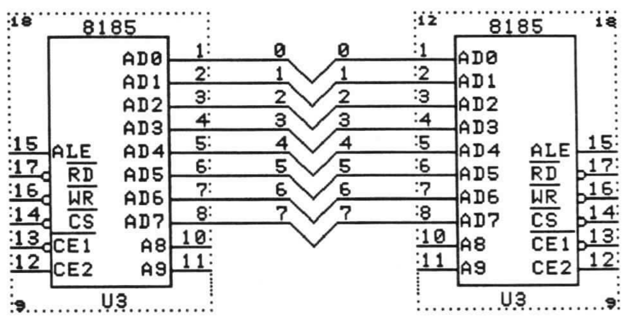
2. Once the area is defined, tag it and copy it. Don't change its vertical positioning; place the ghost immediately to the left of the original 8185, with the ghost boundary touching the original 8185's area boundary. Confirm the copy operation.



3. Using the [RE command, reflect the copied area so you have a mirror image.
4. If your area boundary was correct, the junction segments join perfectly.

   If they do not, resize the area boundary by reentering the [D command with the cursor positioned near a corner on the side of the symbol containing the junction segments. Resize the boundary until it is on top of the junction segments and set it in place again. Then tag the area again, and move it until the right area boundary line touches the right 8185's junction segments.
5. Cancel the area definition by entering the [K command.

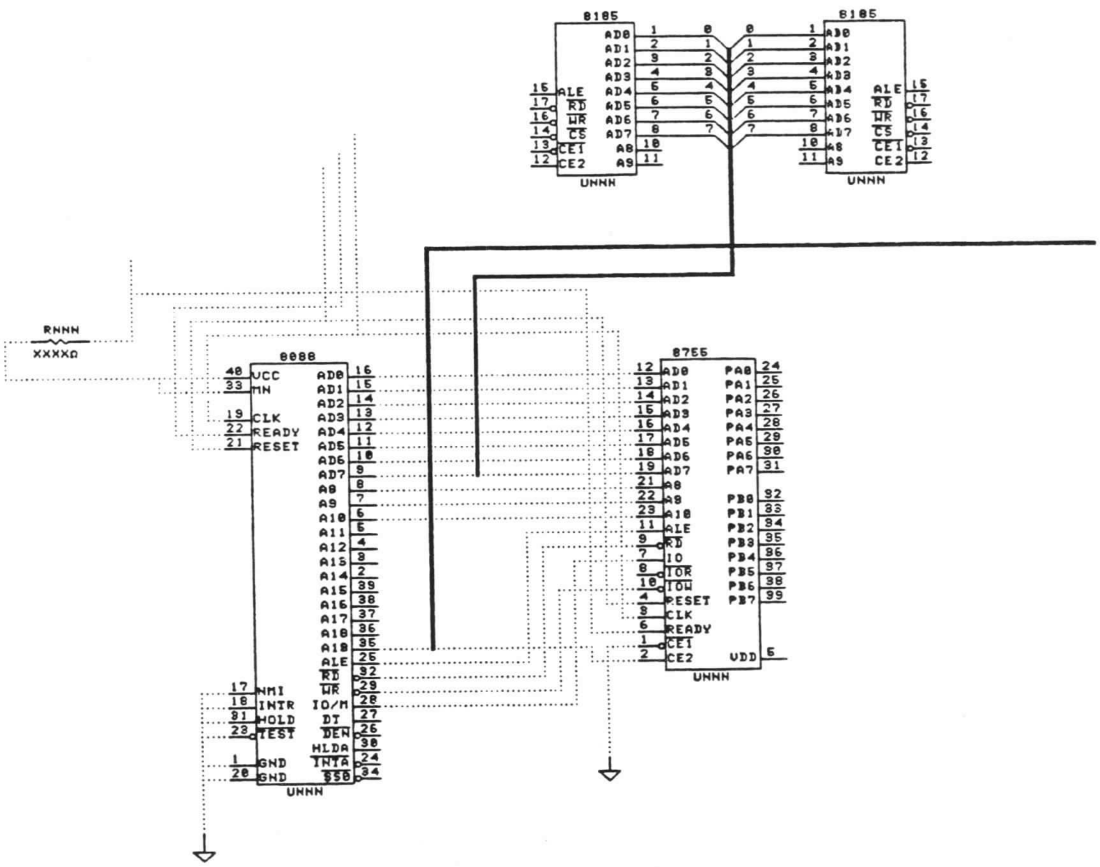
The following illustration shows how the 8185s should look:



Step 5. Create and Name the Buses

The following illustration shows the locations of the buses.

1. Change the line type to /2.
2. The first bus begins where the top two junction segments between the 8185s intersect and turns left 3 display units above the resistor net line. It heads down again when it is about midway between the 8088 and 8755. The bus ends at the line running between 8088 pin 9 and 8755 pin 19.



3. The second bus, as you can see in the illustration, is just two segments. It starts to the right of the right 8185 and moves left, 6 display units above the previous bus. It traverses down the right edge of the 8088, eight display units away from the symbol cell boundary. It ends at 8088, pin 35.

4. To name the buses, verify and change, if necessary, the following:
   - Justification center ('J C)
   - Font size 3 ('3)
   - Attribute BUS 9 ('A BUS or 'A 9)

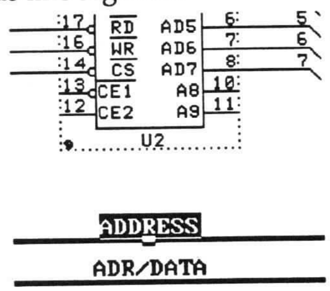
5. Both bus names are centered below the left 8185. Position the graphic cursor on the line before entering alphanumeric mode. The bus names are:

   ADDRESS
   ADR/DATA

Check the Text Position

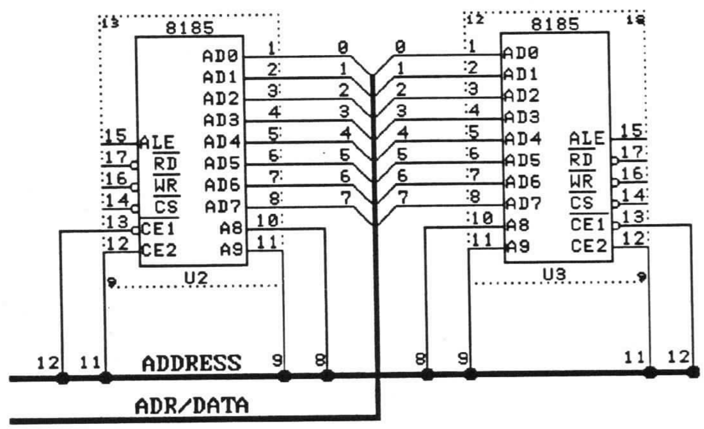
It is crucial that the attribute carried by the bus name be associated with the bus line. What makes this happen is having the point of effect on the line itself.

6. Verify that the text is placed correctly by positioning the graphics cursor in the bus name text field. Check the position of the point of effect. As shown in the illustration below, the point of effect block should be on the bus line segment.



Step 6. Connect to Bus and Name 8185 Signals

The following illustration shows the connections that must be made to the ADDRESS bus.



Create and name the signals:

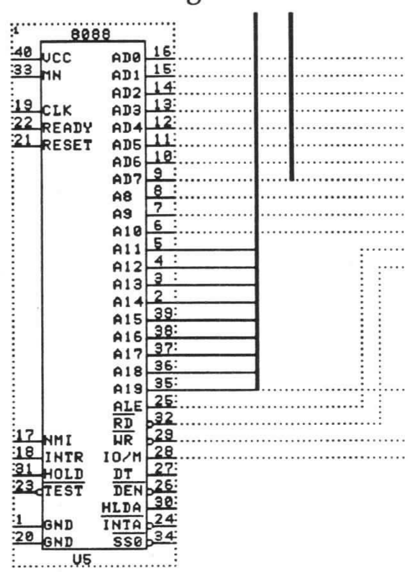
1. Change to line type /1.
2. Referring to the illustration, complete the wires and interconnect dots.
   - The horizontal segments of the lines on pins 10 and 13 on each 8185 should be 9 display units long so they don't crowd the signal names.
   - The horizontal segments on the lines from pins 11 and 12 are 3 display units long.
3. Enter the signal names. Verify, and set if necessary, the following:
   - Attribute SIG 5 ('A SIG or 'A 5)
   - Font type '2 ('2)
   - Right justification ('J R)

Right justification is used because text moves left from the cursor. (Hint: Position the graphics cursor right on the interconnect dot, then move 1 display unit up before entering alphanumeric mode.)

4. If necessary, use the /ES command to shorten the ADDRESS bus, ending it at the right-hand signal 12.

Step 7. Complete Bus Connections

Notice that the 8088 is supposed to completely connect to the ADDRESS bus, but some lines are missing. Referring to the illustration below, complete the lines now using a command list.

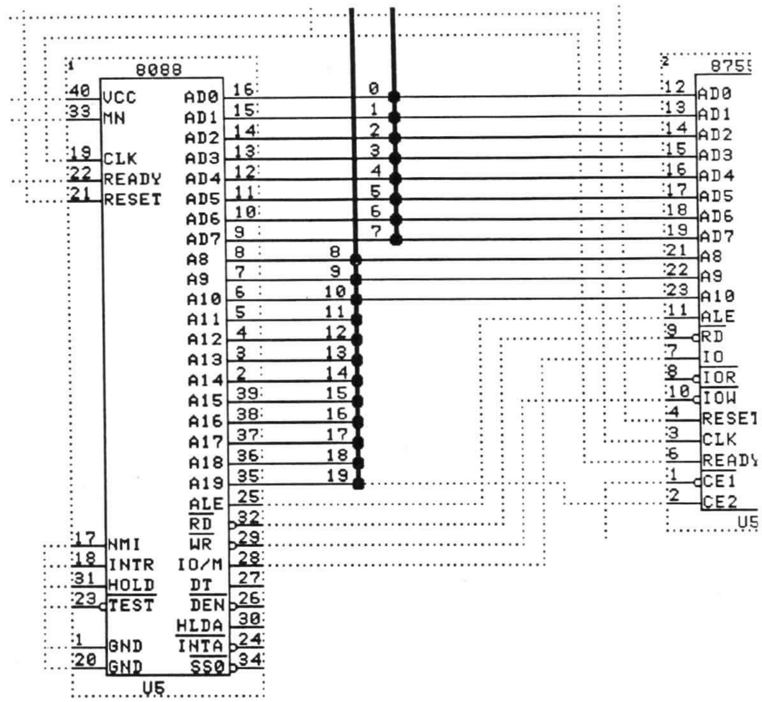


(Hint: measure the distance from the pin to the bus with the cursor—a change in the X direction.)

Step 8. Name Bus Connections

The following illustration shows the information that needs to be added at this point. Basically, it is a signal name and a connect dot. This can be done with a command list.

1. Verify and set, if necessary, the following:
   - Attribute SIG 5 ('A SIG or 'A 5)
   - Font type 2 ('2)
   - Justification right ('J R, so signal names move left from the cursor.)
2. Position the cursor at the intersection of the ADR/DATA bus and the signal running between 8088 pin 16 and 8755 pin 12.



3. Type the following command line (make sure that a zero follows the 'I command):

   /D;LEFT 2;'I 0;DOWN 3;RIGHT 2

   Press ⏎ to name the first 8 signals (0 through 7).

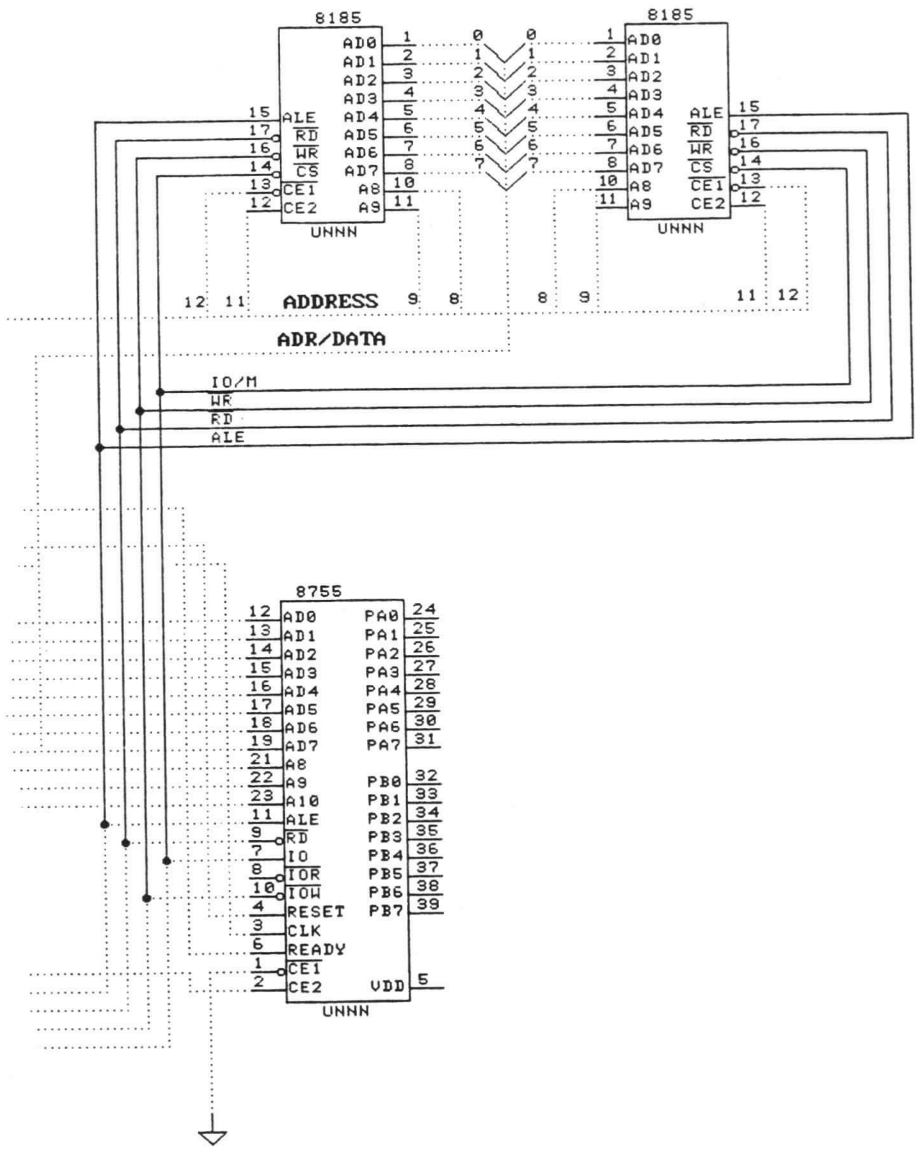
4. When it comes time for signal 8, move the cursor to the intersection of the ADDRESS bus and the signal running between 8088 pin 8 and 8755 pin 21. Add the remaining signals.

Step 9. Open Up Space

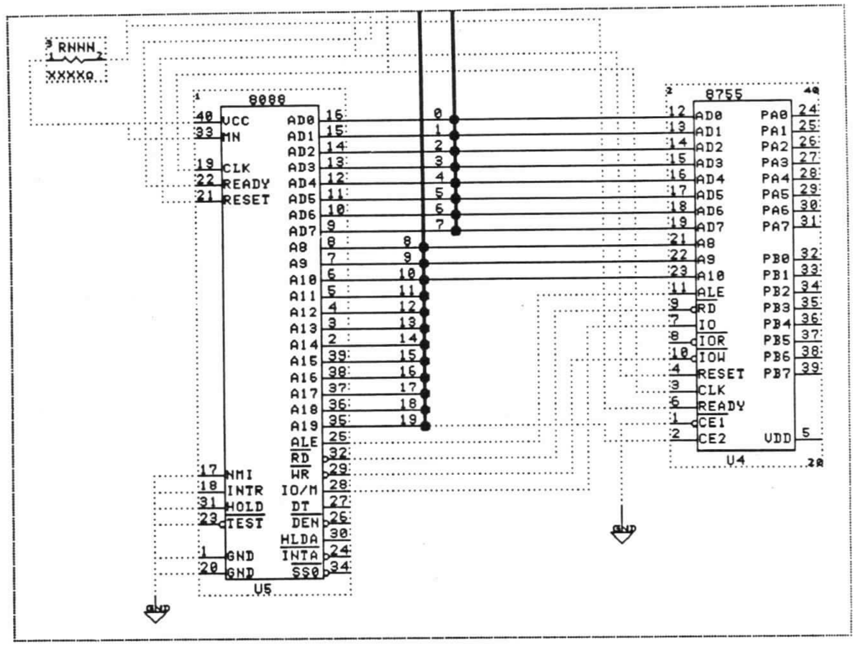
There are still more connections that need to be made. If you started your drawing with the symbols at the recommended locations, however, you do not currently have room for them. The following illustration shows the lines to add.

You can open up space for these connections by creating and moving two areas.

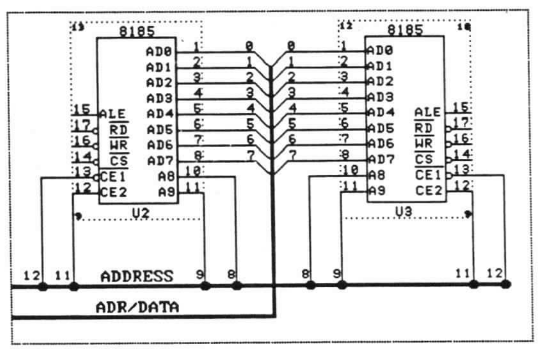
Note: Depending upon what you do, FutureNet may display one or more dotted red lines. These are temporary lines that indicate FutureNet could not adequately reroute the lines. Turn to Session 5 for a discussion on temporary lines, or to the Command Reference *manual, /ET command.*



1. The 8088 and 8755 must be moved down, near the bottom of the drawing. Create an area that encloses the 8088, 8755, the GNDs and the resistor, as shown below.



2. Move the area to near the bottom of the drawing.
3. Create an area that encloses the two 8185s. Make sure that you include the alphanumeric field for signal 12, and make sure that the buses are included, as shown below.



4. Move the area far enough to the right so that the connections can be made. A rough guide for the placement would be to have the left-most 8185 located above the 8755.

Step 10. Make Connections

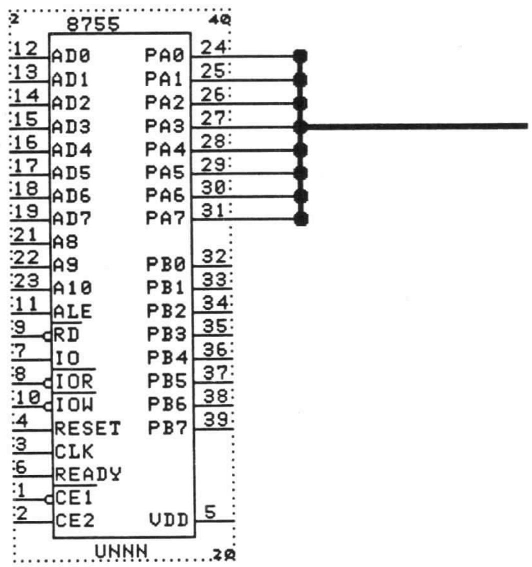
Referring to the illustration at the beginning of the previous step, complete the lines and name them.

- There are several ways to draw the lines. One of the easiest might be to first complete the segments moving down from the left 8185 to the 8755, then make the connections to the right 8185.
- The signal names are attribute SIG 5.
- Note that the signal names WR and RD are inverted.

Step 11. Start 8755 Buses

Connect pins 24 through 31 to a bus. These pins are located on the upper right side of the 8755.

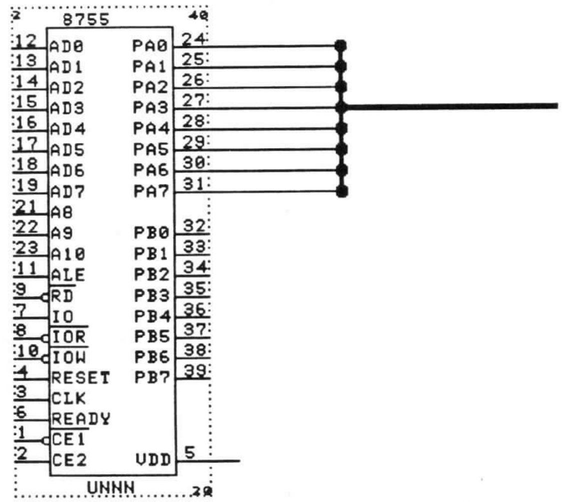
1. Write a command list that draws a line segment to the right 3 display units long, adds a connect dot for connection to the bus, and moves to the next line position. (Hint: /L;RIGHT 3;/L;/L;/D;DOWN 3;LEFT 3.)
2. Change to line type /2 to add the bus.
3. Draw a vertical bus from the segment connected to pin 24 to the segment connected to pin 31. Add the connect dots if needed.
4. Draw the horizontal bus segment from pin 27 to the right 30 display units. Your completed bus looks like this:



Stretch Connections

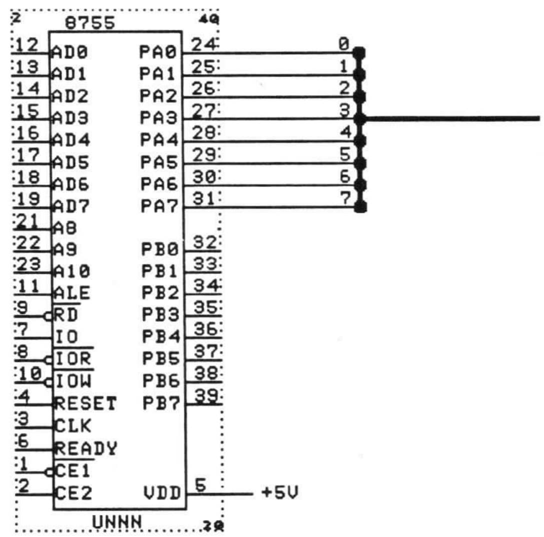
Now that the bus is connected to the 8755, you need to lengthen the connection between them to open up room for signal names.

5. The segments you just drew need to be longer so you can fit signal name fields on them. Position the cursor right at the symbol cell edge above pin 24 and define an area that encloses all the segments, as well as the bus.
6. Tag the area and MOVE it to the right 10 display units. Notice how the lines are stretched. This is called rubberbanding. Set the area in place. It now looks like this:



Step 12. Complete 8755 Buses

You can now complete the first bus by adding the signal names.



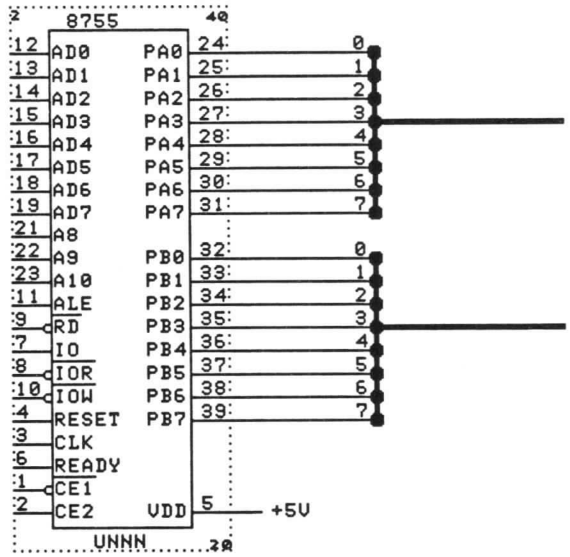
1. Verify and set, if necessary, the following
   - Font type '2
   - Justification right
   - Attribute SIG 5
2. Either manually or with a command list, add signal names as shown above. (Hint: position the cursor 1 display unit to the right of the dot.)

Duplicate the Bus

With pins 24–31 connected to the bus, you need to connect pins 32–39 to another bus. You could create a new set of connections and signal names, or you can duplicate the existing bus.

3. You can now duplicate the bus and attach to it pins 32–39. Define an area. Begin at the symbol cell boundary and include all of the signals, text and bus lines.
4. Position your cursor precisely where signal 0 and pin 24 meet. COPY the area. Notice that the point of the tag cursor is in the same location that the graphic cursor was. You now have a clear reference point to use.
5. To place the copy accurately, position the tag cursor right at the edge of pin 32. Confirm the copy.

The completed 8755 buses appear below.



Step 13. About Points of Effect

In Session 2, when we introduced attributes, we glossed over points of effect. This was because at that time, the drawing didn't have information in it that could adequately demonstrate points of effect. Now it does.

Point of Effect Commands

The point of effect commands assist you in manipulating the point of effect. The commands are:

MOVEPOEE	Enable/disable point of effect movement for electrical attributes
MOVEPOEG	Enable/disable point of effect movement for general attributes
POER	Point of effect range enable/disable
'P	Set point of effect
'PD	Return point of effect to default position.

FutureNet Defaults

The first three commands listed above have default settings. These default settings will not need to be changed except under the most unusual circumstances (which will not be discussed here). This defaults are listed in the Profile screen (enter **PROFILE**).

☐ Move Elec POE

Default is ON. **Move Elec POE** determines whether the point of effect for electrical attributes moves when the alphanumeric text moves. Electrical attributes are all those relating to signal and pin names. In most all circumstances, you will want this option ON, meaning that if you move a piece of text, you'll probably want the point of effect to move with it. For example, if you give the wrong name to a pin or signal, and move it to the correct one, you'll want the point of effect to travel with the alphanumeric text. This option can be changed on the command line with the command **MOVEPOEE**.

☐ Snap Sig POE

Default is ON. **Snap Sig POE** (Point Of Effect Range) influences how the point of effect behaves for **signal-related attributes** only. When on, the point of effect for alphanumeric text with a signal-related attribute snaps to the closest line within a 10-display unit distance. This allows you a little greater freedom in placing signal and bus names. This option can be changed on the command line with the command **POER**.

When MOVEPOEE and POER are ON, you can move signal names to other signal or bus lines, and the point of effect will attach to the nearest bus or signal line within 10 display units. If there is nothing within the 10-display unit limit, the point of effect moves to its default position within the alphanumeric field of the signal or bus name, and will not be attached to a signal or bus line.

When POER is off, the alphanumeric boundary of a signal name must actually touch the line it names in order for the point of effect to be placed correctly.

To see how this works, do the following:

1. Draw a single-segment line, of either type /1 or /2.
2. Position the cursor 5 display units above the line, near the left end.
3. Enter the 'A 0 command to set the default attribute to COM.
4. Enter alphanumeric mode. Type **COM**. Exit alphanumeric mode. Notice that the point of effect is with the text.
5. Change the default attribute to SIG ('A SIG). Move the cursor to the right, but do not change the vertical position.
6. Enter alphanumeric mode. Type **SIG**. Exit alphanumeric mode. Notice that the point of effect is on the line.

☐ Move Gen POE

Default is ON. **Move Gen POE** determines whether the point of effect for general attributes moves when the alphanumeric text moves. General attributes are all those relating to symbols, such as LOC 2 or PART 3, and the drawing, such as TITL 50 (the attribute of the title in the title block). This options can be changed on the command line with the command **MOVEPOEG**.

In almost all circumstances, you will want this left on because, if you move a piece of text, you'll want the point of effect to move with it.

To summarize, FutureNet, operating with these defaults, behaves in this manner:

- If you move any alphanumeric field, the point of effect moves with it.
- When entering new text, position the cursor to be as close as possible to the item being labeled.
- When moving alphanumeric fields, make sure the alphanumeric boundary actually touches the object. The exception is:
  - Alphanumeric text with signal-related attributes may be placed up to 10 display units away from the target. The point of effect snaps to the closest item within 10 display units.

Step 14. Manipulating Points of Effect

The remaining two point of effect commands, 'P and 'PD, allow you to manipulate the point of effect once it is in place. As with changing the defaults mentioned above, generally these commands are only used in unusual situations.

Using 'P

The 'P command is used when you want to actually move the point of effect itself. Do the following:

1. Position the cursor on the alphanumeric field **COM** in the example you created in the last step.
2. Enter the 'P command. The point of effect changes to a diamond with opposite sectors filled. This is the point of effect cursor.
3. Move the mouse. Notice that the point of effect cursor moves as you do. Position the center of the cursor on the line.
4. Reenter the 'P command. The cursor disappears and the point of effect is positioned on the line.

Using 'PD

The 'PD command causes the point of effect to return to its default location. (Refer to the *FutureNet Command Reference* for details on defaults.)

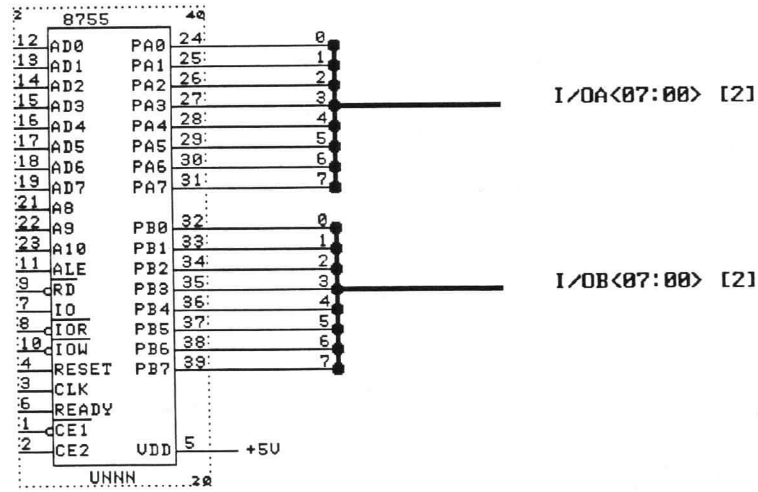
5. Position the cursor in the **COM** field.
6. Enter the 'PD command. The point of effect leaves the line and rejoins the **COM** alphanumeric field in its default location.

Note: With MOVEPOEE and POER on, the default location for the point of effect for text with a signal attribute will be on the closest line or bus within 10 display units.

7. Before leaving this step, delete all of the experimental lines and fields you have created.

Step 15. Name the Buses

Referring to the illustration below, add the bus names:



1. Set and verify, if necessary, the following:
   - Font type '3
   - Justification left
   - Attribute BUS 9
2. Position the cursor near the right end of the bus. Enter alphanumeric mode and type the first bus name (in "I/OA", the "O" is alphabetic):

   I/OA<07:00> [2]

   Exit alphanumeric mode.
3. Move to the end of the next bus and add the other bus name:

   I/OB<07:00> [2]

Looking Ahead

In the next session, you learn about the following:

- Temporary lines
- The tasks in completing a drawing
- Connecting to off-page power sources

If You Are Continuing

If you are continuing immediately with the next session, enter the SAVE command to save your drawing at this point, then continue.

If You Are Stopping Here

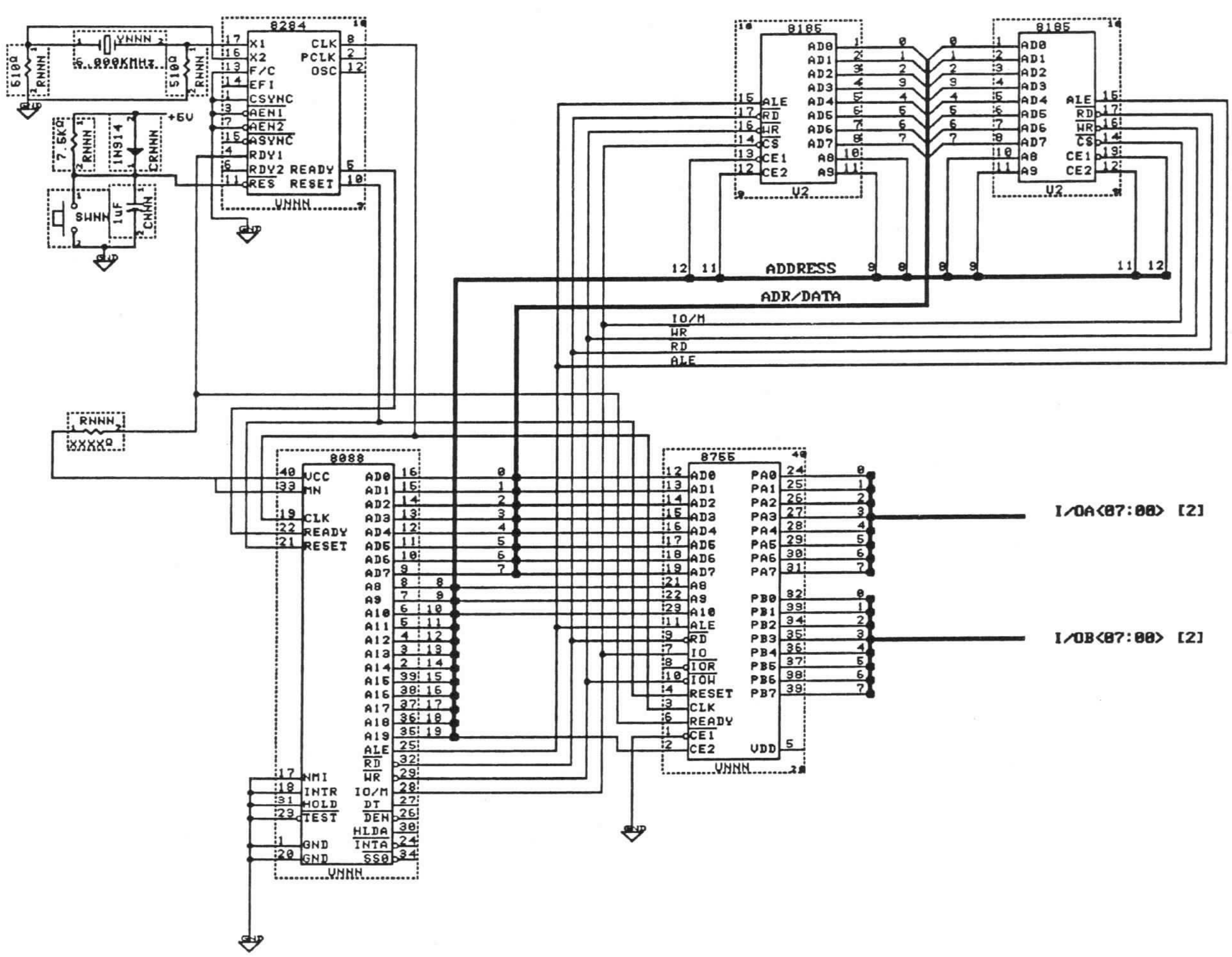
If you are stopping here and plan on exiting FutureNet, enter the SAVE command, then the QUIT command.

Summary

You have now completed Session 4. In summary, here are the commands and functions you learned:

Command	**Function**
MOVEPOEE	Enable/disable point of effect movement for electrical attributes
MOVEPOEG	Enable/disable point of effect movement for general attributes
POER	Point of effect range enable/disable
'P	Set point of effect
'PD	Return point of effect to default position

session4.dwg



11 Session 5: Completing the Drawing

The last four sessions have focused on tasks related by system function, such as symbols, alphanumerics, areas and lines. This session covers the more diverse activities related to actually finishing a drawing, such as:

- Changing LOC data
- Adding the title block
- Connecting to off-page power sources
- Temporary lines that result from symbol or area relocations

Step 1. Before You Begin

Start FutureNet.

Do one of the following:

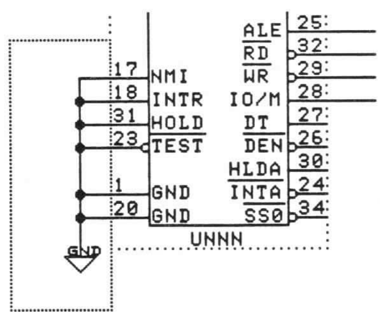
- If you do not want to use your own drawings, or if you have not completed Session 4:
  - LOAD the Data I/O-supplied drawing **session5**.
- If you are beginning the tutorial sessions again and have completed Session 4, do the three steps below. If you are continuing from the previous session, do the second and third steps:
  - LOAD the drawing **session4**.
  - SAVE the drawing on your screen as **gs5**. This creates a new drawing but does not bring it to the screen.
  - LOAD the drawing **gs5**.

Step 2. About Temporary Lines

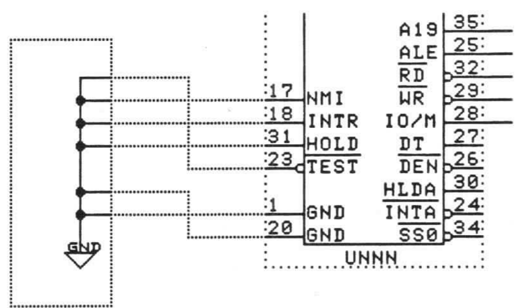
When a symbol, area, or vertex is moved, FutureNet attempts to maintain connections by rubberbanding the lines. Each line connecting to the object is traced back, up to two line segments, until a connection point, a connect dot, or a symbol is encountered. It is at this point that rubberbanding begins. FutureNet attempts to draw permanent lines between this point and the object. If this isn't possible, or if more than two segments are needed to maintain the connections, temporary lines, displayed as dotted lines, maintain the connections.

Note: SAVE your drawing before beginning this step. You do not want to keep the changes made during this exercise.

1. Referring to the illustration below, create an area that completely encloses the GND connection of the 8088. The right edge of the area should overlap the symbol cell boundary.



2. Move the area 20 display units to the left and up 3. Set it in place. Your screen should now look like this:



FutureNet maintains the connections, but with a different type of line. The dotted lines are temporary lines, because in order to maintain the connections, FutureNet had to replace the original single-segment lines with lines of 3 segments.

3. Move your cursor on top of one of the lines. Notice that it highlights and/or changes to a less-dense dot structure. This allows you to trace the connection back to its original point.

Making Temporary Lines Permanent

These temporary connections are true lines. If you were to run a pinlist on drawing at this point, they would perform as if they were permanent lines. But temporary lines should be converted to permanent.

4. The /P command converts a temporary line to a permanent line. Position the cursor on the line from pin 17 and enter the /P command. You are prompted:

   ```
   New connection would be created due to overlap or T
   intersec  Continue (Y/N)?
   ```

   You are being warned that the line you are about to make permanent may actually connect with the line below on its vertical segment. Answer Y. This creates a condition that you will correct in a few steps.

5. Make the line below it connecting to pin 18, permanent also.

6. Tag the area and move it down one display unit.

   You can now see the connections you were warned about in point 4. The connections were created because the vertical segments of the temporary lines touched each other, creating a connection which was preserved when the lines were converted to permanent. Be aware of this when converting temporary lines to permanent.

Rerouting/Removing Temporary Lines

While you are able to convert temporary lines to permanent when the routing of the segments is clear, many times you'll reroute the connections with permanent lines and delete the temporary ones, ignoring the conversion process.

7. Now that the vertical segments are no longer touching, do the following:

   - Overdraw the temporary line connecting to pin 31. Notice that the temporary line remains under it.

   - Overdraw the temporary line connecting pin 23, but choose a different location for the vertical segment. Notice, again, that the temporary line remains.

8. Remove completely the temporary lines by entering the /ET command. You are prompted:

   ```
   OK to erase all temporary lines (Y/N)?
   ```

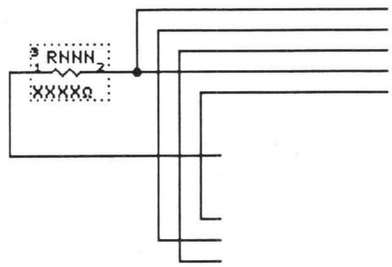
   Answer Y.

   The temporary lines disappear. You would now have to reroute the connections to pins 1 and 20.

9. Abandon these changes by reLOADing the drawing. (Answer Y when FutureNet asks if you want to discard the changes to the drawing.)

Step 3. Edit the Resistor's Value

Notice that the resistor's value is marked with XXXX as a place holder. It must be changed to a valid 510.



Note: Because you are editing an existing field, you do not need change the text attributes of the field if they are acceptable. New text added to an existing field assumes the characteristics of the current text.

1. Position the cursor in the value's field. Notice that the attribute is VAL 4.
2. Enter alphanumeric mode. Edit the field to be 510. Exit alphanumeric mode.

Step 4. Edit LOC Fields

You must now edit the LOC fields, giving the proper number to each symbol. As above, because you are editing an existing field, you do not need to alter the defaults if they are acceptable.

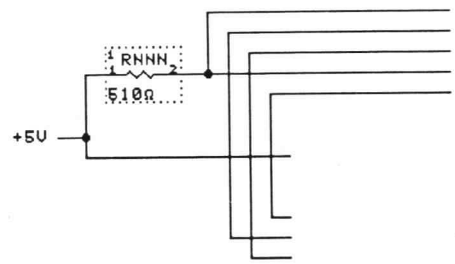
1. Position the cursor over the LOC field on the right 8185.
2. Enter alphanumeric mode. Change the contents of the LOC field from U2 to U3. Exit alphanumeric mode.
3. Use the same technique to change the LOC fields as follows:

8755	U4
8088	U5
Resistor	R4

Step 5. Connect to an Off-page Power Source

Refer to the following illustration to see how to connect to an off-page power source.

1. Verify and set, if necessary, the following:
   - Default attribute to SIGU 14 ('A SIGU)
   - Default justification to right ('J R)
   - Default font size to '2 ('2)
   - Default line type to /1 (/1)



2. This first connection is between pin 1 of the resistor on 8088 pins 33 and 40 and a power source. Referring to the illustration above, position the cursor on the connection and draw a 4-display unit line. Include a connect dot.

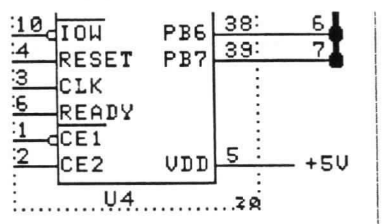
3. Move the cursor 1 display unit left and 1 display unit down from the end of the line. Enter alphanumeric mode. Type the connection name:

   +5V

   Exit alphanumeric mode. Since the SIGU attribute was used with the +5V text field, the Post processors know that these signals are connected to the same source as the implicit power pins of the symbols.

4. Using the following illustration as a guide, complete the connection for the 8755 in a manner similar to that used for the resistor.

Note: You may want to use left justification for this alphanumeric field.



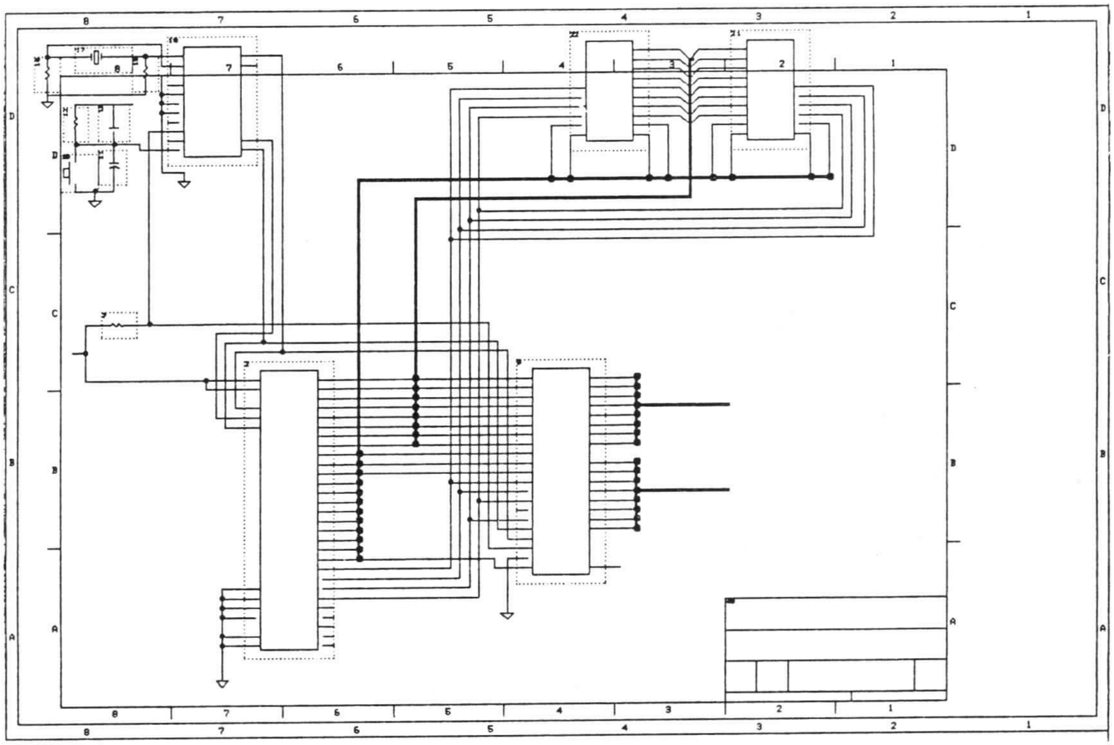
Step 6. Load Title Block

Usually you would load and fill in the title block when you start the drawing, and then arrange the other symbols within it. The title block is being loaded last so that you could make full use of the drawing area on your screen. Follow these steps to load the title block.

1. Switch to reduced zoom level by entering the command ZIN 2 (Zoom-In at level 2).

2. Position the cursor at specific drawing reference coordinates by entering B2. This puts you near the location for the title block, but it will probably not load at that point.

3. Load the symbol TITL ready for moving by entering the * TITL command. If the title block does not load the first time, move the cursor to a new location, and reenter the command by pressing the middle mouse button until it does. The blank title block appears at the cursor location. Your screen will look similar to this:



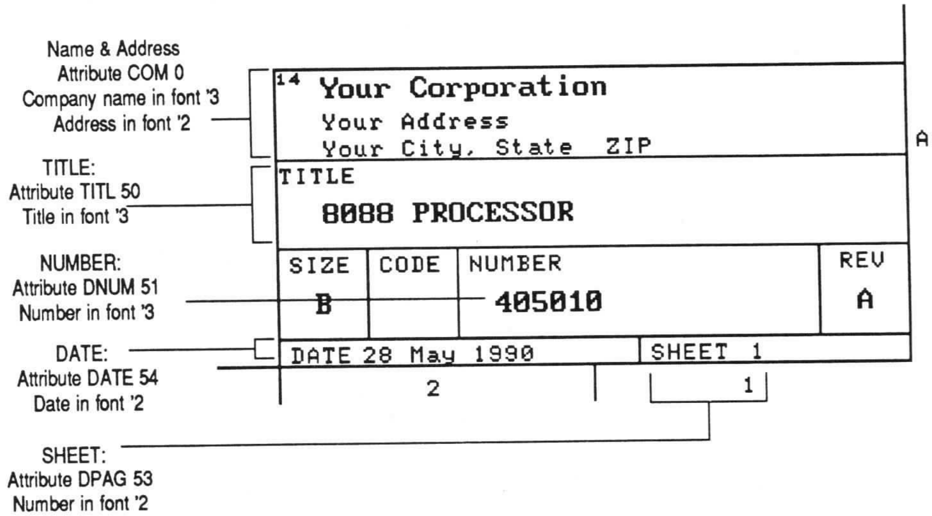
4. Move the ghost around, seeing if the drawing can fit inside the frame. Notice that as you move the ghost up and to the left, it begins to shrink once the top and left sides touch the edge of the workspace. You can enlarge it again by pulling down and to the right until it reaches full size. You can tell when the frame is full size because the lines for the drawing reference coordinates appear (the coordinates themselves do not appear until the title block is set in place, however).
5. If your drawing can fit inside the frame, set the frame in place by confirming the move operation.
6. If your drawing does not fit, press the right mouse button to cancel the load-and-move operation of the title block.
7. You'll need to tighten up spaces in the drawing. For instance:
   - Define an area that includes the original CLKRES.ARA. You can then move it straight down, closer to the 8088. Avoid creating temporary lines, if you can help it.
   - The same can be done for the 8185s.
   - You may need to shorten the distance between the 8088 and its GND by moving the GND up closer to the 8088.

- An overall adjustment of the space in the drawing is a good idea. The clock area was loaded flush with the top and left edges of the drawing. You may have moved it down, but not to the right, so it crowds the left edge of the drawing. Notice that there is also an abundance of space on the right side of the drawing. You can recenter your shrunken drawing in the drawing area by defining the entire drawing as an area, and then moving it down and to the right.

8. Once you have tidied up the spaces, load the title block and set it in place.

Step 7. Fill In the Title Block

The following illustration shows the details on filling in the title block.



1. The top blank space contains the company name and address. Its attribute is COM 0. The company name (first line) is usually entered in font '3, the address lines in '2. To position the text correctly, position the cursor on the upper left corner and then cursor down 4 display units and right 5 display units.

2. For the TITLE field, set the attribute to TITL 50, and enter the text in font '3. To position the text correctly, place the cursor on the upper-left corner of the field, (right above the first "T" in TITLE), and then cursor down 7 display units and right 5 display units. The title is **8088 Processor.**

3. For the NUMBER field, set the attribute to DNUM 51, and enter the text in font '3. To position the text correctly, position the cursor first in the upper-left corner of the field, then cursor down 7 display units and right 5 display units. The number is **405010.**

4. For the DATE field, set the attribute to DATE 54 and enter the text in font '2. Position the cursor flush on the line under DATE and to DATE's right, 2 or 3 display units. Type **today's date.**

5. For the SHEET field, set the attribute to DPAG 53 and enter the text in font '2. Position the cursor flush on the line under DATE and to SHEETS's right, 2 or 3 display units. Type **1.**

Looking Ahead

The general sessions end here. The next few sessions detail working with structured designs, and using FutureNet symbol definition mode. Continue with them if your work requires that you understand those concepts.

If You Are Continuing

If you are continuing immediately with the next session, enter the SAVE command to save your drawing at this point, then continue.

If You Are Stopping Here

If you are stopping here and plan on exiting FutureNet, enter the SAVE command, then the QUIT command.

1. Perform clean-up on your directory. Remove any unnecessary **.dwg** (drawing), **.ara** (area) and **.sym** (library) files.

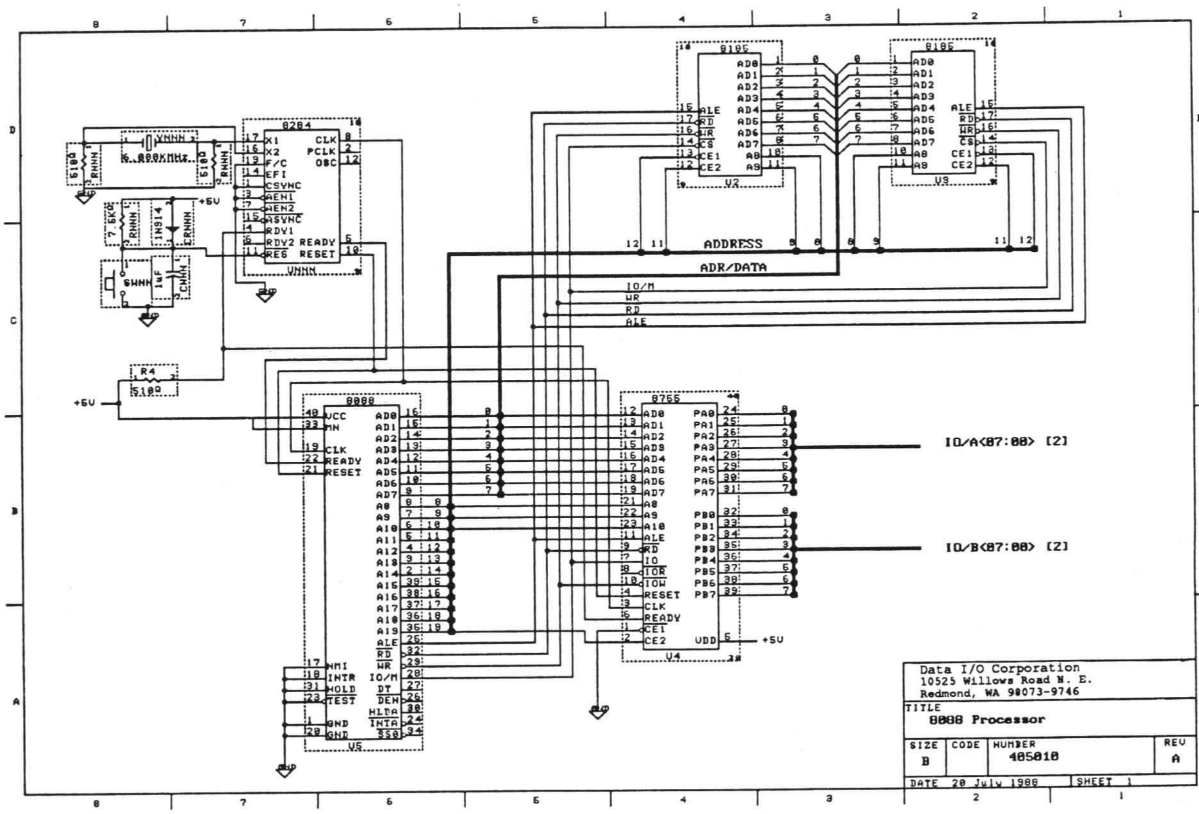
2. With the drawing completed, you could run any of the various Data I/O Post processors through the **EXPORT Generate Reports** menu. The Post processors include the Drawing Preprocessor, Design Rule Check, Pinlist Generator, Netlist Generator, and Parts List Generator.

Summary

In summary, here are the commands and functions you learned:

Command	Function
/P	Convert a temporary line to permanent
/ET	Erase all temporary lines within a drawing

session5.dwg



12 *Session 6: Building Functional Blocks*

This session introduces these structured design concepts:

- Structured designing
- Creating functional blocks
- Connecting signal and bus lines to functional blocks
- Assigning pin names without pin stubs
- Assigning filename pointers

Step 1. Before You Begin

Start FutureNet. Or, if you are already running FutureNet, CLEAR any drawing from your workspace.

Step 2. About Functional Block Symbols

A functional block is a "black box" that represents a portion of the circuitry in a design. It forms a part of structured design, because the functional block represents items that are part of the design, but which are detailed on another sheet.

The advantage of structured design is that when a design calls for multiple appearances of a group of symbols, these symbols do not need to be physically present on every sheet in which they are needed. A functional block, substituting for the real items, can take their place. All Data I/O post processors, such as Pin List Generator and Parts List Generator, understand structured designs.

Functional blocks may have a number of distinctive characteristics to differentiate them from standard symbols. These are discussed below.

Functional Block Attributes

To be considered a functional block, there is only one thing any symbol must have: an alphanumeric field with an attribute of FILE 8 (printing) or FILN 114 (non-printing). An alphanumeric field with one of these attributes points to a drawing file of the same name, giving the location of the drawings which contain the circuitry that the functional block represents. Any number of filenames may be referenced within a functional block.

The PART and LOC Fields

As with all symbols, a functional block must have a PART and LOC field. The contents of these fields depends upon the ultimate function of the functional block. If the block is entirely logical, meaning that it is merely a pointer to other circuitry, the PART and LOC fields may contain any entries that are significant to you.

If, on the other hand, the functional block represents circuitry that will be entered in ASICs or in programmable devices, the PART and LOC fields would probably contain more conventional information that indicates the function of the part and its reference designator.

In short, the PART and LOC fields may be either "logical," meaning they give information, or "physical," meaning they refer to a physical part.

Functional Block Form

The physical form of a functional block is not restricted in any way: it can look like anything you like. This session introduces one way of creating functional blocks that makes them distinctive.

Creating the Block

The functional block symbol may be created with either the .B or .F commands, or in symbol definition mode. Using the .F command creates some distinction because of its double line symbol cell.

Pins and Functional Blocks

Creating a functional block is easier than creating a standard symbol, because you don't need pin stubs nor pin numbers.

Remember that the attribute information that defines a pin is carried by the pin number, and not the pin name or pin stub. Pin stubs, residing between the symbol cell and its boundary, are a standardized way of showing what must be connected on a symbol. Pin stubs can be excluded from functional blocks because a functional block is created for a specific purpose, and often doesn't have real pins.

Step 3. About this Functional Block

The **scontrol** functional block constructed in this chapter is used in the next chapter, which contains the complete detail of the automobile cruise control circuit. Turn there, if you wish, to see the inner workings of the circuit. The **scontrol** functional block needs the following:

- Inputs: a reference speed, an input for drive shaft revolutions, and an interval clock signal.
- Outputs: a signal that tells the car to speed up, and one that tells the car to slow down.

Step 4. Build a Functional Block

To understand how a functional block is built, do the following:

Move the graphics cursor near the center of the screen. Enter:

.F 20,30,0,3

This specifies the following:

.F Create a functional block.

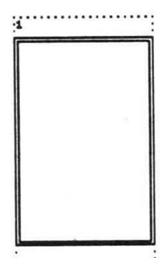
20, Make the rectangle 20 display units wide.

30, Make the rectangle 30 display units high.

0, Make the symbol cell boundary 0 display units in width on the left and right edges of the block symbol.

3 Make the symbol cell boundary 3 display units in width on the top and bottom of the block symbol.

Your functional block symbol looks like this:

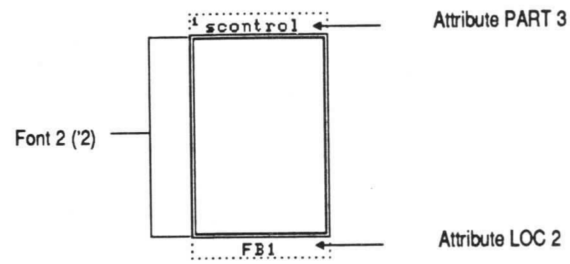


Note the following:

- Because this functional block does not use pin stubs, you can have the boundary overlap the symbol cell on the right and left edges. This gives the functional block an additional distinguishing characteristic.
- Because we still need space for PART and LOC fields, that space is provided at the top and bottom.

Step 5. Assign the PART and LOC Fields

Once created, the functional block must have the appropriate text fields. The following illustration shows the location of the fields and the characteristics and attributes of the alphanumeric fields.



Assign the PART name

The next step in creating a symbol is assigning a PART name to the symbol.

1. Verify, and set if necessary, the following:
   - Font size '2 ('2)
   - Attribute PART ('A PART)
   - Justification center ('J C)
2. Center the cursor between the right and left edges of the symbol, right on the solid symbol cell at the top of the functional block. Enter alphanumeric mode and type the symbol name:

   scontrol

   Exit alphanumeric mode.

Assign the LOC Name

After you have assigned a PART name to the symbol, you should assign a locater to the symbol.

3. Enter the 'A LOC command
4. Center the cursor between the right and left edges of the symbol, right on the dotted symbol cell boundary at the bottom of the functional block. Enter alphanumeric mode and type the locater:

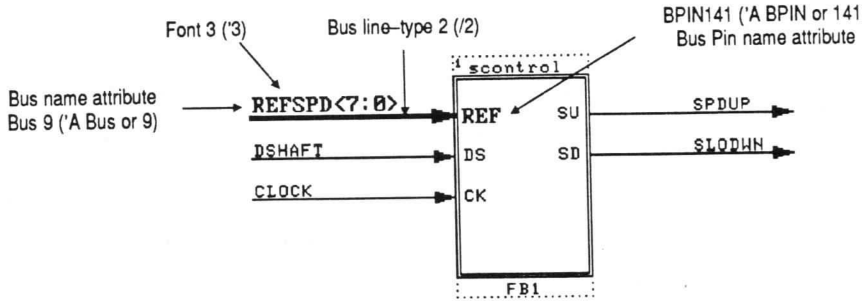
   FB1

   Exit alphanumeric mode.

This symbol now represents the speed controller device.

Step 6. Create the Bus Line and Pin Name

It's now time to add the signal lines and pin names to the functional block. Because the bus line type and alphanumeric font size differ from the others needed on the symbol, we'll just add the bus in this step. The complete bus and pin names are shown below.



Note: As you have learned, lines want to connect to the points of effect for pins. This is important for connectivity, and when pin stubs are present, causes no cosmetic problem. In functional blocks without pin stubs, however, the effect is not desirable. For this reason, it's necessary to add the signal line first, then the text that serves as the pin number.

Make sure that you have the correct attribute at all times!

1. Verify, and set if necessary, the following:
   - Font size '3 ('3)
   - Line type /2 (/2)
   - Attribute Bus ('A Bus or 9)
   - Justification left ('J L)
2. Position the cursor on the symbol cell, 6 display units down from the upper-left outside corner of the symbol cell.
3. Draw the bus line from this point 30 display units to the left.
4. Without moving the cursor, enter alphanumeric mode and type:

   REFSPD <7:0>

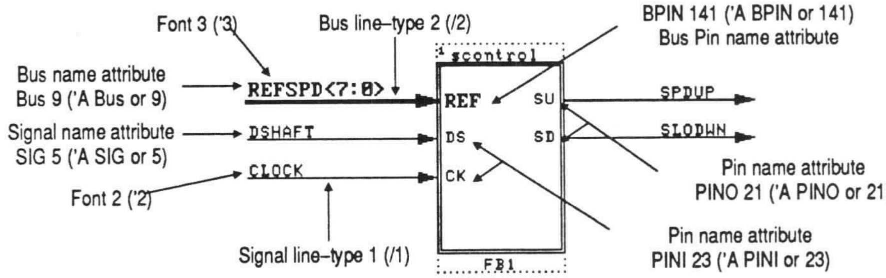
   Exit alphanumeric mode.
5. Return the cursor to where the line intersects the symbol cell. Enter the /AR command to add the arrow symbol.
6. Position the cursor at the intersection of the line with the symbol cell, then move 2 display units down and 1 to the right.
7. Enter the 'A BPIN command to set the new default attribute.
8. Enter alphanumeric mode and type the pin name:

   REF

   Exit alphanumeric mode.
9. Verify your work against the symbol at the beginning of this step.

Step 7. Add the Remaining Lines and Pins

The remaining lines and pins share their type size and line size, so you'll add them now. The illustration below shows what needs to be added.



1. Verify, and set if necessary, the following:
   - Line type /1 (/1)
   - Font size '2 ('2)
   - Attribute SIG 5 ('A SIG *or* 5)

   When adding the signal lines, names and pins, keep the following in mind:
   - The lines are 30 display units long.
   - The left-pointing arrows are added with the /AL command.
   - To correct placement of an arrow, delete it by reentering the command which placed it (toggle it, in other words), and add a new one at the correct location.
2. Add the signal lines and names first:

   DSHAFT Line begins 6 display units below the REFSPD line.

   CLOCK Line begins 6 display units below the DSHAFT line.

   SPDUP Line begins 6 display units below the upper-right outside corner of the symbol cell. Enter the signal name in left justification about 15 display units to the right of the symbol cell.

   SLODWN Line begins 6 display units below the SPDUP line. Enter the signal name in left justification about 15 display units to the right of the symbol cell.
3. Enter the 'A PINI command to change the default attribute for the pins.
4. Add the pins:

   DS Use left justification. Position the cursor 1 display unit down and 1 right of the intersection of the line and the symbol cell.

   CK Use left justification. Position the cursor 1 display unit down and 1 right of the intersection of the line and the symbol cell.

5. Enter the 'A PINO command to change the attribute for the pins.
6. Add the pins:

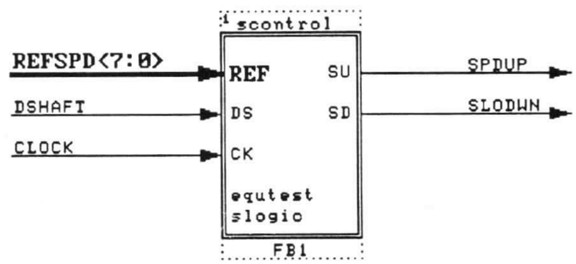
SU	Use right justification. Position the cursor 1 display unit down and 1 right of the intersection of the line and the symbol cell.
SD	Use right justification. Position the cursor 1 display unit down and 1 right of the intersection of the line and the symbol cell.

Step 8. Add Filename Pointers

The internal makeup of the speed controller consists of an equality checker to test the revolutions of the drive shaft against the reference speed, and logic circuitry to determine and control speed.

This suggests that the next node of the design tree should be a drawing set that contains two drawing files: **equtest.dwg** and **slogic.dwg**.

It is necessary to establish a link between the root drawing and the files that will become the detail drawings at the next lower level. This is done by inserting filename pointers in the functional block.



1. Verify, and set if necessary, the following:
   - Default attribute to FILE 8 ('A FILE or 8). This is a printing attribute; FILN 114 is non-printing.
   - Justification left ('J L)
   - Font 3 ('3)
2. Position the cursor on the left edge of the inner rectangle of the functional block, 7 display units below the pin name field, CK.
3. In alphanumeric mode, type the first filename:

   equtest
4. Press ⏎ to move to the next line. Type the next filename:

   slogic

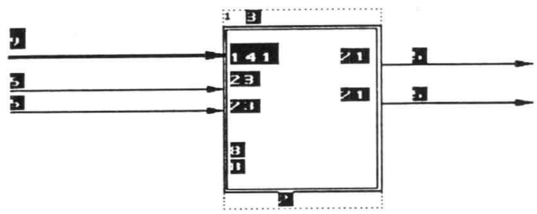
   Exit alphanumeric mode.

You have now assigned filename pointers to the files **equtest.dwg** and **slogic.dwg**, which FutureNet will expect to find in the current drive and directory.

Step 9. Verify Attribute Assignment

It is important that all text fields in the functional block have the right attributes assigned. Check to make sure that they are correct.

1. Enter the 'D command. The attributes for every alphanumeric field are displayed. Compare them against the illustration below.



2. If any are incorrect, use the command

   'CH A *attribute*

   to change them.

Step 10. Save the Drawing and Quit FutureNet

The drawing created in this session is provided for you (with title block and border added) under the filename **scontrol.dwg**.

1. If you wish to save this drawing, use the SAVE command to do so, but be sure to use a filename other than **scontrol** so you won't overwrite the drawing file provided by Data I/O.
2. Quit FutureNet if you will not be continuing with the next session.

13 *Session 7: Advanced Structured Design*

This session discusses the remainder of the design tree for the automobile speed controller, and introduces these concepts:

- Loading a drawing file
- Moving between drawing levels in the design tree
- Associating signal nets between drawing levels
- Moving between sheets of a drawing set
- Displaying all accessed files
- Saving all drawings in the design tree
- Clearing the workspace

You will use the sample **.dwg** files provided by Data I/O.

Step 1. Before You Begin

1. Start FutureNet, if necessary.
2. LOAD the drawing **scontrol** if you have not completed the previous session. If you have completed it, load the drawing that contains the functional block you created there.

   If you are using your own version of **scontrol**, substitute that name for **scontrol** when you perform the steps in this chapter.

Step 2. Move Down in the Design Tree

Recall that **scontrol** contains filename pointers to the drawing files **equtest** and **slogic**. Assume that the first drawing you want to create is **equtest**. To create a workspace for this file do the following:

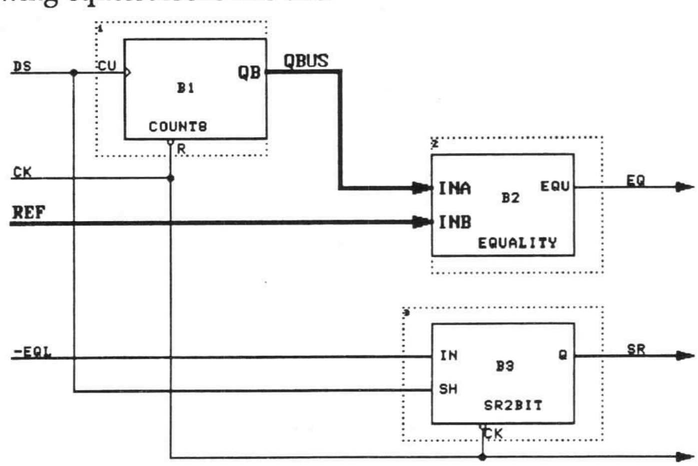
1. Move the graphics cursor into the alphanumeric field that contains **equtest**.

Note: Make sure the current workspace is named before you move down the design tree. If you started this Session by loading the **scontrol** *drawing, or if you have saved your drawing then your workspace is named. Make sure your workspace is named before you proceed with this Session.*

2. Press Ctrl – PgDn (on a Sun, press Ctrl – R15). This causes several things to happen:

   - First, the current workspace is saved into a temporary file in your current directory.
   - Second, FutureNet searches for the drawing file **equtest.dwg.** Because no device or path was specified in the FILE 8 field, FutureNet looks for this file in the current directory. If **equtest** is not found, FutureNet prompts you to create the file. If you choose to create the file, FutureNet displays an empty workspace and prepares to store all new contents as **equtest.dwg** next time the SAVE command is entered. If you do not choose to create the file, FutureNet returns you to the current workspace and does not create the new drawing file.
   - Third, if equtest.dwg is found, the drawing is loaded and the graphic cursor is moved into the top left corner. In this instance, the completed drawing file has been provided for you and is loaded into the workspace.

3. Press PgDn twice (on a Sun, press R15 twice) to zoom-in. The drawing **equtest** looks like this:



4. Look around the drawing using the scroll bars and [Home] (on a Sun, press [R7]). Remember, when you press [Home] (or [R7] on a Sun), FutureNet centers the drawing around the current position of the cursor. As you can see, **equtest** is an intermediate step drawing composed of three functional parts: an 8-bit counter, an equality checker, and a 2-bit shift register. The details of these parts are not included on this sheet. Instead, functional blocks that represent these sections point to more detailed drawings.

Step 3. Move Up in the Design Tree

Although **scontrol** and **equtest** both contain functional blocks, there are several differences between the blocks in **equtest** and the block in the root drawing **scontrol**.

To study the differences between the blocks, move back and forth between the files, do the following:

1. Press [Ctrl] – [PgUp] (on a Sun, press [Ctrl] – [R9]) to return to **scontrol**. **equtest** is saved in a temporary file and the temporary version of **scontrol** is reloaded into the workspace.
2. Notice that the graphic cursor is positioned in the file name pointer field for **equtest**.
3. To return to **equtest**, press [Ctrl] – [PgDn] (on a Sun, press [Ctrl] – [R15]) without moving the graphics cursor.

   Look at the differences between functional blocks in the two drawing files. In **equtest**, the .B command and symbol definition mode instructions were used to create the functional blocks. As a result, the blocks have a single, rather than double outline. In addition, because the blocks do not touch the symbol cell boundary at all edges, some pin stubs were required in the drawing.

   In **equtest**, no part name fields were used in the functional blocks. These fields are optional, although they add to user readability. The character size for filename pointers also makes no difference, although in this example, the '2 size characters are used in **equtest**, and the larger '3 size characters are used in **scontrol**.

Step 4. Associate Signal Nets Between Drawing Levels

Now look at the signal names in **equtest**. Three of these names (DS, CK, and REF) match pin names in the root level drawing, **scontrol**.

These matching names associate the signals DS, CK, and REF in **equtest** with the signals that attach to pins DS, CK, and REF in **scontrol**: DSHAFT, CLOCK, and REFSPD. As a result, when this structure is run through the Pin List Generator, the signal names DSHAFT, CLOCK, and REFSPD are substituted for the names DS, CK and REF.

Note: Before leaving this step, make sure that **equtest** *is on your screen.*

Step 5. Move Between Files in a Drawing Set

The structure branches downward into three separate drawings, **count8, equality,** and **sr2bit**. In addition, the current drawing level is a two-sheet drawing set that also contains the file **slogic**. Any of these four paths may be chosen to continue toward the completion of the design tree.

We know that workspaces for **count8, equality,** and **sr2bit** can be created by using the [Ctrl] – [PgDn] command (on a Sun, press [Ctrl] – [R15]). Although the workspace for **slogic** can be created by first pressing [Ctrl] – [PgUp] (on a Sun, press [Ctrl] – [R9]), placing the graphics cursor on the filename pointer toward **slogic**, and then pressing [Ctrl] – [PgDn] (on a Sun, press [Ctrl] – [R15]), FutureNet offers a way to do this with a single command:

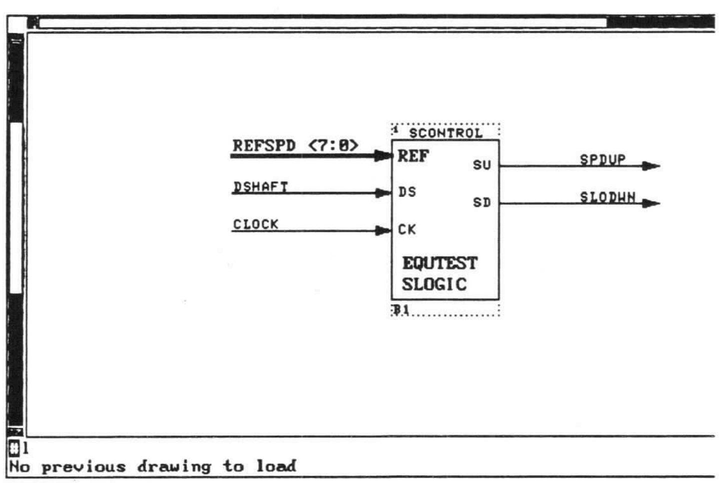
1. Enter the **#R** command. **equtest** disappears and is replaced by **scontrol**. After a moment, **scontrol** disappears, and is replaced by the sample drawing **slogic.dwg**.

   The #R command is the **Move Right in Design Tree** command. When entered, **equtest** was saved into a temporary file. FutureNet then reopened the temporary file for **scontrol** and searched for a filename pointer in the functional block that was to the right or lower than the pointer to **equtest**. When it found the **slogic** pointer, it loaded that file into memory.

2. Examine the detail of the **slogic** drawing.

3. Enter the **#L** command to return to **equtest**. This is the **Move Left in Design Tree** command. This command works just like the #R command, except that it searches for a filename pointer to the left or above the position of the pointer to the file that the command was entered from.

4. With #L still on the command line, press [↵] or the middle mouse button to enter the command. Your screen looks like this:



What happened was that you moved up a drawing level in the design tree. FutureNet loaded and searched the functional block in **scontrol** for a filename pointer that was to the left or above the pointer to **equtest**. But, because it did not find one, FutureNet was unable to move back down to the current drawing level, **scontrol** remained in the workspace. The message:

```
Drawing not found in data structure
```

means that drawing to the left is nonexistent.

Step 6. Move Through Multiple Drawing Levels

The previous section demonstrated a shortcut to move laterally between sheets on a single drawing level. This section describes a shortcut to move vertically between drawing levels.

To move down the design tree to the detail drawing of the 8-bit counter **(count8.dwg)** using the [Ctrl] – [PgDn] command (on a Sun, press [Ctrl] – [R15]), it would be necessary to traverse several drawing files. To simplify this procedure, FutureNet offers a more direct movement command, #D. Enter:

#D count8

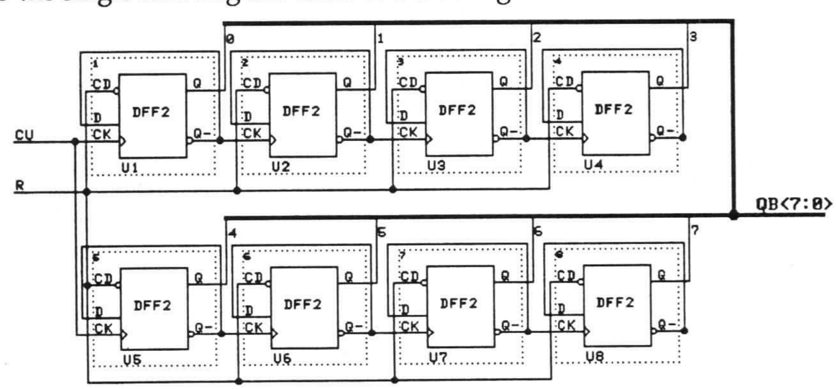
This command saves the current drawing in a temporary file and loads **count8** into the workspace. When jumping to files using this command, [Ctrl] – [PgUp] (on a Sun, press [Ctrl] – [R9]) returns you to the file from which the command was entered.

A similar command, #U *filename*, enables you to move upward in the design tree to any higher level drawing along the current drawing path.

Step 7. Multiple Pointers to the Same Drawing File

The drawing **count8** illustrates another structured design feature.

Note that the FILE attribute field in each of the eight flip-flops all point to the single drawing file **dff2.** The drawing file **count8** looks like this:



When the structure is run through the Pin List Generator, this file is accessed eight times. Each access is given its own unique reference number, allowing each access to be treated as if it were a single unique sheet. As a result, you do not have to replicate drawing files when creating a design.

Move through the structure to the 2-bit shift register detailed in the drawing file **sr2bit.** To get to this file from **count8:**

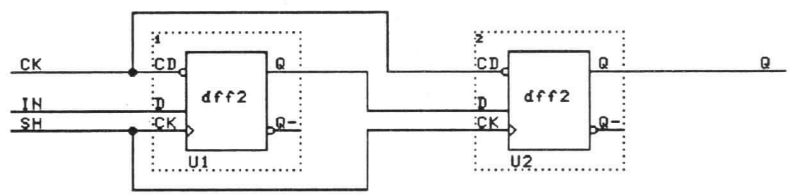
1. Press [Ctrl] – [PgUp] (on a Sun, press [Ctrl] – [R9]) to return to the root drawing.

Note: Because count8 was accessed from the root drawing with #D, ***Ctrl-PgUp*** *(on a Sun, press* ***Ctrl-R9****) returns you to that drawing level.*

2. Enter:

   #D sr2bit

   This loads the 2-bit shift register into the workspace. Press [PgDn] (on a Sun, press [R15]) to observe the detail of **sr2bit.**



sr2bit contains two flip-flops that are detailed in the drawing file **dff2.** This file is the same drawing that was pointed to in the file **count8.** This shows how a file can be pointed to multiple times on a single sheet, and on separate sheets in separate drawing levels.

Step 8. Display All Accessed Files

It is sometimes helpful to see which files have already been accessed during the session, and which files are along the current drawing path.

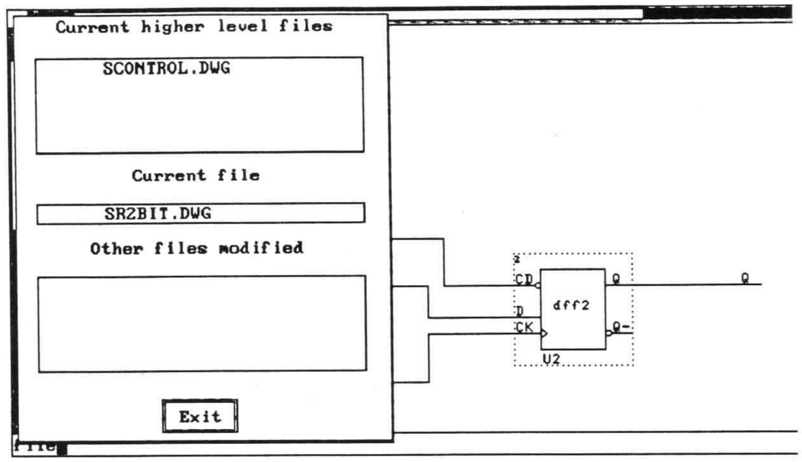
Enter the FILE command.

This display lists the following:

- The path to the root drawing from which the current drawing was traversed.
- The current drawing.
- All other drawing files that have been modified during the session.

Exit the display by pressing any key.

The FILE command is especially useful when using #D and #U, since the file from which the current file was entered may not be on the next higher drawing level.



Step 9. Save All Modified Drawings

In previous sessions you've learned that the SAVE command will store the drawing file currently in the workspace. This command, however, has no effect on the temporary files that have been created while traversing the design tree.

To simultaneously save all files that have been modified in the session, enter the SAVEALL command.

All temporary files are renamed with the extension specified in the filename pointers (or the **.dwg** default).

Step 10. Clear the Workspace

Once you have saved your drawings you may want to clear the workspace in order to work on another design. The ERASE command will clear the screen but will not affect any temporary files that have been made during the session.

To erase both the screen and these temporary files, enter the CLEAR command.

Summary

This concludes the sessions on structured design. For further information, refer to Chapter 1, "Understanding FutureNet," for information on functional block and structured design implementation. Also, see the *Command Reference* manual for more information on all structured design commands.

In summary, here are the commands covered in this chapter:

Command	**Function**
#D *filename*	Moves down in the hierarchical design directly to the drawing named in *filename.*
#L	Moves left to an adjacent drawing at the same level in the hierarchical design.
#R	Moves right to an adjacent drawing at the same level in the hierarchical design.
#U *filename*	Moves up in the hierarchical design directly to the drawing named in *filename.*
CLEAR	Erases the contents of both the drawing screen and the temporary files.
FILE	Displays the names of all drawing files in the current drawing set that have been loaded for editing.
SAVEALL	Saves all drawing files that have been modified during the editing session.
[Ctrl] – [PgDn] or [Ctrl] – [R15]	Moves down to the next lower level drawing in the hierarchical design.
[Ctrl] – [PgUp] or [Ctrl] – [R9]	Moves up to the next higher level drawing in the hierarchical design.

14 Session 8: Symbol Definition

This is an advanced session for users who want to create custom symbols that are not easily defined with the Create Block Symbol and Add Interconnect Stub commands.

This session introduces these FutureNet functions:

- Enter and exit symbol definition mode
- Define the symbol cell size
- Move the symbol definition cursor
- Draw line segments
- Draw diagonal lines
- Draw pins
- Insert symbol and graphic elements
- Delete, replace, and insert instructions

This session guides you through the creation of a variable resistor symbol. To create this relatively complex symbol, you must use the FutureNet symbol definition mode.

Step 1. Before You Begin

1. Start FutureNet, if necessary.
2. If you are already in FutureNet, save any work in progress, then enter the CLEAR command to clear your workspace.

Step 2. Enter Symbol Definition Mode

You can enter symbol definition mode from full scale, intermediate, or reduced zoom level. For this session, work in full scale zoom.

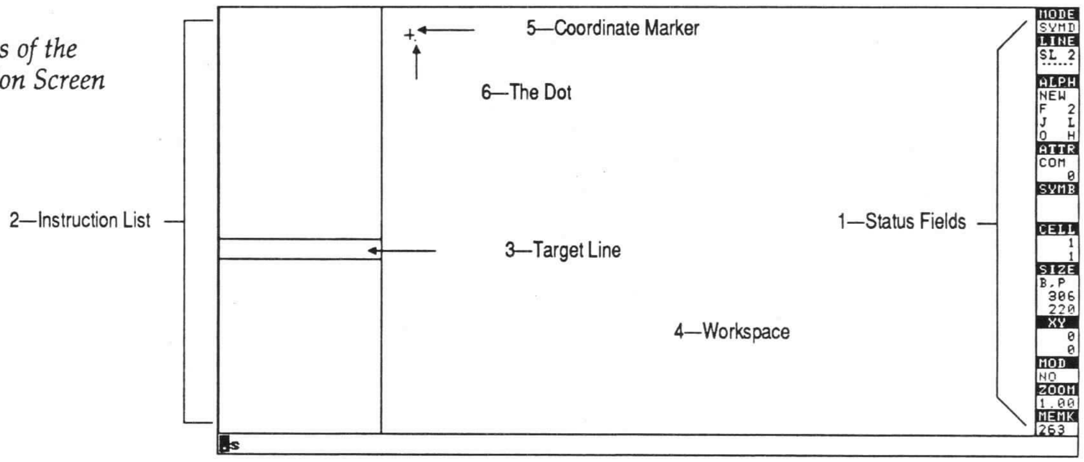
Enter the .S command.

The symbol definition screen appears.

Step 3. The Symbol Definition Screen

The symbol definition screen is pictured below, and its fields are called out and explained.

Figure 14-1
The Components of the Symbol Definition Screen



1—Status Fields

Although the status fields are still displayed on the right side of the screen, only the CELL field displays information pertinent to this mode. Symbol cell information appears in the CELL field when entering symbol definition mode to edit an existing symbol already in the drawing workspace. During creation of a new symbol, however, no information will be displayed in the CELL field.

The XY field shows where the upper left corner of the symbol will be put in the drawing workspace and the SYMB field shows whether the symbol has been rotated or reflected. Note that when you use symbol definition mode to edit a symbol already in a drawing that has been rotated or reflected, the symbol is shown here in it's original *orientation* in this mode.

2—Instruction List

The vertical area that borders the left side of the screen is the instruction list, where the symbol definition instructions are displayed.

3—Target Line

The area between the two horizontal lines in the instruction list is called the target line. Only the instruction on the target line can be deleted or replaced. In insert mode, lines are inserted before the instruction on the target line.

If you make a mistake on the command line, use Ctrl and ← or → to move the cursor.

4—Workspace

The main portion of the screen, to the right of the instruction list, displays the symbol defined by the instruction list, and is called the workspace.

When symbol definition mode is entered with the graphics cursor positioned within a symbol boundary on the drawing screen, that symbol is displayed here (unrotated and unreflected, if necessary).

If entered when the graphics cursor is not inside a symbol boundary, as is the case here, the screen displays a cross symbol and a dot.

5—Coordinate Marker

The small cross is called the coordinate marker. It moves in absolute X,Y increments relative to its last position, and its position reflects the last x,y coordinate value after entering the symbol definition instructions.

6—Dot

The dot marks the lower right corner of a null symbol cell, with height and width of one. The lines of the cross overlap the other four corners of this cell. When a larger symbol cell size is entered, or when a symbol from a drawing is being edited, the larger rectangle of that symbol cell boundary is displayed.

Entering Symbol Definition Instructions

Symbol definition instructions are entered on the command line or chosen from the Symbol Definition menu. The menu is called up by pressing the right mouse button or entering **menu** on the command line.

When a command is entered, the command is copied to the instruction list and the instruction list moves up.

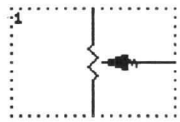
The following commands manipulate the instruction list, moving it through the target line.

.CLR	Clears all symbol definition instructions.
.D	Deletes the instruction on the target line.
.I	Toggles insert mode on and off. In insert mode the target line is solid above and dotted below, and the instruction entered on the command line is placed before the instruction on the target line.
n [↓]	Moves the instruction list down *n* lines through the target line.
n [↑]	Moves the instruction list through the target line.
[PgDn]	Moves the blank line at the end of the instruction list to the target line.
[PgUp]	Moves the first instruction in the instruction line to the target line.
[Home]	Centers the workspace window on the coordinate marker.

Step 4. Define the Symbol Cell Size

To begin defining a new symbol, you must first specify the symbol cell size. The symbol cell size instruction is a dot (.) followed by the width and height of the symbol cell in display units.

Look at the variable resistor symbol:

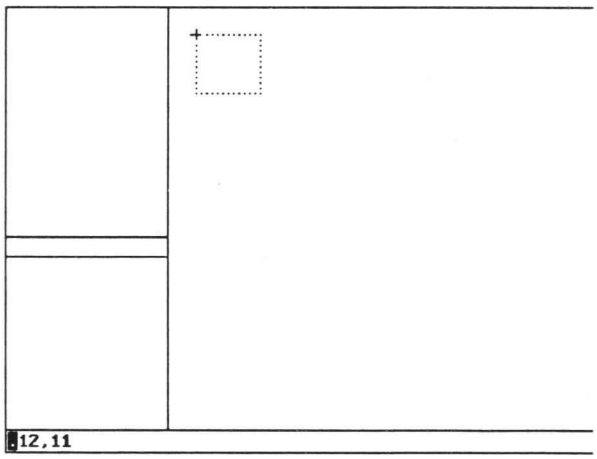


The symbol cell size of this resistor is 12 display units wide by 11 display units high.

Enter on the command line:

.12,11

to set the symbol cell to this size.



A symbol cell of the specified dimensions is displayed as a dotted rectangle. The symbol definition cursor remains at the upper left corner of the cell.

While using symbol definition mode, the symbol you create is simultaneously inserted into the drawing workspace. The upper-left corner of the symbol cell is positioned at the location specified by the XY status field.

If the symbol cell is too large to fit into the drawing at this location without overlapping other drawing elements, you are prompted:

`Boundary conflict`

In this case, either redefine the cell to a smaller size or exit symbol definition mode (using the .Q command) and move the graphic cursor into a less crowded area of the drawing. In this example, we started with an empty workspace so this problem does not exist.

Step 5. Move the Symbol Definition Cursor

To duplicate the resistor symbol, you must first move the coordinate marker away from the left symbol boundary. This is done with a Move Coordinate Along X-axis instruction.

Enter:

MXS 2

This is the "short" version of the instruction. It places the coordinate marker two display units to the right of its previous location. **MXS** instructions can specify values between -16 and 15. If a negative value is used, the coordinate marker moves to the left.

The "long" version of the Move Coordinate Along X-axis instruction, **MX**, allows values between -128 and 127, but requires more memory than the **MXS** version.

The coordinate marker moves along the Y-axis in a similar fashion by using the **MYS** and **MY** instructions.

About Display Units and Dot Units

The **MX, MY, MXS,** and **MYS** instructions move the symbol definition cursor in display unit increments—the standard unit of distance in a FutureNet drawing. Each display unit is, in turn, made up of four pixels, or dot units.

There may be times where positioning in dot units is desired. The Move in Dot Units instructions, **MXD** and **MYD**, enable you to do this. These instructions can specify values between -128 and 127 dot units. There is no "short" form for these instructions.

Step 6. Draw Line Segment

To draw the upper interconnect stub of the resistor, use the Draw Line Along Y-axis instruction.

Enter:

DYS 4

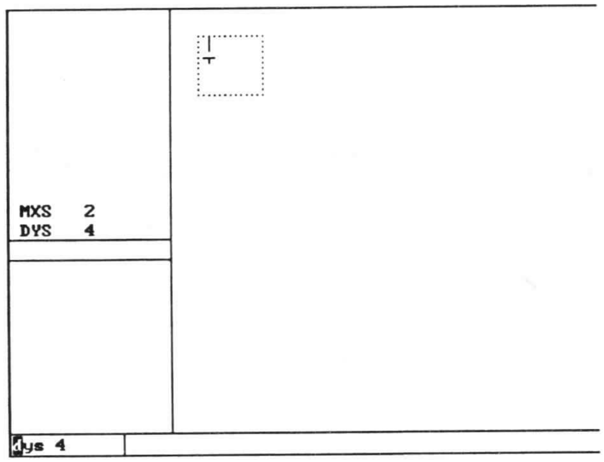
Again, this is the "short" version of this instruction. It draws a line of the length specified in the instruction and relocates the symbol definition cursor at the end of it. If a negative value is entered, the segment is drawn upward.

The "long" version of the instruction, **DY**, could also be used here, but requires more memory. **DY** can use values between -128 and 127, while **DYS** values can only range between -16 and 15.

To draw segments along the X-axis, use the **DX** and **DXS** instructions. Negative values draw lines to the left. Positive values draw lines to the right.

You can also use the **DXD** and **DYD** instructions to specify line lengths in dots, rather than in display units. **DXD** draws a segment along the X-axis. **DYD** draws along the Y-axis. Both instructions can contain values between -128 and 127. There is no "short" version of these instructions.

If you have entered the instructions properly to this point, the screen looks like this:



Step 7. Draw Diagonal Lines

Now you are going to draw several diagonal lines. The first line is two dot units down and to the right of the current coordinate marker location.

1. Enter:

   DXYD 2,2

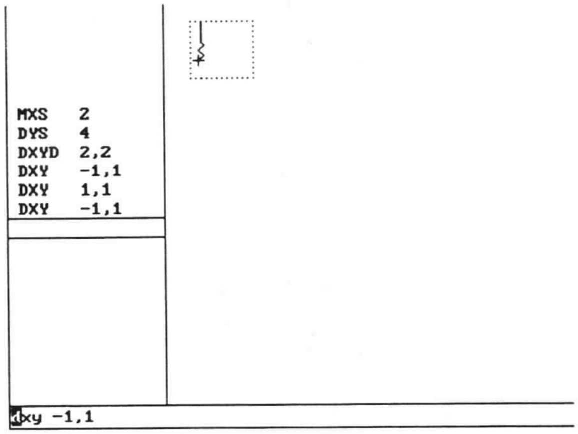
   This is a Draw Diagonal in Dot Units instruction. It can accommodate values between -128 and 127. Like other coordinate-relative commands, positive values move down or right, while negative values move up or left.

   Use the Draw Diagonal in Display Units instruction to draw the next three segments, each of which is a single display unit (four dots) long.

2. Enter each of the following commands:

   DXY -1,1
   DXY 1,1
   DXY -1,1

   If you entered these instructions correctly, the screen looks like this:



3. Like the first segment, the final diagonal segment is two dots toward the lower right. To draw this segment, enter:

   DXYD 2,2

Step 8. Draw Interconnect Stubs

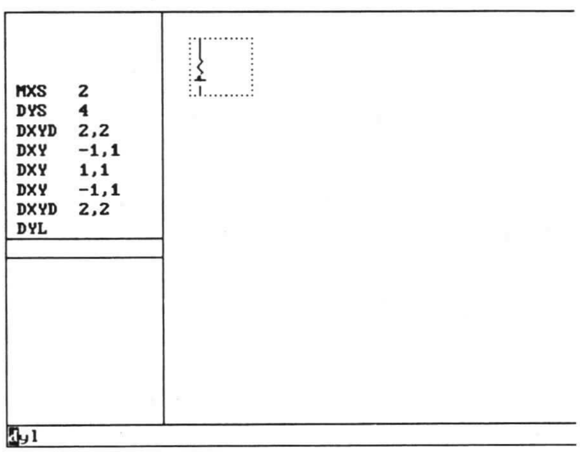
Now draw an interconnect stub to the symbol boundary. Although you already know how to do this using the Draw Line Along Y-axis instruction, a simpler method enables you to draw the interconnect stub without having to calculate its length.

1. Enter:

   DYL

   This is the Draw Y to Lower Symbol Cell Boundary instruction. It draws a segment from the current marker position to the bottom edge of the symbol cell.

   As a result, your screen looks like this:



Unlike other drawing instructions, this command does not update the position of the symbol definition cursor.

You can also draw a third interconnect stub using this type of command.

2. Reposition the symbol definition cursor one display unit to the right and two display units up by using the single command that allows movement along both the X and Y-axis simultaneously. Enter:

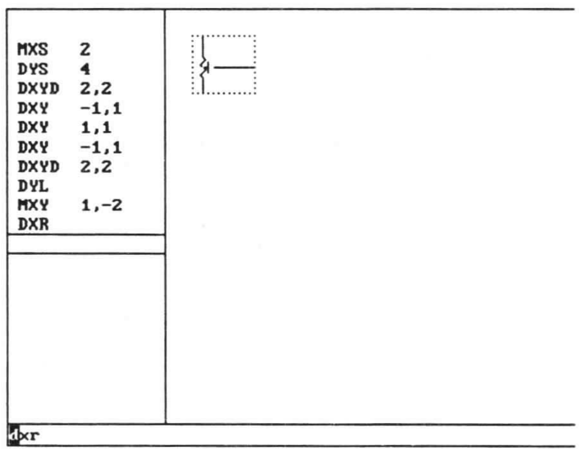
   MXY 1,-2

   The cursor moves one unit in a positive direction (right) along the X-axis and two units in a negative direction (up) along the Y-axis.

3. You can now draw the interconnect stub to the right symbol boundary with the Draw X to Right Symbol Cell Boundary command. Enter:

   DXR

   The screen looks like this:



Other Draw to Symbol Cell Boundary interconnect stub instructions include **DYU** and **DXL**. These instructions draw stubs to the upper and left symbol cell boundaries, respectively.

Step 9. Insert Symbol Element

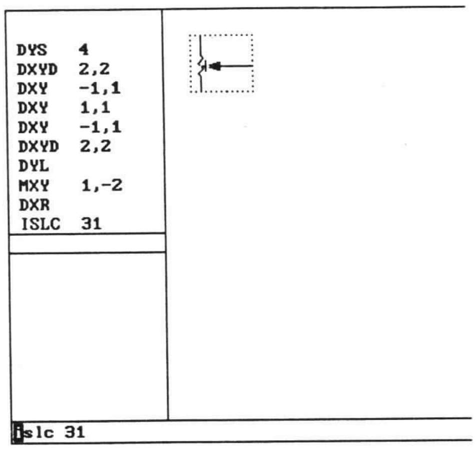
To complete the symbol, place a left arrow at the current location. FutureNet provides an online library of common solid symbol elements (such as arrowheads and dots) that can be inserted while in symbol definition mode. A table of the available elements is provided in Appendix B of this manual.

The left arrow you want to use is symbol element 31. This element can be placed in the symbol so that the center of the element, or the corner or center of any edge is positioned at the coordinate marker. In this instance, place the element so that the tip of the arrow touches the current coordinate marker location.

Enter:

ISLC 31

The screen looks like this:



Note that the symbol definition cursor position does not change.

To learn the command formats needed to position other edges of the symbol at the coordinate marker, refer to the symbol definition commands in the *Command Reference* manual.

Step 10. About Graphic Elements

FutureNet also provides an online library of graphic elements. These elements are ready-made instruction lists that generate many of the basic shapes used to create electronic symbols. They are inserted with the **IG** command.

Most gate types (interconnect stubs included) are available as graphic elements. Shapes used in making symbols for transformers, capacitors, resistors, and other discrete parts are also included.

Among these elements are the jagged diagonal lines used to make resistor symbols. Recall that you used diagonal drawing commands to create these lines. Although the variable resistor symbol is acceptable as is, replacing the diagonals with graphic elements introduces you to several other symbol definition mode features.

You can find a table of these elements in Appendix A of this manual.

Step 11. Manipulate the Instruction List

Before you can edit the instruction list, you must first learn how to manipulate the position of the instruction list in relation to the target line:

- Press [PgUp] to move the first instruction in the list down to the target line. It is the instruction list, rather than the target line itself, that moves. (Note that in many applications, you move a "window" over data in a list. Here, you are actually moving the list.)
- Press [PgDn] to move the blank following the last instruction up to the target line.
- Press [↑] and [↓] to move the instruction list up or down one or more lines.

  Normally, [↑] and [↓] move the instruction list up or down in one line increments. To move multiple lines at a time, type the desired increment on the command line and then press [↑] or [↓].

Manipulate the Target Line

Move the instruction list down 9 spaces, so the target line is resting on the third item in the instruction list.

1. Type **9** on the command line and press [↓].

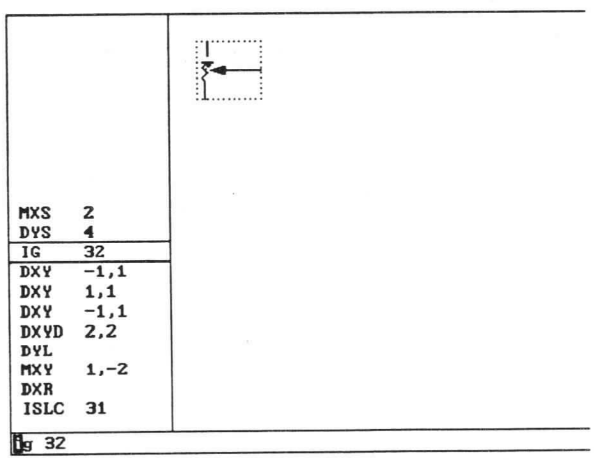
   The instruction list shifts downward to leave DXYD 2,2 between the target lines. This is the position of the first line that you must change.

Replace Instruction

Now you are going to place graphic element 32 at this point in the drawing. This element comprises half of the complete resistor symbol.

2. Clear the command line by pressing [Backspace], then enter:

   IG 32



A number of things occur simultaneously:

- The instruction that was previously on the target line is replaced by the new command.
- Several elements in the symbol display shift upward. This is because these elements are drawn relative to previous coordinate marker locations. When a line is changed or deleted, all successive instructions are affected.
- The doubling of lines, forming a tiny parallelogram, occurs because the position of the symbol definition cursor is not changed when inserting graphic elements. As a result, both the element and the line drawn by the first **DXY** instruction start at the same point.

3. Press [↑] to move the instruction list up one line. This places DXY -1,1 on the target line.

Delete Instructions

At this point you could continue changing commands. Instead, delete the unwanted instructions.

4. Enter:

   .D

5. Press [↵] 3 more times.

   Each time you press [↵], the instruction on the target line disappears and the instructions below it shift up a line. As a result, the remaining diagonal line commands are removed.

Insert Instructions

Now, insert additional commands into the instruction list. Insert mode is used to do this.

6. Enter:

   .I

   You can enter and exit insert mode by entering the **.I** command.

When you enter insert mode, the lower segment of the target line becomes dotted. Any instruction typed on the command line is inserted into the list just above the target line. The instruction that was on the target line remains unchanged.

7. Try entering the other segment of the resistor symbol. Enter:

   IG 32

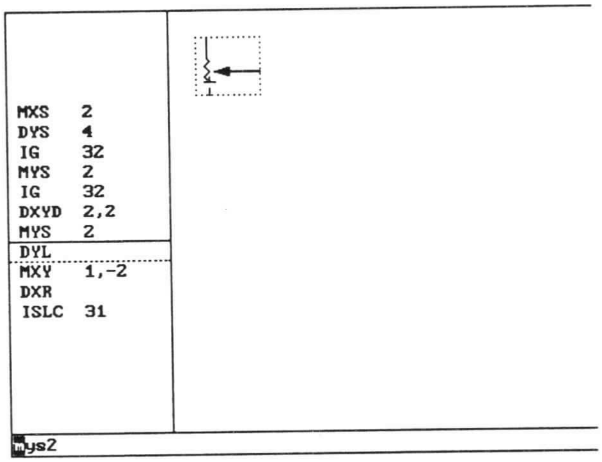
   Although the new instruction is inserted in the list, the symbol display appears unchanged. This is because the symbol definition cursor is not updated by the **IG** command. As a result, the two graphic elements overlay each other.

8. This is easily remedied by inserting a Move Along Y-axis instruction. Press [↓] once so that the second **IG** instruction is on the target line. Then type **MYS 2** and press [↵].

   The cursor shifts down two display units (the length of the graphic element) before entering the second **IG** instruction.

9. Now complete the symbol. Move the instruction list up two lines, pressing [↑] so that **DYL** is on the target line. With the MYS 2 instruction still on the command line, press [↵] once again to insert another command.

The screen looks like this:



Step 12. Exit Symbol Definition Mode

Type **.Q** on the command line and press [↵] to exit symbol definition mode.

FutureNet returns to normal editing mode. The graphic cursor is in the same position it was in when you entered symbol definition mode. The only difference is that the symbol you defined, if any, is now in the drawing.

Once a symbol is defined, you can use it just as you used the block symbol in Session 3. Use alphanumeric mode to annotate it. When it is complete, save it in a symbol library for future use.

Step 13. About Editing an Existing Symbol

If you decide the symbol you just defined needs to be modified, re-enter symbol definition mode by typing **.S** and pressing [↵] while the graphic cursor is in the symbol. The symbol definition instructions of that symbol are then available for editing.

Note: *Symbols created in symbol definition mode cannot be modified on the drawing screen without additional steps. For example, pins or lines of a symbol created in the symbol definition mode cannot be deleted or added in the drawing base mode without additional steps. Refer to information about the* **.SBS** *command for more information about modifying symbols created in the symbol definition mode.*

Summary

You have now completed Session 8.

In summary, here are the commands you learned:

Command	Function
.S	Enters symbol definition mode.
.w,h	Creates a block symbol.
.Q	Exits symbol definition mode.
MXS	Moves the cursor along the X axis.
DYS	Draws a line along the Y axis.
DXYD	Draws a diagonal in dot units.
DXY	Draws a diagonal in display units.
DYL	Draws a line along the Y axis to the symbol cell boundary.
MXY	Moves the cursor along the X and Y axis.
DXR	Draws a line along the X axis to the right symbol cell boundary.
IS	Inserts a symbol element.
PgUp	Moves the first instruction to the target line.
PgDn	Moves a blank line at the end of the instructions to the target line.
# ↑ or ↓	Moves the instructions up or down one line or the number of lines specified by #.
IGxx	Inserts a graphic element.
.D	Deletes an instruction on the target line.
.I	Inserts an instruction on target line.
Home	Centers the workspace window on the coordinate marker.

Notes

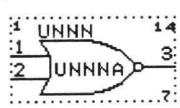
Draw Instructions begin to draw from the end cursor position of the previous instruction, so they're like drawing on a piece of paper without lifting the pen. Move, Save and Restore Coordinates Instructions all move the cursor without drawing; they're commands for "picking up the pen." While most instructions are performed relative to the end cursor position of the last instruction, **MXYA** moves the cursor to absolute coordinates in the symbol cell. Save and Restore instructions allow you to return to a relative cursor position by entering a save instruction in the instruction list, then restoring it later.

15 Session 9: Advanced Symbol Definition

This session introduces these FutureNet symbol definition mode functions:

- Moving the cursor in display units and dot units
- Drawing an arc
- Drawing lines in display units and dot units
- Drawing diagonal lines
- Drawing curves

This session guides you through the creation of the NOR gate illustrated below.



Step 1. Before You Begin

1. Start FutureNet, if necessary.
2. If you are already in FutureNet, save any work in progress, then enter the CLEAR command to clear your workspace.
3. Position the graphic cursor in the center of the screen.
4. Enter symbol definition mode. Enter the .S command. The coordinate marker and dot appear on the symbol definition screen.

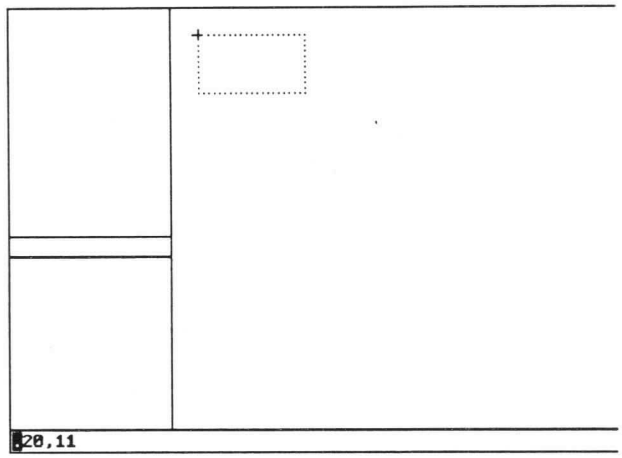
Step 2. Begin the NOR Gate

Now refer to the illustration at the beginning of this session and draw the NOR gate:

1. Define the symbol cell boundary, by entering the command:

   .20,11

   A dotted boundary 20 display units wide by 11 display units high appears. The CELL status field shows the cell size is 20/11.



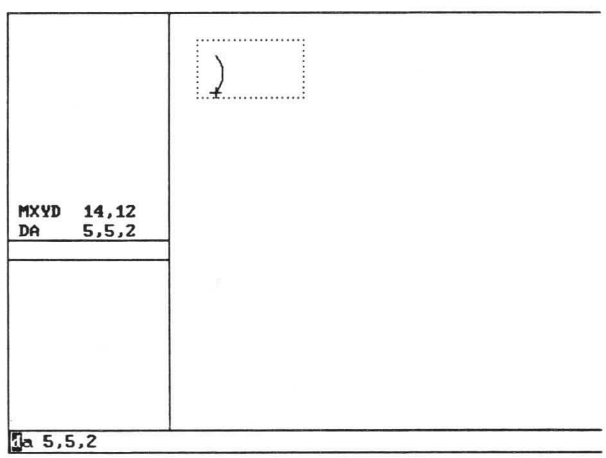
   You will draw the NOR gate inside this dotted boundary.

2. Begin drawing the gate. To move the cursor right (indicated by the positive x value) 14 dot units and down (indicated by the positive y value) 12 dot units., enter:

   MXYD 14,12

3. Draw the arc that makes up the left side of the symbol by entering:

   DA 5,5,2



   The first parameter 5 specifies the radius of the arc in display units. The second parameter 5 specifies the center of the arc relative to its start point. The third parameter 2 specifies the number of eighths of a circle to draw as an arc.

Step 3. Draw the First Curve

Now draw the curves that make up the top and bottom of the gate:

1. To draw a horizontal line right (indicated by the positive x coordinate) 26 dot units long, enter:

   DXD 26

2. To move the cursor diagonally to a point that is right (indicated by the positive x coordinate) one dot unit and up (indicated by the negative y coordinate) one dot unit, enter:

   MXYD 1,-1

3. To select decrement mode, enter:

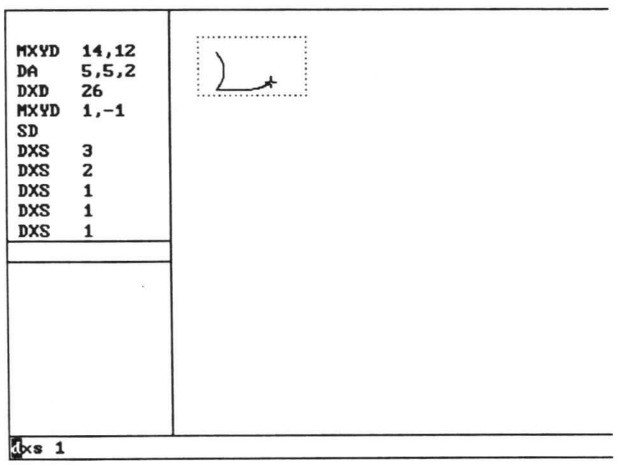
   SD

   This mode was developed for drawing curves using short stepped lines. Decrement mode automatically decrements the opposite coordinate by one dot unit after each Draw Short Line (DXS or DYS) command. So when you use DXS, the cursor automatically moves up one dot; when you use DYS, it moves left one dot.

Note: *The Select Increment Mode (SI) increments the opposite coordinate. DXS automatically moves the cursor down one dot unit, while DYS moves the cursor right one dot unit.*

4. Enter the following instructions to draw short lines:

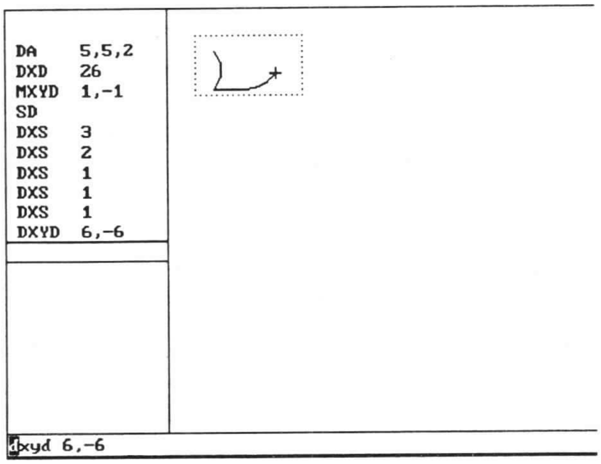
   DXS 3
   DXS 2
   DXS 1
   DXS 1
   DXS 1



The lines are all horizontal and to the right (because of the positive x values). The first is three display units long; the second is started one dot above the first line's endpoint and is two display units long, and so on until short lines have been drawn in incremented steps to form a curve.

5. To complete the bottom part of the gate, draw a diagonal line six dot units right and six dot units up. Enter:

   DXYD 6,-6



6. Save the current x,y coordinates. You will use these coordinates later to move the cursor back to this position with only one command. Enter:

   SXY

7. To draw a line one display unit up, enter:

   DY -1

8. Draw a diagonal line seven display units left and seven display units up. Enter:

   DXYD -7,-7

Step 4. Draw the Second Curve

1. Use the Draw Short Lines command to draw the stepped sets of horizontal lines that make up the top curve of the gate. Enter each of the following separate commands:

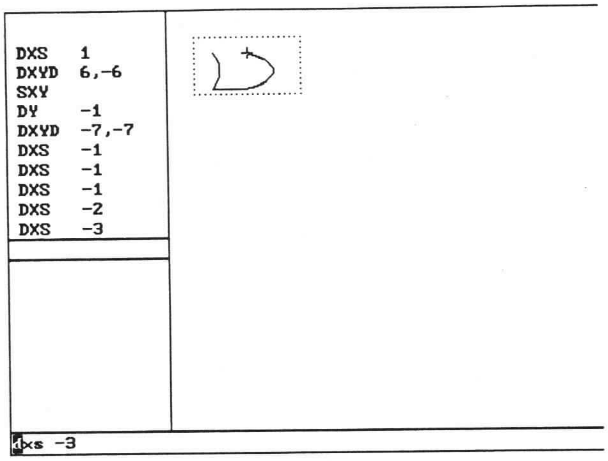
   DXS -1
   DXS -1
   DXS -1
   DXS -2
   DXS -3

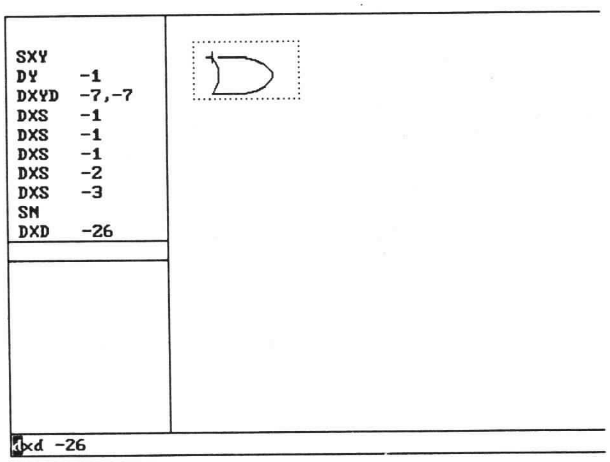
2. You are through drawing the decremented parts of the symbol, so exit decrement mode. Enter:

   SN

3. Complete the outline of the gate by entering the command which draws a horizontal line left 26 dot units.

   DXD -26





Step 5. Drawing Input and Output Pins

Now draw input pins 1 and 2 and output pin 3:

1. To move the cursor one display unit right and two display units down, enter:

   MXY 1,2

2. To draw a horizontal line to the left symbol cell boundary, enter:

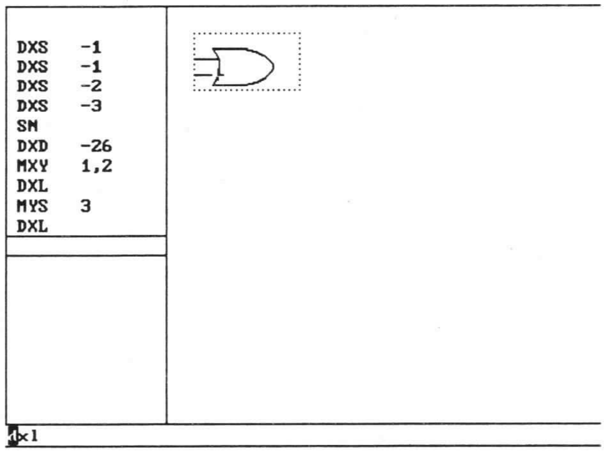
   DXL

3. Move the cursor down three display units and draw the second pin to the symbol cell boundary. Enter each of the following instructions:

   MYS 3
   DXL

4. Move the cursor to the x,y coordinate you saved with the SXY command above. Enter:

   RXY

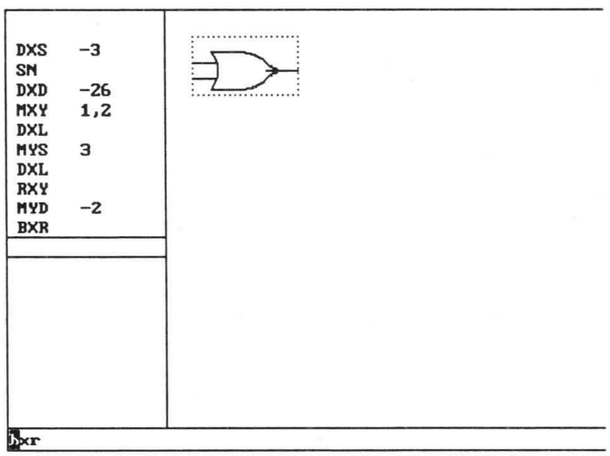


5. To move the cursor up two dot units, enter:

   MYD -2

6. Draw a horizontal line with a bubble to the right symbol cell boundary. Enter:

   BXR



You are now through drawing the outline of the NOR gate. You must still fill in all the appropriate alphanumeric data and assign attributes to the alphanumeric fields.

7. Exit symbol definition mode with the .Q command before typing alphanumeric data. The program returns to the drawing screen.

Step 6. Enter Alphanumeric Data and Attributes

1. Set the default font to size '2 by entering the command:

   '2

   (Remember that the current character size is displayed in the ALPH status field.)

2. Move the cursor to the area for the part name of the symbol, near the top-left corner of the gate (refer to the illustration at the beginning of this chapter, if necessary). To assign this field an attribute of PART/3, enter:

   'A PART

3. Press Esc to enter alphanumeric mode. Type:

   UNNNA

   Exit alphanumeric mode by pressing Esc .

4. In a similar manner, complete the following:

   a. Move the cursor to the area for the circuit designator, in the center of the symbol, on the same level as and one display unit to the right of the bottom input pin.

   b. Assign this field an attribute of LOC (for LOC/2) by entering the 'A LOC command.

   c. Enter alphanumeric mode and type its name: **UNNNA**

   d. Move the cursor to the intersection of the left symbol cell boundary and input pin 1.

   e. Assign this field an attribute of PINI (for PINI/23) by entering the 'A PINI command.

   f. Enter alphanumeric mode and type its pin number: **1**

   g. Press ↵ to move the cursor to the next alphanumeric field at pin 2 and type: **2**

   h. Exit alphanumeric mode.

   i. Move the cursor two display units left of the right symbol cell boundary and above output pin 3.

   j. Assign an attribute of PINO (for PINO/21) by entering the 'A PINO command.

   k. Change the justification to right justified by entering the **'J R** command.

   l. Enter alphanumeric mode and type: **3**

   m. Exit alphanumeric mode.

   n. Change the character size to '1 by entering the '1 command.

   o. Move the cursor to the area for the power supply pin, two display units down and three display units left of the upper right corner of the symbol cell boundary.

   p. Assign an attribute of +5V (for +5V/101) by entering the 'A 101 command.

   q. Enter alphanumeric mode and type: **14**

   r. Exit alphanumeric mode.

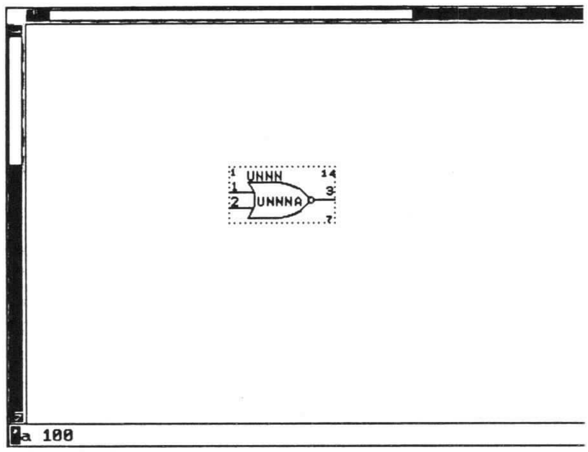
s. Move the cursor to the area for the pin to ground, two display units left of the lower right corner of the symbol cell boundary along the bottom boundary line.

t. Assign an attribute of GND (for GND/100) by entering the 'A 100 command.

u. Enter alphanumeric mode and type: **7**

v. Exit alphanumeric mode.

When you're through, the screen looks like the following illustration.



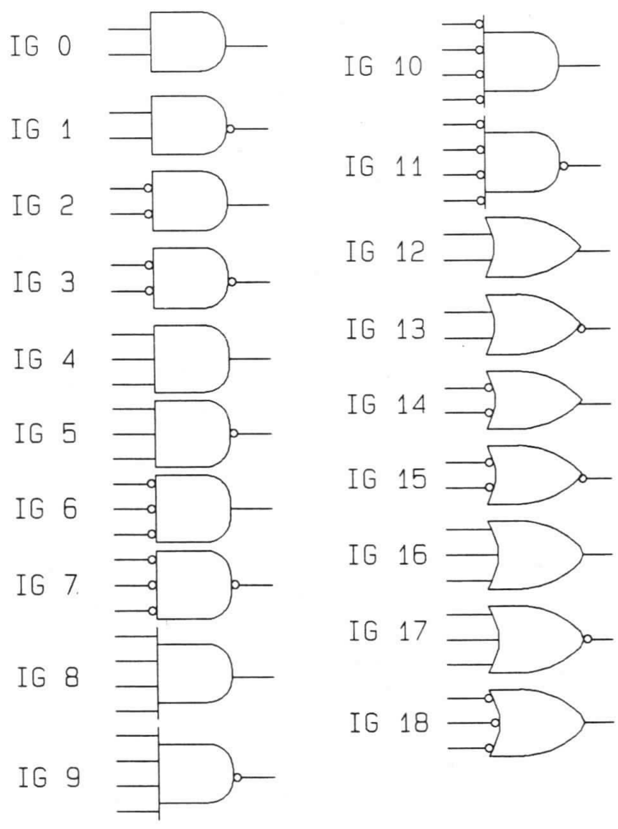
5. Follow the instructions in Session 2 to save this NOR gate in a symbol library called SESSION9. Then quit the editing session.

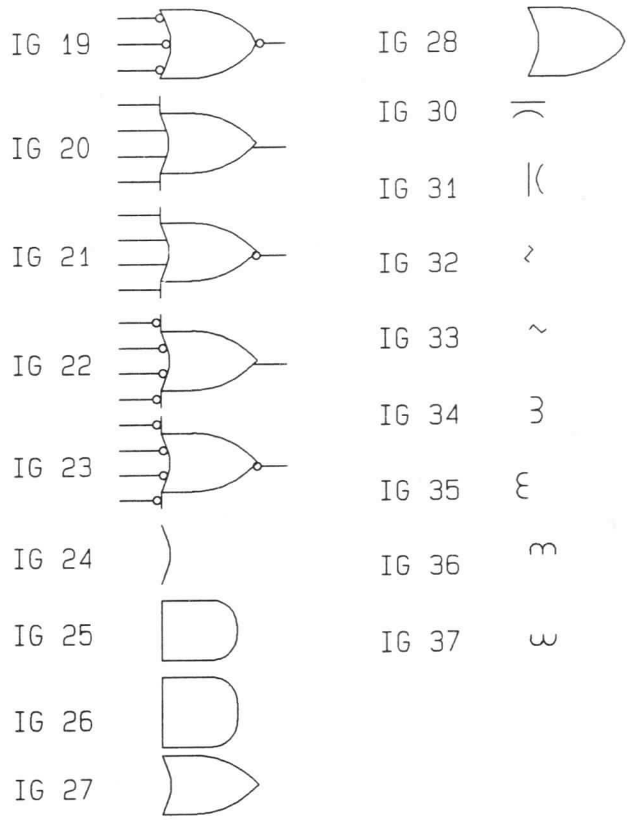
Summary

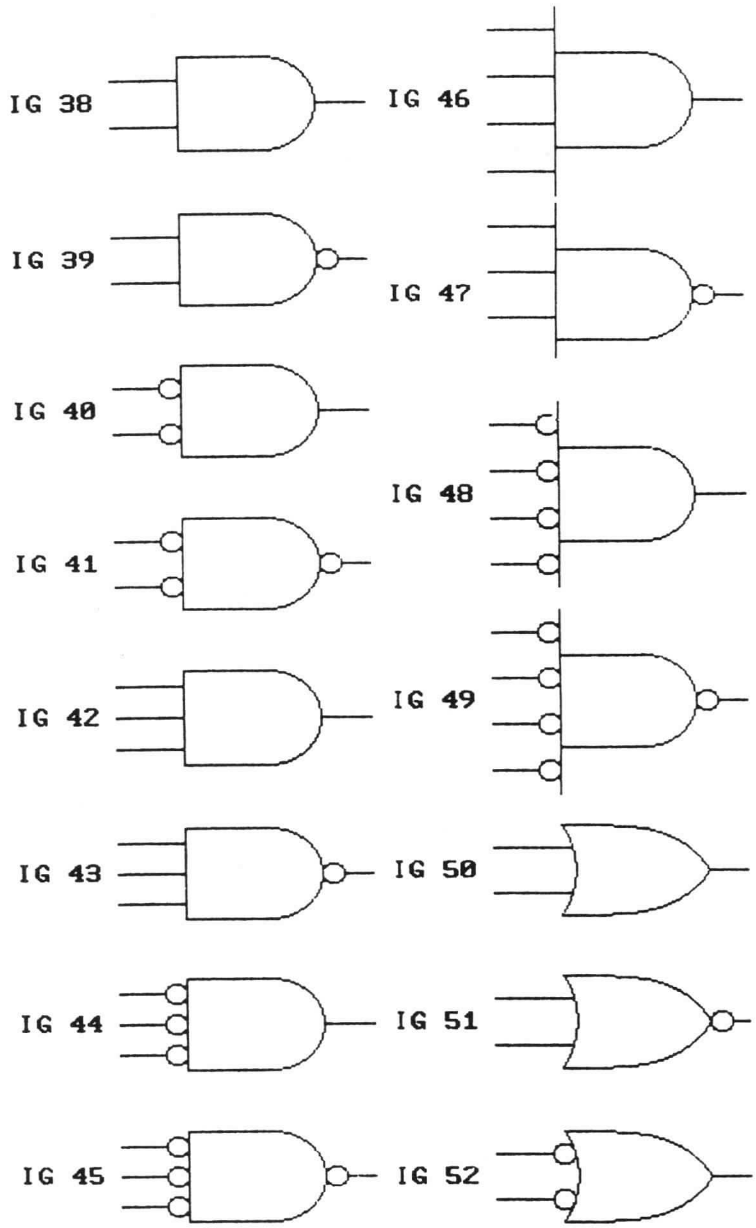
You have completed symbol definition sessions. The *Command Reference* manual contains complete information on all of the symbol definition commands. In summary, here are the commands you learned:

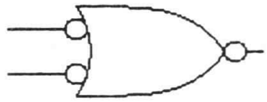
Command	**Function**
BXR	Draws a horizontal line with a bubble to the right symbol cell boundary.
DA	Draws an arc.
DXD	Draws a horizontal line in dot units.
DXL	Draws a horizontal line left to the symbol cell boundary.
DXS	Draws a short horizontal line in display units.
DXYD	Draws a diagonal line in dot units to specified coordinates.
DY	Draws a vertical line in display units.
MXY	Moves the cursor to new coordinates in display units.
MXYD	Moves the cursor to coordinates in dot units.
MYD	Moves the cursor in dot units and changes the y coordinate.
MYS	Moves the cursor and changes the y coordinate.
RXY	Restores and moves the cursor to the previously saved x,y coordinates.
SD	Selects decrement mode.
SN	Selects no increment/decrement mode.
SXY	Saves the x,y coordinates.

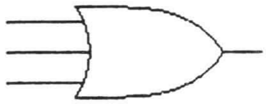
A Graphic Elements

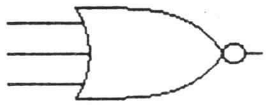


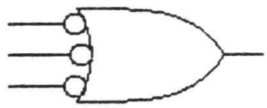

IG 19
IG 20
IG 21
IG 22
IG 23
IG 24
IG 25
IG 26
IG 27
IG 28
IG 30
IG 31
IG 32
IG 33
IG 34
IG 35
IG 36
IG 37

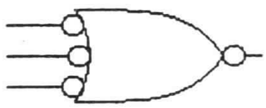

IG 38
IG 39
IG 40
IG 41
IG 42
IG 43
IG 44
IG 45
IG 46
IG 47
IG 48
IG 49
IG 50
IG 51
IG 52

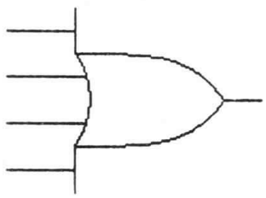
IG 53


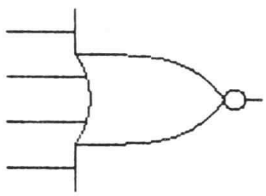
IG 54


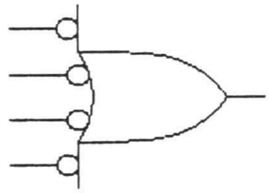
IG 55


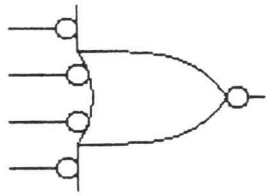
IG 56


IG 57


IG 58


IG 59


IG 60


IG 61


IG 62


IG 63


IG 64


IG 65


IG 66


IG 67


IG 68


IG 69


IG 70


B Symbol Elements

This appendix consists of two tables of symbols which are accessed by using the IS*xx* command.

The default symbol element table lists the graphic elements available by default each time FutureNet is started, and loaded when the RS command is executed.

The ASCII symbol element table lists the graphic elements available when an ST *n* command is executed.

Refer to the appropriate command for further information.

ISxx Command/Symbol

ISxx 11	•	ISxx 35	↖
ISxx 12	●	ISxx 36	↗
ISxx 14	>	ISxx 37	↘
ISxx 15	<	ISxx 38	▶
ISxx 16	∘	ISxx 39	◀
ISxx 30	→	ISxx 40	▲
ISxx 31	←	ISxx 41	▼
ISxx 32	↑	ISxx 42	▸
ISxx 33	↓	ISxx 43	◂
ISxx 34	↙	ISxx 44	▴

ISxx 45	ISxx 56
ISxx 46	ISxx 57
ISxx 47	ISxx 58
ISxx 48	ISxx 59
ISxx 49	ISxx 60
ISxx 50	ISxx 61
ISxx 51	ISxx 62
ISxx 52	ISxx 63
ISxx 53	ISxx 64
ISxx 54	ISxx 65
ISxx 55	ISxx 66

ASCII Symbol Element Table

ISxx 32	<SPACE>	ISxx 73	I	ISxx 114	r
ISxx 33	!	ISxx 74	J	ISxx 115	s
ISxx 34	"	ISxx 75	K	ISxx 116	t
ISxx 35	#	ISxx 76	L	ISxx 117	u
ISxx 36	$	ISxx 77	M	ISxx 118	v
ISxx 37	%	ISxx 78	N	ISxx 119	w
ISxx 38	&	ISxx 79	O	ISxx 120	x
ISxx 39	'	ISxx 80	P	ISxx 121	y
ISxx 40	(	ISxx 81	Q	ISxx 122	z
ISxx 41	)	ISxx 82	R	ISxx 123	{
ISxx 42	*	ISxx 83	S	ISxx 124	¦
ISxx 43	+	ISxx 84	T	ISxx 125	}
ISxx 44	,	ISxx 85	U	ISxx 126	~
ISxx 45	-	ISxx 86	V	ISxx 127	▓
ISxx 46	.	ISxx 87	W	ISxx 128	█
ISxx 47	/	ISxx 88	X	ISxx 129	α
ISxx 48	0	ISxx 89	Y	ISxx 130	β
ISxx 49	1	ISxx 90	Z	ISxx 131	■
ISxx 50	2	ISxx 91	[	ISxx 132	▓
ISxx 51	3	ISxx 92	\	ISxx 133	▓
ISxx 52	4	ISxx 93	]	ISxx 134	Φ
ISxx 53	5	ISxx 94	^	ISxx 135	≥
ISxx 54	6	ISxx 95	_	ISxx 136	⌶
ISxx 55	7	ISxx 96	`	ISxx 137	▓
ISxx 56	8	ISxx 97	a	ISxx 138	▓
ISxx 57	9	ISxx 98	b	ISxx 139	▓
ISxx 58	:	ISxx 99	c	ISxx 140	✦
ISxx 59	;	ISxx 100	d	ISxx 141	⎍
ISxx 60	<	ISxx 101	e	ISxx 142	¬
ISxx 61	=	ISxx 102	f	ISxx 143	Ω
ISxx 62	>	ISxx 103	g	ISxx 144	π
ISxx 63	?	ISxx 104	h	ISxx 145	µ
ISxx 64	@	ISxx 105	i	ISxx 146	✦
ISxx 65	A	ISxx 106	j	ISxx 147	Σ
ISxx 66	B	ISxx 107	k	ISxx 148	▽
ISxx 67	C	ISxx 108	l	ISxx 149	◇̲
ISxx 68	D	ISxx 109	m	ISxx 150	◈̲
ISxx 69	E	ISxx 110	n	ISxx 151	◇̄
ISxx 70	F	ISxx 111	o	ISxx 152	◈̄
ISxx 71	G	ISxx 112	p	ISxx 153	◁
ISxx 72	H	ISxx 113	q	ISxx 154	▷

C Attributes and Properties

Introduction

This appendix lists the FutureNet attributes and properties. Attributes are listed in both alphabetical and numerical order. Properties are listed after both attribute listings.

Attributes and Properties

Both attributes and properties are assigned to schematic elements through alphanumeric fields. An alphanumeric field is a displayed line of text on the schematic with an associated attribute, layered text, point of effect, and points of justification and orientation.

Attributes

An attribute is a set of properties, represented by a name or number, that is characteristic of a schematic element. One attribute is assigned to each alphanumeric field in a FutureNet drawing. Attributes can be referenced by either their name (mnemonic) or number. Attributes provide part of the information about a schematic element that is needed by the Post processors. For example, an attribute can indicate that the displayed text in an alphanumeric field is a signal name, part number, pin name, etc.

Table C-1 lists attributes in alphabetical order. Table C-2 lists them in numeric order.

Table C-1
Attributes Listed in Alphabetical Order

+12V	**105**	Symbol power pin (+12V) identifier. Used with emitter- coupled logic. This attribute implies an implicit signal name +12V is connected to this pin, and will be part of any signal net explicitly named +12V.
+5V	**101**	Symbol power pin (+5V) identifier. This attribute implies an implicit signal name +5V is connected to this pin, and will be part of any signal net explicitly named +5V.
-12V	**106**	Symbol Power pin (-12V) identifier. Used with emitter- coupled logic. This attribute implies an implicit signal name -12V is connected to this pin, and will be part of any signal net explicitly named -12V.
BNAM	**132**	Set bus identifier. Same as BUS/9.
BPIN	**141**	Bus pin identifier. Used to pass a bus to within a functional block.
BUS	**9**	Drawing set bus signal identifier.
BUSN	**115**	Set signal bus identifier. Same as BUS/9, except that will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
CDAT	**150**	Date of the circuit. This attribute defines the text as the date on which the design was last modified.
CENG	**153**	Circuit engineer. Thisattribute defines the text as the name of the project engineer for the complete design or circuit.
CNUM	**148**	Circuit number. This attribute defines the text as the number of the complete design or circuit.
COM	**0**	Indicates a comment field. The text field is passed on to post processors.
CONB	**26**	Connector bidirectional pin identifier. This attribute denotes a connector pin that serves as an input or an output.
CONI	**24**	Connector input pin identifier. This attribute denotes a connector pin that serves as an input only.

CONO	**25**	Connector output pin identifier. This attribute denotes a connector pin that serves as an output only.
CREL	**152**	Circuit release. This attribute defines the text as the release number of the complete design or circuit.
CREV	**149**	Circuit revision. This attribute defines the text as the revision of the complete design or circuit.
CRLT	**158**	This attribute is assigned to displayed text that is to be treated as circuit related layered text.
CRRL	**163**	Circuit related text field that is comment data on the drawing. Associated layered text is design related.
CTIT	**151**	Circuit title. This attribute defines the text as the title (name) of the complete design or circuit.
DATE	**54**	Drawing date. This attribute denotes the displayed text field as the date.
DENG	**154**	Drawing engineer. This attribute defines the text as the name of the engineer responsible for the drawing.
DNUM	**51**	Drawing number. This attribute denotes the displayed text field as the drawing number.
DPAG	**53**	Drawing page. This attribute denotes the displayed text field as the page number of the drawing.
DREV	**52**	Drawing revision. This attribute denotes the displayed text field as the revision level of the drawing.
DWLT	**159**	This attribute is assigned to displayed text that is to be treated as drawing related layered text.
DWRL	**129**	Drawing related text field that is comment data on the drawing. Any associated layered text is drawing related.
FILE	**8**	File name pointer. This attribute is assigned to the displayed name(s) assigned to a functional block in a hierarchical design structure. The name references a drawing file within the drawing system. The name is a printable text field. If layered text is associated with the field, the displayed text is treated as comment data. The layered text contains a list of the drawing files that represent the functional block.

FILN	**114**	File name pointer. Same as FILE/8, except that it will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
GATE	**155**	Multi-gate part (e.g. hex inverter, quad two-input AND, etc.). This attribute is assigned to the gate identifier of a multi-gate part.
GND	**100**	Symbol ground pin (signal ground) identifier. This attribute implies an implicit signal name GND is connected to this pin, and will be part of any signal net explicitly named GND.
GNDS	**157**	Universal ground signal identifier.
GNLS	**168**	Local ground signal identifier. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
LOC	**2**	Circuit location or reference designator, such as U11, C81, R22, etc. This attribute causes the location name to be unique within the design.
LOCL	**146**	Local location designator. This attribute causes the location name to be local to the drawing file. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
LOGC	**165**	Logical page connection. (Reserved for future products)
NULL	**128**	Indicates a comment field. The text field is not passed on to post processors.
PART	**3**	Part number (e.g. 74LS04) or part name (e.g. HEX INVERTER).
PIN	**1**	Symbol pin identifier. This attribute conveys no input or output information regarding the pin and is generally used in resistors, capacitors, and similar analog type of symbols.
PINB	**28**	Symbol bidirectional pin identifier. This attribute denotes a symbol pin that serves as an input or an output.
PINI	**23**	Symbol input pin identifier. This attribute denotes a pin that serves as an input only.

PINN	**103**	Symbol pin identifier. Same as PIN but default printability of the text field is off. See SGNU in Table 3-5.
PINO	**21**	Symbol output pin identifier. This attribute denotes a pin that serves as a TTL output only. This type of output cannot be an open-collector type and cannot be a wired-OR type.
PINT	**20**	Symbol output pin identifier for an output that can be switched to a high-impedance state.
PNAM	**169**	This attribute is assigned to displayed text field that is the logical pin name of a symbol, such as CLK, Q1, etc.

Note: If the pin identifier is alphabetic, the PNAM (pin name) property will be derived from the field that assigns the identifier. If the pin identifier is a number, the PNUM property will be derived from the field that assigns the pin identifier.

PNBT	**22**	Symbol bidirectional 3-state pin identifier. This attribute denotes a pin that serves as an input or an output, and can be switched to a high-impedance state.
PNLT	**162**	This attribute is assigned to displayed text that is to be treated as pin related layered text.
PNBO	**29**	Symbol open collector bidirectional pin identifier. This attribute denotes a pin that serves as an input or an output, and can also be switched to an open collector output.
PNOC	**27**	Symbol open collector pin identifier. This attribute denotes a connector pin that can be switched to an open collector output.
PNRL	**145**	Pin related text field that is comment data on the drawing. Associated layered text is pin related.
PNUM	**144**	This attribute is assigned to a displayed text field that is the physical pin number of a symbol.

Note: If the pin identifier is alphabetic, the PNAM (pin name) property will be derived from the field that assigns the identifier. If the pin identifier is a number, the PNUM property will be derived from the field that assigns the pin identifier.

PRLS	**167**	Local power signal identifier. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
PWRS	**156**	Universal power signal identifier.
SGLP	**166**	Local packaged signal name. Names a signal that uses a bus for connection purposes within a drawing sheet, but is not identified with the bus. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
SGLT	**160**	Displayed text field that is to be treated as signal related layered text.
SGNU	**104**	Universal signal identifier, similar to SIGU/14, except that will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
SGRL	**136**	Signal related text field that is treated as comment data on the drawing. Associated layered text is signal related.
SIG	**5**	Drawing set signal identifier (e.g. clock enable, IRQ, SRQ).
SIGB	**12**	Drawing set signal identifier for a bidirectional signal name. This attribute denotes an signal that goes to, or originates from, off the drawing.
SIGI	**10**	Input signal name. This attribute denotes an input that originates from off the drawing, such as from a connector.
SIGL	**147**	Local signal identifier; used to identify a signal as existing only within the drawing sheet. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
SIGN	**102**	Set signal identifier, similar to SIG/5, except that will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
SIGO	**11**	Drawing set signal identifier. This attribute denotes an output that goes off the drawing, such as to a connector.

SIGP	**164**	Packaged signal identifier. Names a signal that uses a bus for connection purposes, but is not identified with the bus.
SIGU	**14**	Universal signal identifier. This attribute should be assigned to signal name fields that are to be universal in scope.
SNAM	**130**	Set signal identifier. Same as SIG/5.
SPIN	**139**	Signal pin identifier.
STR	**7**	Component stress (e.g., 60vac, 0.25 watts)
SYLT	**161**	Assigned to displayed text that is to be treated as symbol related layered text.
SYRL	**138**	Symbol related text field that is treated as comment data on the drawing. Associated layered text is symbol related.
TITL	**50**	Drawing title. This attribute denotes the displayed text field as the title of the drawing.
TOL	**6**	Component tolerance (e.g., 5%).
VAL	**4**	Component value (e.g., 75 ohms, 0.002 microfarads).
VEE	**107**	Symbol power pin identifier. This attribute implies an implicit signal name VEE is connected to this pin, and will be part of any signal net explicitly named VEE.

Table C-2
Attributes Listed in Numerical Order

0	**COM**	Indicates a comment field. The text field is passed on to post processors.
1	**PIN**	Symbol pin identifier. This attribute conveys no input or output information regarding the pin and is generally used in resistors, capacitors, and similar analog type of symbols.
2	**LOC**	Circuit location or reference designator, such as U11, C81, R22, etc. This attribute causes the location name to be unique within the design.

3	**PART**	Part number (e.g. 74LS04) or part name (e.g. HEX INVERTER).
4	**VAL**	Component value (e.g., 75 ohms, 0.002 microfarads).
5	**SIG**	Drawing set signal identifier (e.g. clock enable, IRQ, SRQ).
6	**TOL**	Component tolerance (e.g., 5%).
7	**STR**	Component stress (e.g., 60vac, 0.25 watts)
8	**FILE**	File name pointer. This attribute is assigned to the displayed name(s) assigned to a functional block in a hierarchical design structure. The name references a drawing file within the drawing system. The name is a printable text field. If layered text is associated with the field, the displayed text is treated as comment data. The layered text contains a list of the drawing files that represent the functional block.
9	**BUS**	Drawing set bus signal identifier.
10	**SIGI**	Input signal name. This attribute denotes an input that originates from off the drawing, such as from a connector.
11	**SIGO**	Drawing set signal identifier. This attribute denotes an output that goes off the drawing, such as to a connector.
12	**SIGB**	Drawing set signal identifier for a bidirectional signal name. This attribute denotes an signal that goes to, or originates from, off the drawing.
14	**SIGU**	Universal signal identifier. This attribute should be assigned to signal name fields that are to be universal in scope.
20	**PINT**	Symbol output pin identifier for an output that can be switched to a high-impedance state.
21	**PINO**	Symbol output pin identifier. This attribute denotes a pin that serves as a TTL output only. This type of output cannot be an open-collector type and cannot be a wired-OR type.
22	**PNBT**	Symbol bidirectional 3-state pin identifier. This attribute denotes a pin that serves as an input or an output, and can be switched to a high-impedance state.
23	**PINI**	Symbol input pin identifier. This attribute denotes a pin that serves as an input only.

24	CONI	Connector input pin identifier. This attribute denotes a connector pin that serves as an input only.
25	CONO	Connector output pin identifier. This attribute denotes a connector pin that serves as an output only.
26	CONB	Connector bidirectional pin identifier. This attribute denotes a connector pin that serves as an input or an output.
27	PNOC	Symbol open collector pin identifier. This attribute denotes a connector pin that can be switched to an open collector output.
28	PINB	Symbol bidirectional pin identifier. This attribute denotes a symbol pin that serves as an input or an output.
29	PNBO	Symbol open collector bidirectional pin identifier. This attribute denotes a pin that serves as an input or an output, and can also be switched to an open collector output.
50	TITL	Drawing title. This attribute denotes the displayed text field as the title of the drawing.
51	DNUM	Drawing number. This attribute denotes the displayed text field as the drawing number.
52	DREV	Drawing revision. This attribute denotes the displayed text field as the revision level of the drawing.
53	DPAG	Drawing page. This attribute denotes the displayed text field as the page number of the drawing.
54	DATE	Drawing date. This attribute denotes the displayed text field as the date.
100	GND	Symbol ground pin (signal ground) identifier. This attribute implies an implicit signal name GND is connected to this pin, and will be part of any signal net explicitly named GND.
101	+5V	Symbol power pin (+5V) identifier. This attribute implies an implicit signal name +5V is connected to this pin, and will be part of any signal net explicitly named +5V.

102	SIGN	Set signal identifier, similar to SIG/5, except that will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
103	PINN	Symbol pin identifier. Same as PIN but default printability of the text field is off. See SGNU in Table 3-5.
104	SGNU	Universal signal identifier, similar to SIGU/14, except that will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
105	+12V	Symbol power pin (+12V) identifier. Used with emitter- coupled logic. This attribute implies an implicit signal name +12V is connected to this pin, and will be part of any signal net explicitly named +12V.
106	-12V	Symbol Power pin (-12V) identifier. Used with emitter- coupled logic. This attribute implies an implicit signal name -12V is connected to this pin, and will be part of any signal net explicitly named -12V.
107	VEE	Symbol power pin identifier. This attribute implies an implicit signal name VEE is connected to this pin, and will be part of any signal net explicitly named VEE.
114	FILN	File name pointer. Same as FILE/8, except that it will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
115	BUSN	Set signal bus identifier. Same as BUS/9, except that will not print as part of the drawing unless you specifically command the editor to do so (used with older editors to selectively turn off the print capability).
128	NULL	Indicates a comment field. The text field is not passed on to post processors.
129	DWRL	Drawing related text field that is comment data on the drawing. Any associated layered text is drawing related.
130	SNAM	Set signal identifier. Same as SIG/5.
132	BNAM	Set bus identifier. Same as BUS/9.

136	**SGRL**	Signal related text field that is treated as comment data on the drawing. Associated layered text is signal related.
138	**SYRL**	Symbol related text field that is treated as comment data on the drawing. Associated layered text is symbol related.
139	**SPIN**	Signal pin identifier.
141	**BPIN**	Bus pin identifier. Used to pass a bus to within a functional block.
144	**PNUM**	This attribute is assigned to a displayed text field that is the physical pin number of a symbol.
145	**PNRL**	Pin related text field that is comment data on the drawing. Associated layered text is pin related.
146	**LOCL**	Local location designator. This attribute causes the location name to be local to the drawing file. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
147	**SIGL**	Local signal identifier; used to identify a signal as existing only within the drawing sheet. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
148	**CNUM**	Circuit number. This attribute defines the text as the number of the complete design or circuit.
149	**CREV**	Circuit revision. This attribute defines the text as the revision of the complete design or circuit.
150	**CDAT**	Date of the circuit. This attribute defines the text as the date on which the design was last modified.
151	**CTIT**	Circuit title. This attribute defines the text as the title (name) of the complete design or circuit.
152	**CREL**	Circuit release. This attribute defines the text as the release number of the complete design or circuit.
153	**CENG**	Circuit engineer. This attribute defines the text as the name of the project engineer for the complete design or circuit.

154	**DENG**	Drawing engineer. This attribute defines the text as the name of the engineer responsible for the drawing.
155	**GATE**	Multi-gate part (e.g. hex inverter, quad two-input AND, etc.). This attribute is assigned to the gate identifier of a multi-gate part.
156	**PWRS**	Universal power signal identifier.
157	**GNDS**	Universal ground signal identifier.
158	**CRLT**	This attribute is assigned to displayed text that is to be treated as circuit related layered text.
159	**DWLT**	This attribute is assigned to displayed text that is to be treated as drawing related layered text.
160	**SGLT**	Displayed text field that is to be treated as signal related layered text.
161	**SYLT**	Assigned to displayed text that is to be treated as symbol related layered text.
162	**PNLT**	This attribute is assigned to displayed text that is to be treated as pin related layered text.
163	**CRRL**	Circuit related text field that is comment data on the drawing. Associated layered text is design related.
164	**SIGP**	Packaged signal identifier. Names a signal that uses a bus for connection purposes, but is not identified with the bus.
165	**LOGC**	Logical page connection. (Reserved for future products)
166	**SGLP**	Local packaged signal name. Names a signal that uses a bus for connection purposes within a drawing sheet, but is not identified with the bus. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
167	**PRLS**	Local power signal identifier. In a post processor listing, a text field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.
168	**GNLS**	Local ground signal identifier. In a post processor listing, a text /field with this attribute is appended with ***ref, where ref is the unique drawing occurence number.

169	**PNAM**	This attribute is assigned to displayed text field that is the logical pin name of a symbol, such as CLK, Q1, etc. (See note below.)

Properties

Properties identify a single characteristic about a schematic element. They identify and describe lines, symbols, and their components. They can be used to imply connections. They are assigned to schematic elements through alphanumeric fields.

This properties listed in Tables C-3 through C-8 are the reserved property names recognized by the DASH Post post processing tools. Properties are grouped according the type of schematic element they are associated with.

Table C-3
Circuit Related Properties

CNUM	The CNUM property is the circuit number. It can be derived for the CNUM attribute or from text.
CREV	The CREV property is the circuit revision. It can be derived for the CREV attribute or from text.
CDAT	The CDAT property is the circuit date. It can be derived for the CDAT attribute or from text.
CTIT	The CTIT property is the circuit title. It can be derived for the CTIT attribute or from text.
CREL	The CREL property is the circuit release. It can be derived for the CREL attribute or from text.
CENG	The CENG property is the circuit engineer. It can be derived for the CENG attribute or from text.

Table C-4
Drawing Related Properties

TITL	The TITL property is the drawing title. It can be derived for the TITL attribute or from text.
DNUM	The DNUM property is the drawing number. It can be derived for the DNUM attribute or from text.
DREV	The DREV property is the drawing revision. It can be derived for the DREV attribute or from text.
DPAG	The DPAG property is the drawing page. It can be derived for the DPAG attribute or from text.
DATE	The DATE property is the drawing date. It can be derived for the DATE attribute or from text.
DENG	The DENG property is the drawing engineer. It can be derived for the DENG attribute or from text.

Table C-5
Symbol Related Properties

LOC	The LOC property is the circuit location or reference designator. It can be derived from the LOC or LOCL attribute or from text.
PART	The PART property is the part number or part name. It can be derived from the PART attribute or from text.
VAL	The VAL property is the component tolerance. It can be derived from the VAL attribute or from text.
STR	The STR property is the component stress. It can be derived from the STR attribute or from text.
GATE	The GATE property is a multi-gate part designator. It can be derived from the GATE attribute or from text.

The pin related attribute mnemonics **PNUM** and **PNAM** are identical with the reserved pin properties and can be used in the property assignment statement to assign tool-specific or non-displayed property information.

Table C-6
Pin Related Properties

PNAM	The PNAM property is the alpha numeric name of a pin. It maybe derived from the display text of a alphanumeric field with a pin attribute or from layered text. If the pin display text is alphanumeric and the PNAM property has not be specified, the PNAM property value will be the pin display text.
PNUM	The PNUM property is the pin number. It maybe derived form the display text of a alphanumeric field with a pin attribute or from layered text. If the pin display text is numeric and the PNUM property has not be specified, the PNUM property value will be the pin display text.

TYPE The TYPE property specifies the pin characteristics. Values for the TYPE property are predefined by DASH and derived from the pin attribute.

This property and the associated values are not user-specifiable.

The following values are possible:

Attributes	Values	Definitions
PIN,PINN,SPIN	P	simple pin
PINT	T	tristate
PINI	I	input
PINO	O	output
PNBT	BT	bidirectional tristate
CONI	CI	input connector
CONO	CO	output connector
CONB	CB	bidirectional connector
PNOC	OC	open collector
PINB	B	bidirectional
PNBO	BO	bidirectional open collector
GND	GND	ground pin
+5V	+5V	+5V power pin
+12V	+12V	+12V power pin
-12V	-12V	-12V power pin
VEE	VEE	VEE power pin
BUSP	BP	bus pin

The reserved signal property NAME can be used in the property assignment statement to assign a tool-specific signal name.

Table C-7
Signal Related Properties

NAME The NAME property specifies the signal NAME. If a NAME property has not be assigned, a value for the NAME property is derived. Name values may come from display text and signal attributes; or assigned by Dash through implicit connection of pins with attributes +5V,-5V,+12V,-12V,VEE, or GND; or though Dash assigned names for unnamed signals.

Note: The property NAME cannot be used in property assignment statements to alias signals. The NAME value is used in post-processor output only, not in determining connectivity.

TYPE The TYPE property specifies signal characteristics. Values for the TYPE property are predefined by DASH and derived from the signal attribute.

This property and the associated values are not user-specifiable.

The following values are possible.

Attributes	Values	Definitions
SIG,SIGR,SIGU SIGN,SGNU,SNAM, SIGL,SIGP,SGLP, BUS,BUSN,BNUM	S	signal or bus
SIGI	I	input
SIGO	O	output
SIGB	B	bidirectional
PWRS,PRLS	PWR	power signal
GNDS,GDLS	GND	ground signal

SCOPE The SCOPE property specifies the range of connectivity of a signal. Values for the SCOPE property are predefined by DASH and derived from the attribute.

This property and its associated values are not user-specifiable.

The following values are possible:

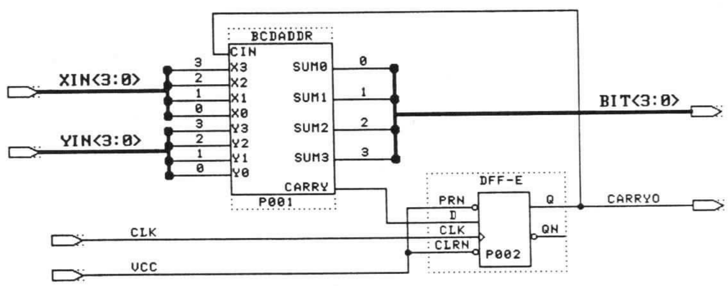
Attributes	Values	Definitions
SIGU,SGNU,PWRS, GNDS	U	universal
SIG,SIGR,BUS, BUSN,SIGN,BNAM, SIGP,SIGI,SIGO, SIGB	S	drawing set
SIGL,SGLP,PRLS, GDLS	L	drawing

D Example of Functional Block Modularity

The following series of FutureNet drawings serves to illustrate the concept of functional blocks as they might be integrated into a larger, hierarchical design.

In Figure D–1, a 4-bit adder with carry is functionally depicted. Besides the CLK (clock) and Vcc, the inputs to this adder are two buses (XIN and YIN), each containing four signals. The binary count of each four-signal bus is summed in a functional block called BCDADDR and the sum is output, again in binary form, on the output bus (BIT). Any carry generated as a result of the summing operation is passed to an edge-triggered D flip-flop shown in the drawing as DFF-E. The carry signal, CARRYO, is the output of DFF-E.

Figure D-1
Adder



The BCDADDR symbol is an example of a functional block. The inputs and outputs are shown, but the logic is not evident. Figure D–2 shows a schematic of what is contained within the BCDADDR functional block of ADDER.DWG. Note that within BCDADDR still more functional blocks are found. There are some discrete logic components at this level but most of the logic is still represented by functional blocks. Of particular significance are the HALFADDR and FULLADDR blocks; they will be seen again and examined in detail at a deeper level of the design.

Figure D-2
BCD Adder

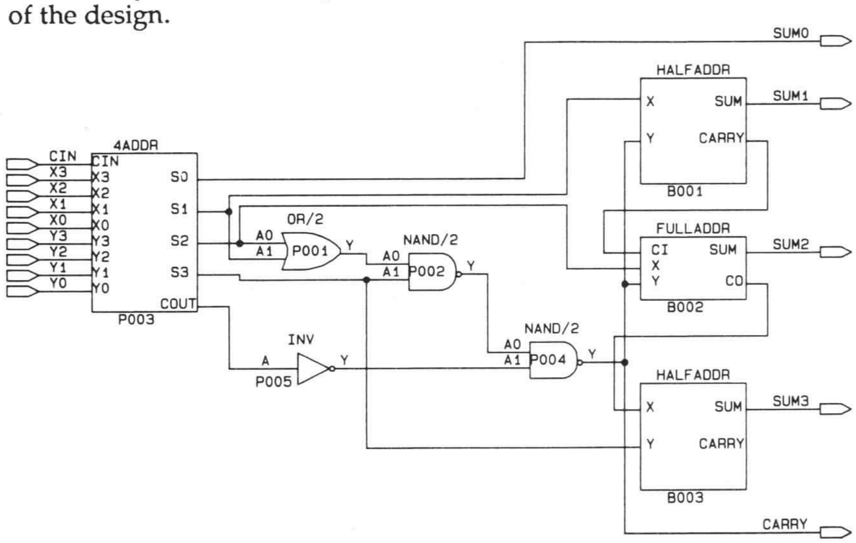
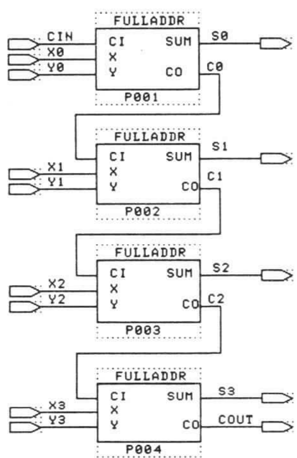


Figure D–3 reveals the functional contents of the 4ADDR functional block in BCDADDR.DWG. As you can see, 4ADDR consists of four identical full adders, each represented as functional blocks named FULLADDR. These FULLADDRs are the same as the one seen at the previous level.

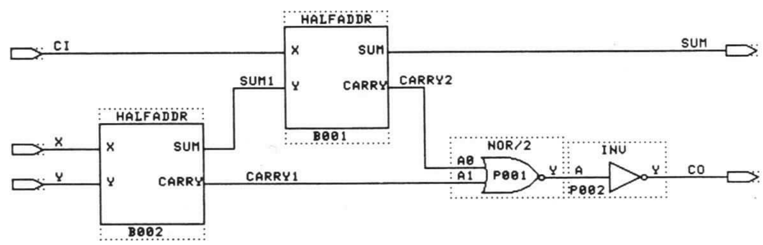
Figure D-3
4-bit Adder



Input signals, output signals and interconnecting symbols are clearly labeled. You can follow the logic of each full adder as each pair of bits is summed and carries, if any, are passed on to the next successive stage. The full adders, however, are still represented by functional blocks.

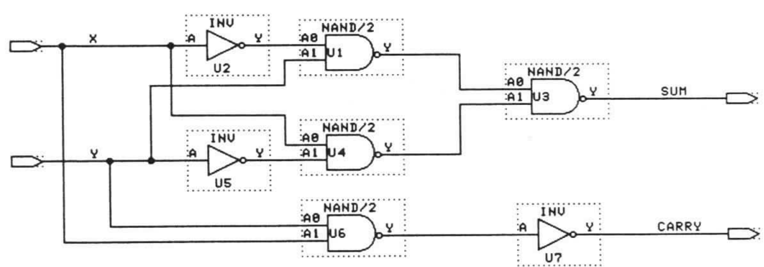
Figure D–4 shows the schematic of the elements that comprise a FULLADDR. The full adder consists of two half adders and associated logic. Again inputs and outputs are labeled and other interconnection circuitry is illustrated. The half adders depicted by functional blocks in Figure D–4 are identical to the half adders shown at the second level of this design (shown in Figure D–2). There are no restrictions on when and where functional blocks may be used within a design.

Figure D-4
Full Adder



One final level remains. The half adders that make up the full adder are illustrated in Figure D–5. At this level, all of the discrete gates are shown. This is the lowest level of detail for this drawing set. It is important to note that half adders were used at two levels of this design. Both levels used a reference to a single, lower level drawing that showed the complete circuitry in detail. Full adders were also used in this manner, referencing the lower level half adder circuitry as needed.

Figure D-5
Half Adder



E Example of Separate Sheet Modularity

The FutureNet drawings on the next several pages illustrate the concept of separate sheet modularity. The design shown is a multifunction serial/parallel I/O board. All drawing sheets have been created at the same level; that is, there are no subordinate drawing levels. By examining each sheet, you will note that each drawing sheet contains a separate functional area of the overall design.

Figure E–1 shows the schematic for the computer bus connection and associated circuitry.

Figure E-1
Multifunction Serial/Parallel I/O Board - Sheet 1

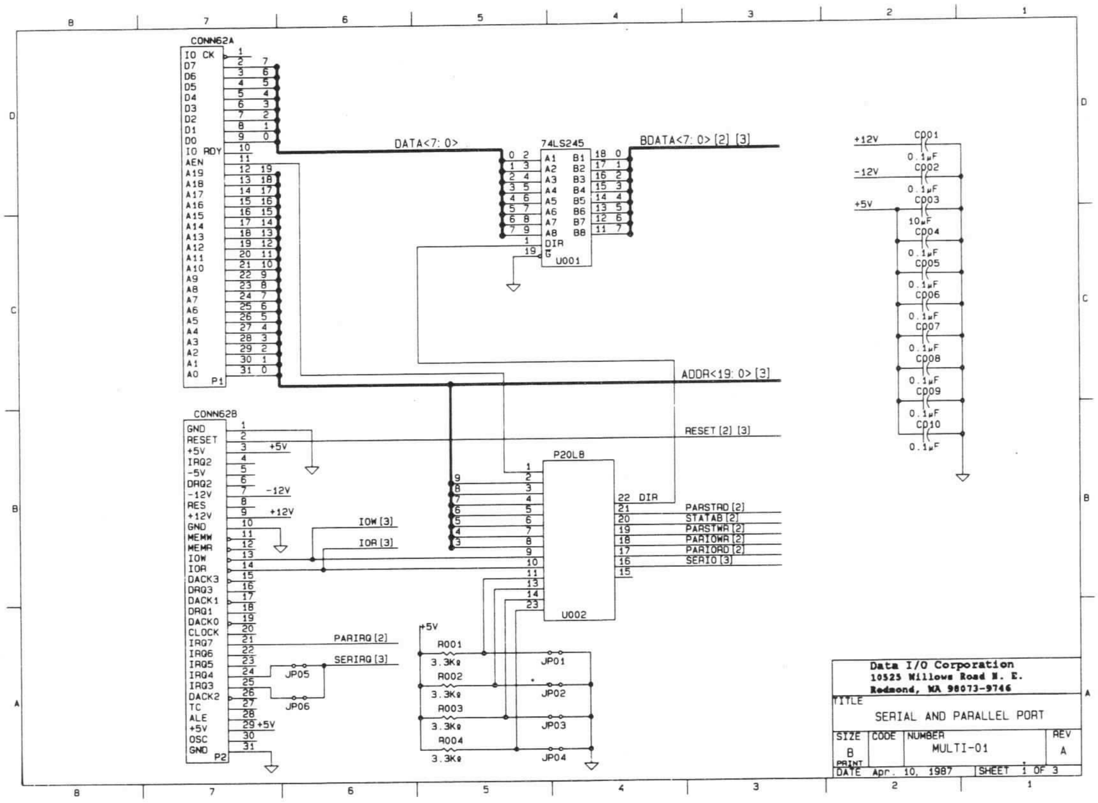


Figure E–2 depicts the parallel interface circuitry for the multifunction I/O board.

Figure E-2
Multifunction Serial/Parallel I/O Board - Sheet 2

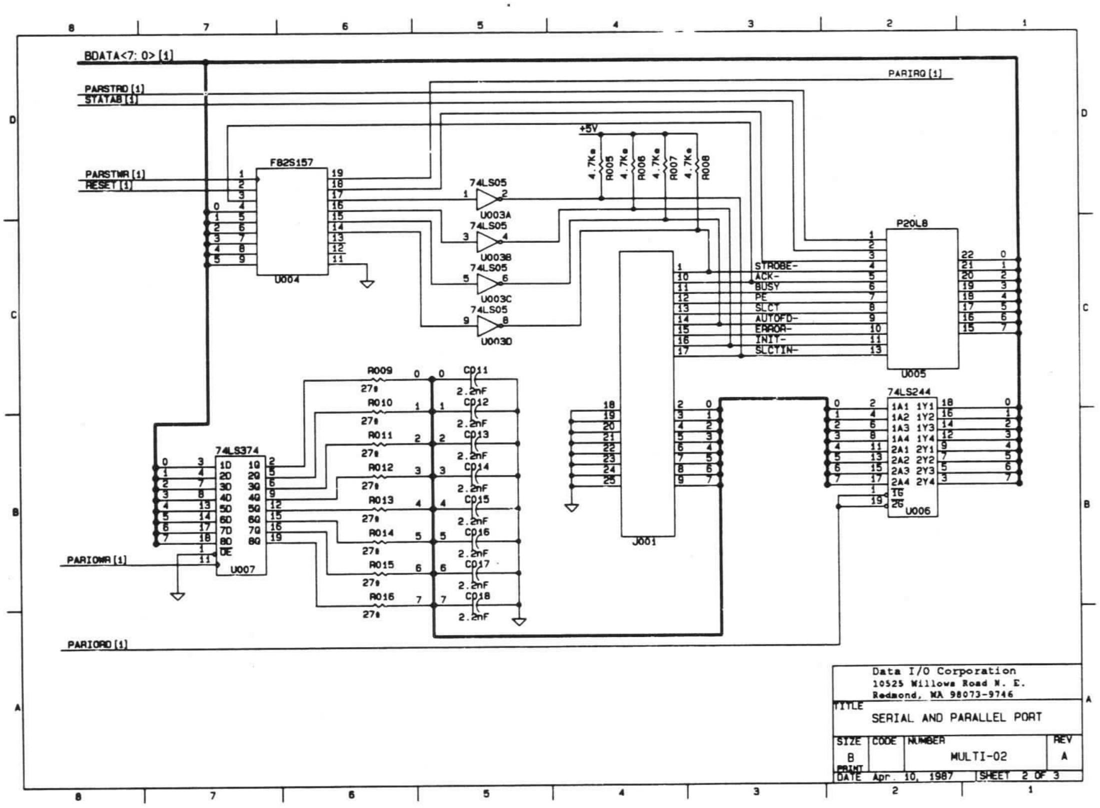
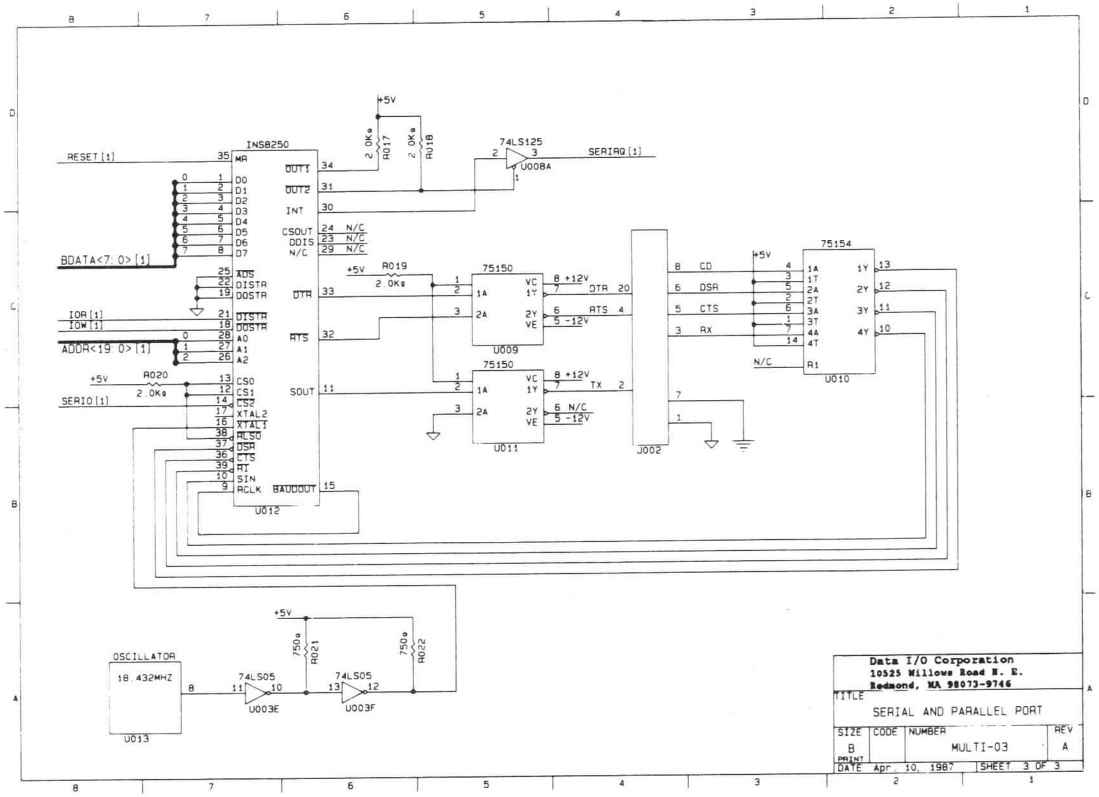


Figure E–3 contains the details of the serial interface circuitry.

Figure E-3
Multifunction Serial/Parrallel I/O Board - Sheet 3



F Conversion of Existing Drawings to New Format

Existing drawings in earlier DASH/FutureNet formats are automatically converted when they are loaded into the new software. For conversion to proceed smoothly, check the following items:

- An FNPRO.CMD file must be processed (this is normally done on startup).
- Use the LOAD command (with a complete path and filename if the drawing is not in the default directory).
- Use the SAVE command to save the drawing in converted format. Note that once the existing files are converted and saved to disk, you cannot use them with earlier versions of the software. You may want to save the converted file under a different name.

As the file is loaded for dynamic conversion, you will see the following message:

```
Analyzing graphics to supply points of effect
```

This message is only seen during conversion of older version FutureNet drawings to the current format. Once the drawing has been loaded and displayed, it has been converted. You may edit the drawing or save it in the new format.

Note: When saving files, FutureNet changes the file extension of the original drawing file to .BAK. If you accidentally copy over your original drawing file, you can rename the converted filename.dwg to a different name or rename filename.bak to filename.dwg

DASH-4 Point-of-Effect Conversion

Because of differences in the way that *point-of-effect* is handled between DASH-2 or DASH-3, and DASH-4, during conversion some signal names may give ambiguous information about the real point-of-effect with respect to associated symbols.

For example, DASH-2/3 provides that a signal outside a symbol, but adjoining a symbol boundary and touching a pin field, is connected electrically to the pin. Releases of DASH-4 prior to version 4.00 did NOT necessarily recognize the connectivity. They instead looked for lines (wires) within 10 display units. Because DASH-4 supports placement of the point-of-effect not immediately adjacent to the alphanumeric field, it was possible for the conversion to erroneously attach a line that was not previously connected. If no wire was found within 10 display units, the point-of-effect for the alphanumeric field would be at its justification point, that is, the point from where the text was justified, either right, center, or left (using the DASH-4 'J command). In any case, both of these possible alternatives would be erroneous.

Figure F–1, below, illustrates the problem. The symbol and alphanumeric field boundaries are shown for clarity. Notice that both Enable 1 and Enable 2 are clearly abutted to pin 1 and 19 respectively, of the 74LS244 symbol. It is clear that they are NOT associated, in any way, with the signal lines around them.

Figure F-1
DASH-3 Point-of-effect

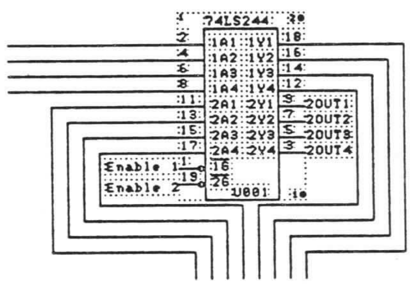


Figure F–2, illustrates what happened before the problem was solved. After conversion to DASH-4, the points of effect for both Enable 1 and Enable 2 were affixed to the signal coming from pin 17 of the 74LS244 symbol. This is obviously NOT intended.

The Solution

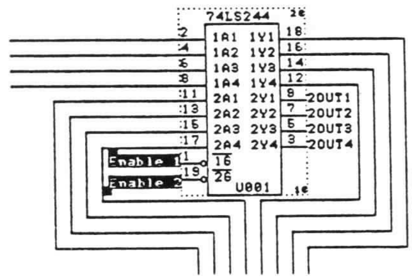
In Conventions

The following conventions describe the current resolution of possible signal ambiguity when converting from earlier format drawings to DASH-4 or later format.

For any given signal external to a symbol:

- If the signal touches a pin, but not a line, the point-of-effect is placed at the pin's point-of-effect.

Figure F-2
DASH-4 Point-of-effect Conversion Example



- If the signal touches a line, but not a pin, the point-of-effect is placed on the line.
- If the signal touches neither a line nor a pin, the point-of-effect is placed at the justification point. Because this results in an unassigned signal name, you are informed of an unassigned signal.
- If the signal touches both a line and a pin, and they are not connected to each other, the point-of-effect is placed at the justification point. You are informed of the ambiguity and the drawing must be edited by hand to properly place the point-of-effect.
- If a signal touches more than one pin of a symbol, the point-of-effect defaults to the lower, left-most pin that the signal touches.
- If a signal touches more than one pin and the pins are on **two different symbols**, you are informed of the possible ambiguity, and told where (X,Y coordinates) to look.

In FutureNet

FutureNet informs you of the occurrence of possible point-of-effect ambiguities during file conversion. If there is at least one possible ambiguity, you receive the following message on the FutureNet command line.

```
Possible point-of-effect ambiguity--See file drawing.poe

--Press any key to continue
```

drawing is the same unique name as the original drawing file. The file *drawing*.poe contains a listing of all possible problems. Explicit X,Y cursor position information is provided so you know where to look for the possible problem.

A typical *drawing*.poe file follows:

```
Possible point of effect ambiguities in signal(s) at:

59, 12

63, 15

(Unassigned signal at the next location listed below)

83, 36

77, 99

99, 107

129, 56
```

Index

(pound)

' (apostrophe)

. (period)

/ (slash)

[(bracket)

A

B

C

F

G

H

I

J

K

Q

R

S

FutureNet®

Schematic Designer

Command Reference

DATA I/O
Corporation

096-0086-003

August 1991 096-0086-003

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA
(206) 881-6444

Acknowledgments:

Data I/O and FutureNet are registered trademarks of Data I/O Corporation.
Sun is a registered trademark of Sun Microsystems, Inc.

Table of Contents

1 *Introduction*

The *Command Reference* manual contains descriptions in alphanumeric order of all the commands used in FutureNet®. A comment on the first line of the "Remarks" section indicates commands not supported by FutureNet OEM products.

This manual provides a quick reference to FutureNet commands for users who already understand how FutureNet operates. You should be familiar with the *FutureNet User Manual* before using this reference.

The Command Reference Card found in the inside cover of this binder gives short descriptions of each command arranged by type of commands.

The command reference is also available online by using the HELP command.

Types of Commands

Below is a list of the command types available in FutureNet; short descriptions of their purpose follow the listing. The "Command Reference" chapter lists all commands in alphanumeric order, and not by the type of command; however, some types of commands can be distinguished by the first symbol in the command. For example, all commands beginning with a slash (/) are line drawing commands, and all commands beginning with a left bracket ([) are area editing commands. The descriptions below give the special symbol, if any, that distinguishes that type of command. The Command Reference Card lists all commands by type of command.

Most commands are available from either the menu or the command line. The types of commands available in FutureNet are

- System commands
- Session commands
- Drawing file commands
- Cursor commands
- Line drawing commands
- Alphanumeric commands
- Area editing commands
- Symbol management commands
- Command file command
- Symbol definition commands

System Commands

System commands are operating system commands (that is, DOS or UNIX commands) that can be entered from within FutureNet.

Session Commands

Session commands control FutureNet operation during an editing session, such as AUTOSAVE, COLOR, HELPFILE, PRINTOPT, and PROFILE.

Drawing File Commands

Drawing file commands allow you to manipulate or change an entire drawing and perform a few frequently-used operating system file commands without exiting FutureNet.

The drawing file commands allow you to load an existing drawing, save and print a drawing, and erase the drawing currently displayed.

Also included are commands that allow you to specify the size of the drawing and renumber all of the symbols loaded into a drawing, as well as commands that allow you to list the files in the current directory, change directories, or delete a file from the hard disk.

Some drawing file commands begin with a pound sign (#).

Cursor Commands

The cursor movement commands described in this reference are entered from the keyboard and are valid in these modes: tag and drag, line drawing, area definition, and menu and fast drawing modes. These cursor movement commands do not work in alphanumeric entry and Symbol Definition Modes, or when the menu is displayed.

Line Drawing Commands (/)

Line drawing commands are used to draw and erase lines, add graphic items (arrows), select the line type, and invoke various functions like PINSNAP and CONNECT.

All line drawing commands begin with a slash (/).

Alphanumeric Commands (')

Alphanumeric fields are used to name signals, label parts, number and name pins, add comments, and more. In conjunction with attributes and properties, alphanumeric fields provide the information used by the post processors.

Alphanumeric fields can be manipulated using commands and edited using a special alphanumeric mode.

Alphanumeric mode is entered and exited using **Esc** . A text cursor appears at the location of the graphics cursor when alphanumeric mode is entered. ALPH appears in the MODE status field.

Most alphanumeric commands begin with an apostrophe (').

Area Editing Commands ([)

An area is a user-defined rectangular region of the drawing that can be any size and include any number of symbols, lines, and alphanumeric fields. Once defined, an area can be tagged and moved, copied, erased, saved, and loaded. Portions of a drawing can be saved as an area and then loaded later (in the same or a different drawing session).

When an area is defined, the region of the drawing in the area is outlined by a dotted line boundary.

Symbols and alphanumeric fields are considered part of an area only if they are completely inside the area boundary. Lines are broken at the area boundary.

All area editing commands begin with a left bracket ([).

Symbol Management Commands (.)

Symbol management commands include commands to load an existing symbol from the library into a drawing, to manipulate symbols in the drawing, and to create and edit simple block symbols.

These commands also include how to specify a library for use and how to save or delete entries in that library.

Except for the * version of .L (load a symbol command), all symbol editing commands begin with a dot (.).

Command File Commands

Several commands exist to support using command files with FutureNet. These commands are used to run command files, modify the visual effect of commands on the screen and control of the command file session.

A command file is composed of FutureNet editing commands. A command file can be created with a text editor and can contain one command per line or more than one command per line provided the commands on the same line are separated by semicolons (;). Each line must end with a complete command (that is, a command cannot be broken by a carriage return) and no line can exceed 256 characters, although we suggest keeping the lines to 80 characters or less. When a command file is read, the commands are run in the order listed.

Most of the commands described in this manual can be entered in some form into command files.

Function keys cannot be used in command files. Use the command equivalent instead.

Running commands automatically with a command file also works in Symbol Definition Mode. See the section on symbol definition commands for details.

Symbol Definition Commands

Symbol Definition Mode is used to create new symbols and modify existing symbols by precisely specifying each line and element of the symbol. This special mode uses a different set of commands than the other FutureNet modes. In the "Command Reference" chapter, commands used for symbol definition are indicated with a comment: Symbol Definition Mode.

Symbol definition editing commands are used to enter and exit Symbol Definition Mode and edit symbol definition instruction lists. The editing commands available in Symbol Definition Mode allow you to define the size of a symbol cell, and replace, delete, or insert symbol definition instructions in the list that defines a symbol.

Symbol definition commands are not supported in FutureNet OEM products.

2 *Command Reference*

This chapter contains all of the FutureNet commands in alphanumeric order. The reference card contains a listing of commands by type.

! — Enter Operating System Command

Format

! [*command*]

Purpose

Run an operating system command, or (on PC only) enter a DOS shell to enter commands.

Remarks

The ! command enters the command given, or temporarily leaves FutureNet to create a DOS shell if no command is given.

This command can also be used in the Symbol Definition Mode.

Note: *For PC systems, if there isn't enough disk space to operate the DOS shell, you will be returned to FutureNet. FutureNet requires about 500KB of disk space to start a DOS shell. (No error message is given when you are returned to FutureNet.)*

Note: *You cannot run FutureNet (fn command) from within a DOS shell called from FutureNet; you can lose files and autologging.*

Related Commands

DOS

.- .> .= .-O .>O — Add/Replace Pin Stub

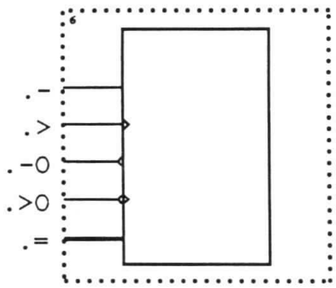
Format

.- | .> | .= | .-O | .>O

Purpose

Add or replace the specified pin stub at the edge of a symbol outline. The pin stub commands are intended to be used in symbols created with the .A, .B and .F commands.

Remarks



.-	Creates a pin stub.
.>	Creates a pin stub with a clock designation.
.=	Creates a bus pin stub.
.-O	Creates a pin stub with an inversion bubble.
.>O	Creates a clock pin stub with an inversion bubble.

The graphics cursor must be positioned between the symbol cell and the symbol block outline where the pin stub is to be drawn.

The stub is drawn from the symbol cell to the symbol cell boundary at the location of the graphics cursor.

If a stub is already present at the graphics cursor position, it is replaced by the specified stub.

If the cursor is not in a symbol boundary, then the command is ignored and the system displays

```
Cursor not in symbol
```

The error message

```
Block symbol bit not set
```

means that the special identifier flag for block symbols is not set. Using the symbol editor on a block symbol will clear the flag. Use the .SBS command inside the symbol editor to set the flag.

The error message

```
Not a block symbol
```

means that the symbol was created in such a way that it was never intended to have block symbol pin stubs. You will be unable to add pins stubs to this symbol. The .SBS command will be unable to convert this symbol to a block symbol.

Related Commands

.-AI
.A
.B
.F
.SBS

.-AI .-AO .>A .<A — Add/Replace IEC/ANSI Pin Stub

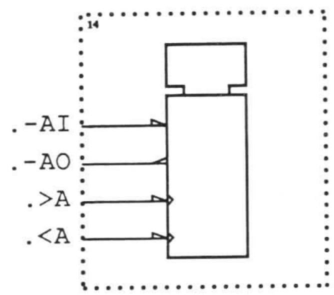
Format

.-AI | .-AO | .>A | .<A

Purpose

Add or replace the specified pin stub to the left or right side of IEC/ANSI symbol outline. The pin stub commands are intended to be used in symbols created with the .A, .B and .F commands.

Remarks



.-AI	Creates an input pin stub.
.-AO	Creates an output pin stub.
.>A | .<A	Creates a pin stub with a clock designator.

The graphics cursor must be positioned between the symbol cell and the symbol block outline on the left or right side of the symbol.

The stub is drawn from the symbol cell to the symbol cell boundary at the location of the graphics cursor.

If a stub is already present at the graphics cursor position, it is replaced by the specified stub.

If the cursor is not in a symbol boundary, then the command is ignored and the system displays

```
Cursor not in symbol
```

No IEC/ANSI pin stubs can be added at the corner, nor top or bottom, of a symbol. If the graphics cursor is in one of these areas, or inside the symbol block outline, the command is ignored and the system displays

```
Cursor not on symbol outline
```

The error message

```
Block symbol bit not set
```

means that the special identifier flag for block symbols is not set. Using the symbol editor on a block symbol will clear the flag. Use the .SBS command inside the symbol editor to set the flag.

The error message

```
Not a block symbol
```

means that the symbol was created in such a way that it was never intended to have block symbol pin stubs. You will be unable to add pins stubs to this symbol. The .SBS command will be unable to convert this symbol to a block symbol.

'0...'7 — Select Font

Format

'0 | '1 | '2 | '3 | '4 | '5 | '6 | '7

Purpose

Select font.

Remarks

The name of the current font is displayed in the ALPH status field.

When entering text between lines, either horizontally or vertically, leave sufficient room between the lines to accommodate the size font being used.

'1	A a	'3	A a	'5	A a	'7	A A
'2	A a	'4	A a	'6	A a		

'0 Cycle through the available font sizes. Each time the command is given, the font size will be changed to the next higher font number. Font 7 wraps to font 1.

'1 2x2 display unit font

'2 2x3 display unit font

'3 2x4 display unit font

'7 2x2 display unit font.

For font 7, all characters are uppercase. However, any characters entered without either the Shift key or the Caps Lock key on will be underlined to indicate they were entered in lowercase. If the 'CH F command is used to change these characters to another font having a lowercase, the characters will appear as lowercase characters in the new font.

Note: The following fonts, Fonts '4 through '6, are intended for use when conforming to MIL-D-1000 specifications.

'4 3x5 display unit font.

'5 5x6 display unit font.

'6 6x7 display unit font.

Fonts cannot be changed while in alphanumeric mode.

The font of an existing alphanumeric field can be changed using the 'CH F command.

Related Commands

'CH F

/0.../10 — Select Line Type

Format

/0 | /1 | /2 | /3 | /4 | /5 | /6 | /7 | /8 | /9 | /10

Purpose

Select the style of line to be used when lines are drawn.

Remarks

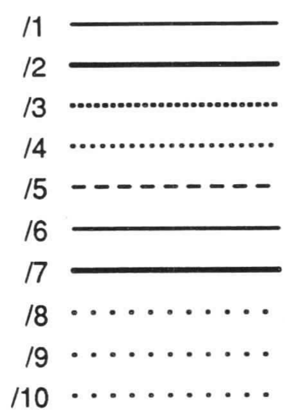
The name and sample of the current line type are displayed in the LINE status field.

/0 — Cycles through the available selections (/1 through /10). Line type 10 wraps to line type 1.

/1 — Selects wires. This is the default value when FutureNet is started. /1 lines can be rubberbanded.

/2 — Selects bus lines, which are the thickness of three wires. /2 lines can be rubberbanded.

/3 - /10 — Selects a variety of dashed or dotted lines. These lines have no electrical significance and cannot be used to establish connectivity. They are available as graphics only and can be used as comment lines.



Note: Although line type /6 and /7 look the same as /1 and /2, they are treated differently by post processing. Line types /1 and /2 are treated as electrically significant, but /6 and /7 are for graphics only.

? — Read Custom Menus

Format

? [["]help_string["]]

Purpose

Access custom menus and help screens.

Remarks

The **?** command provides access to custom menus and help screens that are created with FutureNet and the **HELPSAVE** command. An example of a custom menu library is supplied with FutureNet (SAMPLE.MNU). Refer to the *FutureNet User Manual* for more information on creating custom menus and help screens.

Note: *This command does not provide access to the online command reference. Use the HELP command to access the online command reference.*

If a "*help_string*" is not entered, then the default menu, "?HELP" is used. If it does not exist, then following error message appears on the status line:

```
Symbol not found
```

If a "*help_string*" is included in the "?" command, then the the help symbol "?help_string" is loaded if it is in either the update library or one of the reference libraries (see .LIB or LIB commands).

If the "?help_string" symbol does not exist in an open library, then the symbol "help_string" is used. If neither symbol is found then the command you will receive the following error message:

```
Symbol not found
```

If the SAMPLE.MNU library is loaded then the sample custom menu can be used.

Note that any symbol can be loaded as a help screen.

You can choose a custom menu selection the same way the standard FutureNet menu selections are chosen, by pressing the left mouse button or the ↵ key.

Exit a custom menu by pressing the right or middle mouse button or the Esc key.

Move in the menus by moving the mouse or using the arrow keys. All other keys on the keyboard are disabled during custom menu mode.

Related Commands

HELP
HELPSAVE
LIB
.LIB
.DIR

'A — Set Attribute for New Alphanumeric Field

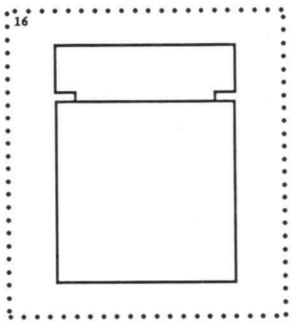
Format

'A *name* | *number*

Purpose

Set the default attribute to be assigned to new alphanumeric fields.

Remarks



Specify the default attribute type using an attribute name or number. Attribute names and numbers are explained in the chapter "Understanding FutureNet" in the *FutureNet User Manual.*

Until the system default attribute type is changed using the 'A or 'S commands, the system default of COM 0 will be assigned.

The attribute specified with 'A will be assigned to future alphanumeric fields. It does not affect the attribute assignments of existing fields.

When the cursor is not on an existing alphanumeric field, the attribute specified with the 'A command is shown in the ATTR status field. Both the attribute's mnemonic and numeric values are displayed.

If 'A is entered without specifying an attribute, the screen will display a menu of available attributes. Use the mouse to select an attribute on this menu.

To change the attribute of an existing field, use 'S or 'CH A.

Note: The 'D command can be used to display the numeric attributes of all existing fields in the drawing. Another way to check the attribute assignment for an alphanumeric field is to place the graphics cursor on the field and check the value in the ATTR status field. The ALPH status field will change from NEW to OLD or ONP to verify that you are in an existing field.

Related Commands

'CH A
'D
'S

.A — Create IEC/ANSI Symbol Outline

Format

.A *w,hh,hr[,ws[,hs]]*

Purpose

Create an IEC/ANSI block symbol outline with a "hat."

Remarks

Not supported by FutureNet OEM products.

w	The width of the rectangle and hat in display units.
hh	The height of the hat.
hr	The height of the rectangle.
ws, hs	The distance from the sides (*ws*), and from the top or bottom (*hs*) of the rectangle to the symbol cell. If *ws* is not specified, the system default of 7 display units is used. *hs* has a default of 3 display units.

The combined numeric value of *w* and two times *ws* (one *ws* per side) cannot exceed 127. Similarly, the combined numeric value of *hh, hr,* and two times *hs* cannot exceed 127. If the value exceeds 127, or if 0 is entered for any value, the system will display `Value out of range` and the command is ignored.

The symbol is created with the top left corner of the symbol cell positioned at the graphics cursor.

If another symbol, an alphanumeric field, a line, or the drawing edge is in the area to be allocated to the new symbol cell, a symbol boundary conflict will exist, preventing the symbol from being placed. The system will display `Symbol boundary conflict`. The conflict must be resolved before the new symbol can be entered into the drawing.

A unique reference number is assigned to the block symbol in the upper left-hand corner.

.A uses symbol definition instructions to generate a symbol. Symbol definition instructions are described in the chapters on symbol definition in the *FutureNet User Manual*. The list of instructions used to define the block symbol can be viewed by entering Symbol Definition Mode (.s command), and even edited. However, once a symbol has been viewed or edited in Symbol Definition Mode, a bit is set that prevents it from being edited using symbol definition commands, namely the Add/Replace Pin Stub commands. In order to use these commands on symbols that have been edited in Symbol Definition Mode, enter the .SBS command prior to exiting Symbol Definition Mode.

Related Commands

.-	.B
.>	.F
.-AI	.SBS
.-AO	

/AD /AL /AR /AU — Draw/Erase Arrowhead

Format **/AD | /AL | /AR | /AU**

Purpose Insert or delete the arrow type specified.

Remarks

/AU	Inserts/deletes an arrow pointing up.
/AD	Inserts/deletes an arrow pointing down.
/AL	Inserts/deletes an arrow pointing left.
/AR	Inserts/deletes an arrow pointing right.

Arrows may be inserted anywhere on any type of line. The point of the arrow will be located at the graphics cursor. Arrows can be deleted by placing graphics cursor at the point of the arrow and entering the command option that corresponds to the direction of the arrow.

An arrow pointing into a line cannot be at the end of the line. Also, an arrow cannot be drawn on a line segment shorter than three display units.

If the graphics cursor is not located on a line, the system will display the error message

```
Cursor not on a vertex or line
```

Related Commands /D

A1...D8 — Move to Coordinates

Format

A1 | A2 | . . . D8

Purpose

Move the graphics cursor and drawing viewpoint to a particular region of the screen.

Remarks

The coordinates available varies with the drawing size. The grid of the coordinates is about 1/8 of the drawing width by 1/4 of the drawing height. The A1 command will place the drawing viewpoint in the lower right corner, approximately 1/8 the drawing width from the drawing's right edge and 1/4 of the drawing height from the bottom.

The D8 command will place the drawing viewpoint in the upper left corner of the drawing.

In full or intermediate zoom, this command will, if possible, update the display so that the cursor is centered on the screen.

In fit zoom, this command moves the cursor to the intersection of the requested coordinates.

Related Commands

PAN
WINDOW
CURSOR

Add/Replace Target Line

(Symbol Definition Mode)

Format

instruction ⏎

Purpose

Add a new instruction to the symbol definition instruction list.

Remarks

Not supported by FutureNet OEM products.

This command adds a new instruction to the symbol definition list or replaces an existing instruction.

To add an instruction, enter the instruction on the command line and enter. The instruction list will move up, leaving a blank target line that is ready for the next instruction.

To replace an existing instruction, position the instruction to be replaced on the target line and enter a new instruction on the command line and enter. Note that the instruction list will not move up, but will remain at the newly added instruction. You can replace this instruction, move to another instruction and replace it, or move the target line to the end of the list and continue adding instructions.

The entire symbol definition instruction must be retyped if any changes are to be made to it; no provisions are made for editing an existing symbol definition line.

The instruction is checked for syntax errors before the target line is replaced.

Related Commands

.D
.I
.CLR

Arrow Keys — Move Multiple Display Units

Format

LEFT *number*
RIGHT *number*
UP *number*
DOWN *number*

Keys

[*number*] ←
[*number*] →
[*number*] ↑
[*number*] ↓

Purpose

Move the graphics cursor the specified number of display units or grid units in the direction indicated.

Remarks

Default cursor movement is one display unit, or one grid unit when **Snap to Grid** is enabled.

Typing a number on the command line and then pressing any of the arrow keys moves the cursor the specified number of display units or grid units in the direction of the arrow. While the number remains on the command line, subsequent use of any of the arrow keys moves the cursor the number of display units or grid units specified.

When the graphics cursor is being operated from the command line, it can move in any direction until it encounters the edge of the drawing screen. At that point, continuing to enter the command in the same direction has no effect. Once the graphics cursor encounters the edge of the screen, it can be centered again using the HOME command, then moved further in the same direction until the edge of the drawing or screen is again encountered.

When automatic panning is on, and the cursor reaches the edge of the screen, the cursor is automatically centered to allow further cursor movement. The cursor can move in any direction until the edge of the drawing is reached.

If the number value given would take the cursor past the edge of the drawing, the cursor moves to the edge of the drawing.

For command line or command file operation, use the commands LEFT, RIGHT, UP and DOWN.

Related Commands

CURSOR
GRID

Arrow Up/Down Keys — Move Symbol Definition List Up/Down

(Symbol Definition Mode)

Format

UP ***number***
DOWN ***number***

Keys

[*number*] ↑
[*number*] ↓

Purpose

Move the symbol definition instruction list up or down past the target line.

Remarks

Not supported by FutureNet OEM products.

The symbol definition list moves up or down one line at a time, if no number is specified. If a number is specified, the list moves up or down by number of lines specified.

*Note: If you specify 8 – **Arrow Up** and there are only 5 lines above the cursor, the cursor will move up only 5 lines.*

The symbol definition list can also be traversed using the mouse. Clicking the left button on the desired command will move that command to the target line. Clicking the left button above or below the list will cause a Page Up or Page Down.

For command line or command file operation, use the commands UP and DOWN.

Related Commands

PC	SUN
PgUp	R9
PgDn	R15

AUTO — Automatic Command Execution

Format

AUTO *filename*
AUTOEX *filename*

Purpose

Run the specified command file automatically.

Remarks

A filename is required. AUTO will search for command files with the default .cmd command file extension unless a different file extension is specified.

This command can also be used in the Symbol Definition Mode.

The command file named must be a standard ASCII text file and must follow the command file format. That is, commands on the same line must be separated by semicolons (;), a line cannot exceed 256 characters, and commands must include all required parameters. Commands are entered in sequence.

During automatic command file execution, commands can be typed on the command line. After the command on the command line has been entered, automatic command execution will resume, unless the command is one that affects command execution, such as PAUSE or STOP.

A second command file can be initiated from within the first by including the AUTO *filename* command within the first command file. Naming one command file within another command file stops the execution of the first command file and starts execution of the second command file. It does not clear the current drawing work space, and there is no return to the first file.

Note, however, that AUTO *filename* must end the line it is entered on. Any commands on the same line that follow AUTO will not be entered.

When a command file is being processed with AUTO, commands that require a yes or no response are automatically given a yes so execution is not halted.

Commands which call dialog boxes are ignored.

To automatically run a command file when initiating FutureNet, see the reference for the fn command.

Related Commands

EXEC
NOTE
PAUSE
STOP
VIEW

AUTOPAN — Enable/Disable Automatic Panning

Format

AUTOPAN [ON | OFF]

Purpose

Turn automatic panning on and off.

Remarks

Autopanning is enabled by default.

When automatic panning is enabled, and the graphics cursor moved with the mouse reaches the edge of the display screen, the display shifts so that the cursor is centered on the screen. Automatic panning occurs at the edge of the drawing space when the graphics cursor is moved with the arrow keys.

When automatic panning is off, the HOME command must be used to reposition the display.

Entering AUTOPAN without parameters toggles the setting.

When AUTOPAN is entered, a status message appears indicating whether automatic panning is enabled or disabled.

Related Commands

PROFILE

AUTOSAVE — Save Drawing Automatically

Format

AUTOSAVE [[*commands*][,*minutes*] | [ON] | [OFF] | [STAT]]

Purpose

Periodically save drawing contents to temporary file.

Remarks

The drawing saved automatically is written to a temporary file. If the current drawing session should terminate abnormally, this temporary file can be copied and used in lieu of the original drawing file.

On PC systems, drawing data is saved to the current directory. On Sun systems, drawing data is saved to the user's home directory. The files are given a filename following these conventions:

dwgfilename.nnn	(If a drawing filename has been given)
autotemp.*nnn*	(If no drawing filename has been given)

where *dwgfilename* is the name of the drawing file being edited, and *nnn* is the number of successful automatic saves of the drawing, ranging between 1 and 999. Each time an automatic save is successful, the previous AUTOSAVE file is deleted and a new one created.

AUTOSAVE ON is the default, with the *commands* and *minutes* parameters set as outlined below.

Command parameters work in this fashion:

commands	Causes an automatic save after the given number of commands have been entered. The default is 50.
minutes	Causes an automatic save after the given number of minutes have passed. The default is 30 minutes.
ON	Turns on AUTOSAVE with current/default parameters active.
OFF	Turns off AUTOSAVE.
STAT	Returns the current status of AUTOSAVE.
no options	Toggles AUTOSAVE on or off.

Entering AUTOSAVE with one option changes only that option. For instance, the following command causes an automatic save every 5 minutes, and the number of commands between saves is unaffected:

AUTOSAVE ,5

Automatic saves occur when either option is reached, then both counters are reset. Using the default options as an example, if an automatic save happens at 30 minutes and only 15 commands have been entered, both the *command* and *minute* counters are reset.

Automatic saves occur only if the drawing has been modified, and only if a command has been issued since the last automatic save.

You may standardize how AUTOSAVE functions for you by including the command in your **fnpro.cmd** file. Temporary files created by AUTOSAVE are not deleted automatically at the end of a session.

'B — Enable/Disable Alphanumeric Field Boundary Display

Format

'B [ON | OFF]

Purpose

Display or hide all alphanumeric field boundaries.

Remarks

The default setting is OFF.

When ON, each alphanumeric field is displayed with a dotted boundary and its point of justification. The point of justification is the round dot that appears in the lower left, center, or right of the field.

When OFF, alphanumeric field boundaries and points of justification are not displayed.

If no parameter is specified, 'B toggles the current setting.

Displaying alphanumeric field boundaries aids in the positioning of text fields, particularly signal name or pin name fields.

The boundaries are displayed at all zoom levels.

This feature can be particularly useful for solving boundry conflict situations at zoom levels greater than full zoom (when alphanumeric fields are not visible).

.B — Create Block Symbol

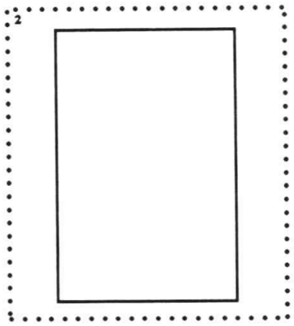
Format

.B *w,h* [,*ws* [,*hs*]]

Purpose

Create a block symbol of the specified size.

Remarks



Not supported by FutureNet OEM products.

w is the width of the symbol rectangle expressed in display units; *h* is the height of the rectangle.

ws is the distance between the sides of the rectangle and the symbol cell. If *ws* is not specified, the system default of 5 display units is used.

hs is the distance between the top or bottom of the rectangle and the symbol cell. If *hs* is not specified, the system default of 3 display units is used.

The combined numeric value of *w* and two times *ws* (one *ws* per side) cannot exceed 127. Similarly, the combined numeric value of *h* and two times *hs* cannot exceed 127. If the value exceeds 127, or if 0 is entered for any value, the system will display

```
Value out of range
```

and the command will be ignored.

The symbol is created with the top left corner of the symbol cell positioned at the graphics cursor location. A new reference number is assigned to the block symbol.

If another symbol, an alphanumeric field, a line, or the drawing edge is in the area to be allocated to the new symbol cell, a symbol boundary conflict will prevent the symbol from being placed. The message

```
Symbol boundary conflict
```

will be displayed. The conflict must be resolved before the new symbol can be entered into the drawing.

.B uses symbol definition instructions to generate a symbol. Symbol definition instructions are described in the chapters on symbol definition in the *FutureNet User Manual*. The list of instructions used to define the block symbol can be viewed and edited by entering Symbol Definition Mode (.s command). However, once a symbol has been edited in Symbol Definition Mode, a bit is set that prevents it from being edited using drawing mode commands, such as the Add/Replace Pin Stub commands. In order to use these commands on symbols that have been edited in Symbol Definition Mode, enter the .SBS command prior to exiting Symbol Definition Mode.

Related Commands

.–
A
.D
.F
.SBS

BLINK — Enable/Disable Blinking

Format

BLINK [ON | OFF]

Purpose

Turn blinking on and off.

Remarks

The BLINK command controls blinking for the following:

- Zoom window (visible when in fit zoom).
- Alphanumeric fields the graphics cursor is located in.
- Tagged alphanumeric fields, symbols, and areas.
- The direct-connection cursor (/C command).
- Temporary lines.

Blinking is the default value on monochrome systems. Non-blinking is the default value on color systems.

Entering BLINK without parameters toggles the setting.

BX, BY — Draw Broken Line

(Symbol Definition Mode)

Format

BX ***length***
BY ***length***

Purpose

Draw a textured line of the specified *length* along the *x* or *y* axis.

Remarks

Not supported by FutureNet OEM products.

The texture of the line to be drawn is determined by the SL command. The default texture type is a dotted line. The line type can be seen in the Line Status box, along with the appropriate SL command.

The line begins at the current *x,y* coordinates.

Length is in display units and can have values between +127 and -128.

BX *length* draws a textured horizontal line along the *x* axis from the current *x,y* coordinates. If *length* is negative, the line is drawn to the left; if *length* is positive, the line is drawn to the right.

BY *length* draws a textured vertical line along the *y* axis from the current *x,y* coordinates. If *length* is negative, the line goes up; if *length* is positive, the line goes down.

The symbol editor coordinate marker is relocated to the end of the line drawn as a result of this instruction.

Related Commands

SL
BXD
BYD

BXD, BYD — Draw Broken Line in Dot Units

(Symbol Definition Mode)

Format

BXD *length*
BYD *length*

Purpose

Draw a textured line of the specified *length* along the *x* or *y* axis.

Remarks

Not supported by FutureNet OEM products.

The texture of the line to be drawn is determined by the SL command. The default texture type is a dotted line. The line type can be seen in the Line Status box, along with the appropriate SL command.

The line begins at the current *x,y* coordinates.

Length is in dot units and can have values between +127 and -128.

BXD *length* draws a textured horizontal line along the *x* axis from the current *x,y* coordinates. If *length* is negative, the line is drawn to the left; if *length* is positive, the line is drawn to the right.

BYD *length* draws a textured vertical line along the *y* axis from the current *x,y* coordinates. If *length* is negative, the line goes up; if *length* is positive, the line goes down.

The symbol editor coordinate marker is relocated to the end of the line drawn as a result of this instruction.

Related Commands

SL
BX
BY

BXL, BXR — Draw x to Symbol Cell Boundary with Bubble

(Symbol Definition Mode)

Format

BXL
BXR

Purpose

Draw a line to the left or right symbol cell boundary along the *x* axis from the current *x,y* coordinates, placing a bubble at the *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

This command is particularly useful in drawing line stubs to the symbol cell boundary.

BXL places a bubble on the symbol cell at the current *x,y* coordinates and draws a line along the *x* axis to the left symbol cell boundary.

BXR places a bubble on the symbol cell at the current *x,y* coordinates and draws a line along the *x* axis to the right symbol cell boundary.

The symbol editor coordinate marker location is not changed by these instructions.

Related Commands

.w, h
DXL
DXR
BYU
BYL

BYU, BYL — Draw y to Symbol Cell Boundary with Bubble

(Symbol Definition Mode)

Format

BYU
BYL

Purpose

Draw a line to the upper or lower symbol cell boundary along the *y* axis from the current *x,y* coordinates, placing a bubble at the *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

This command is primarily used to draw line stubs within the symbol cell.

BYU places a bubble at the current *x,y* coordinates and draws a line along the *y* axis to the upper symbol cell boundary.

BYL places a bubble at the current *x,y* coordinates and draws a line along the *y* axis to the lower symbol cell boundary.

The symbol editor coordinate marker location is not changed by these instructions.

Related Commands

.w, h
DYU
DYL
BXL
BXR

'C — Copy Alphanumeric Field

Format

'C

Purpose

Tag the current alphanumeric field and move a copy of it to a different location.

Remarks

Place the cursor within the target field's boundary, and enter 'C. When entered, the MODE status field changes from the current mode to COPY and the graphics cursor becomes the tag cursor.

When the tag cursor is moved, a dotted boundary detaches from the field and can be moved around the drawing using the mouse or cursor movement commands. The tagged field retains a highlighted boundary. The ghosted boundary aids in locating the text field so that it can be positioned without causing a boundary conflict. You can also use 'B to display boundaries of all alphanumeric fields. On monochrome systems, BLINK causes the tagged field to blink so that it is easier to see. On color systems, the tagged field changes color.

When 'C is entered again, the field is copied into the new cursor location, assuming there are no boundary conflicts with other symbol cells, alphanumeric fields, lines, or the edge of the drawing. To resolve boundary conflicts, move the copy to a valid location, or move or erase conflicting objects.

The location of the original alphanumeric field remains unchanged. The tag adjusts only the location of the copied field.

After the field is copied and tag and drag are canceled, control returns to the previous mode, and the graphics cursor is restored.

Related Commands

'B
'K
BLINK
'M

'CH A — Change Attribute for Existing Field

Format

'CH A ***name*** **|** ***number***
'CH ATTRIBUTE ***name*** **|** ***number***

Purpose

Change the attribute of an existing alphanumeric field.

Remarks

This command changes the attribute for the field where the graphics cursor is located. The attribute is shown in the ATTR status field.

Attribute names and numbers are listed and explained in the chapter on "Understanding FutureNet" in the *FutureNet User Manual.*

If the name or number is not specified, a menu of available attributes will be displayed and can be selected with the mouse.

Note that the current attribute assignments of all fields can be seen by entering 'D command.

Related Commands

'A
'D
'S

'CH F — Change Font

Format

'CH F [0 | 1 | 2 | 3 | 4 | 5 | 6 | 7]

Purpose

Change the font size of an existing alphanumeric field.

Remarks

This command changes the font size for the current alphanumeric field. The font size is shown in the ALPH status field.

Any of the seven font sizes can be selected. If you do not specify a font size, 'CH F will cycle through the font sizes in ascending order, beginning with the current font size, showing them in the ALPH field.

Changing a field's font to a larger size means the size of the field will increase to accommodate the larger font. If this is impossible because of the field's proximity to other fields or graphic items, a boundary conflict will prevent the change from occurring until the boundary conflict is eliminated.

Changing a field's font to a smaller size means the size of the field will decrease to accommodate the smaller font. As the field decreases in size, the graphics cursor may be left outside the field boundary. This may be important when a command file is being used because it can result in a previously selected alphanumeric field no longer being selected and commands left unentered.

Related Commands

'0...'7

'CH J — Change Justification

Format

'CH J [L | R | C]
'CH JUSTIFICATION [L | R | C]

Purpose

Change the justification (left, right, or center) of an existing alphanumeric field.

Remarks

This command changes the justification for the current alphanumeric field. The minimum entry is 'CH J, which cycles through the options from left to right to center. The justification is shown on the center line of the ALPH status field.

Select left, right, or center justification. Left, L, enters text to the right of the initial cursor position. Right, R, enters text to the left of the cursor position. Center, C, enters text evenly on either side of the initial cursor position.

Note that the 'B command displays the field boundary and the point of justification, which is the dot appearing at the lower left, center, or right of the field.

Related Commands

'B
'J

'CH O — Change Orientation

Format

'CH O [H | V]
'CH ORIENTATION [H | V]

Purpose

Change the orientation of an existing alphanumeric field.

Remarks

This command changes the orientation of the current alphanumeric field. The orientation, which is shown on the fourth line of the ALPH status field, can be either horizontal (H) or vertical (V).

If no orientation is specified, the command toggles the setting. The default setting is horizontal.

If changing a field's orientation is impossible because of the field's proximity to other fields or graphic items, a boundary conflict message is displayed, and the orientation is not changed. The center of the field is the pivot point for changing orientation. Ensure that there is adequate space for the text field in the new orientation or a boundary conflict will prevent the field from being placed.

In the vertical orientation, text is displayed and read from bottom to top. In the horizontal orientation, text is displayed and read from left to right.

Note: When using 'CH O in command files, make sure the graphics cursor is positioned in the center of the text field it is to change. Otherwise, the command will be ignored.

'CH P — Change Printability

Format

'CH P [ON | OFF]
'CH PRINTABILITY [ON | OFF]

Purpose

Identify an existing alphanumeric field as being printable or non-printable.

Remarks

This command operates on the current alphanumeric field. Printability is shown on the top line of the ALPH status field. OLD = printable. ONP = old non-printable.

ON sets a field to printable; OFF sets a field to non-printable.

If no parameter is specified, entering 'CH P toggles the current setting.

Printability may be determined by the default attribute.

Note: All non-printable fields can be forced to print, without changing each individual field, using the **R/RNO** *option of the printopt command. This is useful for checking a drawing for completeness of information not normally printed on a finished drawing.*

Related Commands

PRINTOPT
PROFILE

'CH R — Change Reverse Video State of Alphanumeric Field

Format

'CH R [ON | OFF]

Purpose

Toggle the reverse video display of an existing alphanumeric field.

Remarks

This command toggles the reverse video display on the current alphanumeric field.

ON causes the field to appear in reverse video in the drawing.

OFF causes the field to appear in normal video.

If no parameter is specified, reverse video toggles on and off.

Reverse video is useful for emphasis in custom menus or help screens. Refer to the *FutureNet User Manual* for more information on creating custom menus and help screens.

Related Commands

HELPSAVE
?
'CH V

'CH V — Change Visibility of Alphanumeric Field

Format

'CH V [ON | OFF]

Purpose

Toggle the visibility of an existing alphanumeric field

Remarks

This command toggles the visibility for the current alphanumeric field.

ON causes the field to be visible during the editing session.

OFF causes the field to invisible during the editing session.

If no parameter is specified the visibility toggles on and off.

Invisible alphanumeric fields follow the same boundary conflict rules as visible text.

To edit invisible text, place the cursor on the text and the text will appear. When the cursor is not on the field, it is invisible.

To locate invisible text, use the 'B (Display Alphanumeric Boundary Display) command. The boundaries of all alphanumeric fields, visible or invisible, are displayed.

Invisible text is used in custom menus and help screens. Refer to the *FutureNet User Manual* for more information on creating custom menus and help screens.

Related Commands

HELPSAVE
?
'CH V

.C — Copy Symbol

Format

.C

Purpose

Copy the tagged symbol to the current cursor location.

Remarks

When the symbol to be copied is not tagged, first move the cursor into the symbol and enter .C or press the left mouse button to tag the symbol and press middle mouse button to cycle through MOVE/COPY/ERASE mode.

When the symbol to be moved is tagged and the cursor is moved, the boundary detaches from the symbol and moves around the drawing with the mouse or cursor movement commands. The tagged symbol boundary and cursor blink on monochrome systems or change color on color systems so that the symbol and its new location are obvious. When .C is entered again, the symbol is copied into the boundary.

The Copy command enters only if the symbol is being copied to a location clear of symbol cells, alphanumeric fields, and lines, and does not overlap the edge of the drawing. If there are any interfering drawing elements, the command is ignored and the system displays

```
Symbol boundary conflict
```

A new reference number is assigned to the copied symbol.

The original symbol remains unchanged. When a symbol is copied, MOVE/COPY/ERASE mode is canceled.

Related Commands

.K
.M
BLINK

.CLR — Erase Symbol Definition

(Symbol Definition Mode)

Format

.CLR

Purpose

Clear the symbol definition list and the symbol definition workspace, so that a new symbol can be created.

Remarks

Not supported by FutureNet OEM products.

The symbol cell size remains as previously defined and the symbol cell boundary is still displayed.

All symbol definition instructions are deleted from the symbol definition workspace.

Following a .CLR command, the symbol definition coordinate marker is in the upper left corner of the symbol cell.

Related Commands

.D
.S

/C — Draw Direct Connection Through Symbol

Format

/C

Purpose

Draw intra-symbol connections.

Remarks

/C is used to electrically connect two pins on the same symbol by drawing a line through the symbol. /C cannot be used to connect pins on different symbols.

If a pin is connected to a signal and the pin is also directly connected to other pins in the symbol, all of those pins are considered to be connected in the same net.

Position the graphics cursor on the desired pin stub and then enter the /C command. Draw a line to connect the two pin stubs and re-enter the command or click the left mouse button.

Direct connections are only displayed when the .DCON command is on. The direct connection appears as a dotted line inside a symbol.

If the graphics cursor is not located on a symbol boundary when the first /C command is entered, the command is ignored and the message line displays

```
Invalid line start location
```

If the direct connection does not end on a symbol boundary, the line is not drawn and the system displays

```
Direct connect segment must end on a pin stub or symbol
boundary
```

If a second /C is entered when the cursor is in a different symbol than the first /C, the system displays

```
Direct connect segment must end in same symbol
```

If two symbols are 2 display units apart or closer, then when the first /C command is entered, the line drawing may snap to an adjacent symbol. You may need to turn PINSNAP off temporarily or move the symbols farther apart.

Related Commands

.DCON
/ES
/EL
/EN

[C — Copy Area

Format

[C

Purpose

Copy the tagged area to the current cursor location.

Remarks

The area to be copied must first be defined using [D, and then tagged using the [C command or the left mouse button.

When [C is entered, the area to be copied is tagged and a boundary detaches from the area. The boundary can be moved to the desired location using the mouse or cursor movement commands. When [C is entered again, the area is copied to the new cursor location, assuming there are no boundary conflicts.

Symbols and alphanumeric fields are treated as being within the area to be copied only if they are completely within the area boundary. Symbols and alphanumeric fields intersecting the area boundary are excluded from the copy. Line segments are broken at the area boundary and the segments inside are copied. The densely dotted area boundary is moved with the area.

New symbol reference numbers are assigned to copied symbols.

The Copy command works only if the new location is clear of symbol cells, alphanumeric fields and lines, and does not overlap the edge of the drawing.

If the area cannot be copied because of a boundary conflict, the command is ignored and the system displays

```
Boundary conflict
```

To resolve the boundary conflict, move or erase the conflicting drawing elements or reposition the area.

The original area remains unchanged. The dotted boundary stays with the copied area. After the area has been copied, tag and drag is canceled and control returns to the previous mode.

Related Commands

[D
[K
[M

CD — Change Current Directory

Format

CD [*drive*][*path*]
CHDIR [*drive*][*path*]

Purpose

Change the current path to the one specified.

Remarks

The CD command operates a little differently in FutureNet than in DOS. If a drive is specified in the FutureNet CD command, then the current drive, as well as the current directory, is modified. Under DOS, only the current directory of the specified drive is modified. The current drive is not modified.

The CD command operates similar to the UNIX CD command on a Sun platform.

Related Commands

PC	Sun
DIR	LS
DEL	RM

CLEAR/ERASE — Clear Drawing Hierarchy or Erase Current Drawing

Format

CLEAR
ERASE

Purpose

CLEAR clears the current drawing in the work space and all drawings in the hierarchy that have been accessed and modified.

ERASE erases only the current drawing from the work space, and does not erase other drawings in the hierarchy that may have been accessed and modified.

Remarks

Use the CLEAR command to clear the current drawing in the work space and any drawings in a structured design hierarchy that have been accessed using the #D, #U, #R, and #L commands and then modified. The portions of the drawing structure that have not been accessed or modified will not be affected.

To help avoid clearing drawings that may have been accessed and modified elsewhere in the hierarchy, the CLEAR command will ask for verification before clearing the memory. If no files have been modified, the command clears the memory without asking for verification.

If a current version of the drawing has not been saved when the ERASE command is entered, the ERASE command will ask for verification before erasing the drawing from memory. If the drawing file in memory has not been modified, the command erases the drawing without asking for verification.

Neither CLEAR nor ERASE removes the last-saved version of the drawing from the hard disk.

The CLEAR command clears the UNDO/REDO stacks. The ERASE command can be reversed using the UNDO command, if UNDO is enabled.

COLOR — Modify Color Palette

Format

COLOR [*color1,color2,...color8*]

Purpose

Select the background color and modify the current color palette used by the DISPGRP*n* command.

Remarks

This command performs two functions: selects the background color, which is always assigned to position five (color5) on the palette; and selects which colors are assigned to the remaining seven positions on color palette.

The colors of the color palette are used by the DISPGRP*n* command, which assigns colors to the elements of the FutureNet display.

The COLOR command changes the color settings of the current palette. The color palette has eight color positions, numbered color1 through color8. One color can be assigned to each position.

The COLOR command has eight positional parameters, separated by commas (,). Each position represents one of the colors on the color palette. To change a color on the palette, assign a different color to a given position. For example

COLOR ,,,,,G

only affects the color assignment for position 6 (color6), which would have its present color assignment changed to G (green). All drawing components that have been assigned the position 6 color (as specified by the DISPGRP*n* command) will be changed to green. Drawing components that have been assigned position 1, 2, 3, 4, 5, 7, and 8 colors on the palette will retain their current color assignments.

There are three ways to assign colors: enter the initial(s), the number, or the name.

For example

COLOR GREEN,WHITE,R,B,BLACK

Note: Components that are the same color as the background will be indistinguishable, even though they are displayed.

Default color assignments are

Assignment	Name
Color 1	Yellow
Color 2	Light-White
Color 3	Light-Red
Color 4	Light-Cyan
Color 5	Blue
Color 6	Light-Green
Color 7	Brown
Color 8	Magenta

There are 16 colors available. Specify colors by their initial(s), number, or name.

Initial	Number	Name
BL	**0**	Black
B	**1**	Blue
G	**2**	Green
C	**3**	Cyan (blue-green - BG)
R	**4**	Red
M	**5**	Magenta
Y	**6**	Yellow
W	**7**	White
GY	**8**	Grey
LB	**9**	Light-Blue
LG	**10**	Light-Green
LC	**11**	Light-Cyan (light-blue-green - LBG)
LR	**12**	Light-Red
LM	**13**	Light-Magenta
BR	**14**	Brown
LW	**15**	Light-White

Related Commands

DISPGRPn
PALETTE

CONNECT — Enable/Disable Maintenance of Line Connections (Rubberbanding)

Format

CONNECT [ON | OFF]

Purpose

Disable or enable the maintenance of line connections (rubberbanding) when moving a symbol or area.

Remarks

When a symbol or an area is moved with CONNECT enabled, its connections are maintained unless invalid line routing is created. If this occurs, temporary lines are created. Temporary lines are dotted (they also change color on color systems), and overlay everything in their path.

Only /1 and /2 line types can be rubberbanded.

When CONNECT is disabled and a symbol or area is moved, lines are broken at the symbol or area boundary.

If CONNECT is enabled and a symbol is tagged and moved but not set, entering CONNECT will disable rubberbanding as well as cancel any rubberbanding in progress for the current symbol.

When CONNECT is entered, a message appears indicating whether CONNECT is enabled or disabled.

Enabled is the default setting.

Related Commands

PROFILE

CONTEXT — Restore Drawing Context

Format

CONTEXT

Purpose

Restore the context of a drawing.

Remarks

Context refers to the state of a drawing at the time it was last saved. Context includes cursor position, drawing screen window, area definition, discrete zoom level (not dynamic zoom level), font type, attribute type, and line type. All are saved with the drawing. When the CONTEXT command is entered after a Load Drawing command, the drawing is restored to the state it was in when last saved. No command is necessary to store the information that CONTEXT restores.

Ctrl – Backspace Keys — Erase Alphanumeric Data

Keys

[Ctrl] – [Backspace]

Purpose

While in alphanumeric mode, this command erases the data in the current alphanumeric field.

Remarks

After erasing the data, the cursor is placed at the beginning of the field.

If alphanumeric mode is exited immediately after entering [Ctrl] – [Backspace], the field is deleted from the drawing.

Ctrl – Home or Ctrl – R7 Keys — Move Cursor to Beginning of Alphanumeric Field
Ctrl – End or Ctrl – R13 Keys — Move Cursor to End of Alphanumeric Field

Keys

PC	Sun
Ctrl – Home	Ctrl – R7
Ctrl – End	Ctrl – R13

Purpose

Move the cursor to the beginning or end of an alphanumeric field (when in alphanumeric mode) or the command line.

Remarks

Ctrl – Home or Ctrl – R7 moves the cursor to the beginning of the current alphanumeric field, or to the beginning of the command line.

Ctrl – End or Ctrl – R13 moves the cursor to the end of the current alphanumeric field, or to the end of the command line.

Moving the cursor with these commands does not affect the alphanumeric field.

CURSOR — Set Cursor to Absolute Location

Format

CURSOR *x,y* [*,xw,yw*]

Purpose

Specify up to two sets of coordinates: one set to specify the graphics cursor location (*x,y*) and a second optional set to specify the location of the upper left corner of the display area (*xw,yw*).

Remarks

The *x,y* are required; *xw,yw* are optional.

If both *x,y* and *xw,yw* are specified, the upper left corner of the window is placed at the intersection of the *xw,yw* coordinates and the cursor is placed at the intersection of the *x,y* coordinates. If the *x,y* coordinates fall outside the window area specified by the *xw,yw* coordinates, the *xw,yw* coordinates are ignored and the window is centered around the *x,y* coordinates specified for the cursor. If the *xw,yw* coordinates are too close to the border to accommodate the entire window, then the location of the window is adjusted so that the upper left corner of the window is placed as near to the *xw,yw* coordinates as possible.

This command is intended primarily for use in automatic command execution sequences.

#D — Move Down in Design Hierarchy

Format

#D[*filename*]

Keys

PC	SUN
Ctrl – PgDn	Ctrl – R15

Purpose

Move down in the design hierarchy to the specified file or create a new file with the specified name.

Remarks

#D *filename* loads the specified drawing file from a lower level in the hierarchy or creates a new file with the specified name at a lower level in the design hierarchy. It saves the current drawing, its status, and the graphics cursor location in a temporary file. Drawings that are saved into temporary files when **#D** is entered are given unique filenames based on the root filename with a number for the extension. If a file of the same name exists, then the number will be incremented until the filename is unique. The drawings must be saved using the SAVEALL and SAVE commands before exiting FutureNet.

This command requires that a root drawing be loaded with the LOAD command before **#D** is entered. If the root drawing is just being created, then it must first be saved before **#D** is entered. If **#D** is entered before the root drawing is saved, the system displays

```
Drawing must be saved before moving down in the drawing
structure
```

The optional parameter *filename* is available only with the **#D** form of the command. Ctrl – PgDn or Ctrl – R15 does not accept a filename.

#D and Ctrl – PgDn or Ctrl – R15 can be used to create a lower level drawing without specifying a filename, if the cursor is located on an alphanumeric field that has been assigned the FILE or FILN attribute. These attributes (file name pointers) are assigned to the displayed name of a functional block in hierarchical drawings. A newly created file will use the field's display text as the name of the file.

If the graphics cursor is not on an alphanumeric field with attribute FILE or FILN and no filename is entered, the command has no effect.

If the file named or located by the graphics cursor does not yet exist, the system displays

```
File not found.  OK to create new file (Y/N)?
```

Y creates a new blank file with that name; N cancels the command.

If the file named or located by the graphics cursor has already been viewed at a higher level in the hierarchy, and therefore is presently saved in a temporary file, the system cancels the command and displays

```
File already accessed in drawing structure
```

If the drawing named or located by the graphics cursor exists, has been viewed in this session and left again, and is at a lower level in the design hierarchy, then it is loaded with its status and the graphics cursor as they were when it was left.

#D ***filename*** can be used to jump lower in the design hierarchy without accessing intermediate drawing levels by specifying the file to be opened.

Related Commands

#U
#L
#R
SAVEALL
SAVE

'D — Enable/Disable Attribute Display

Format

'D [ON | OFF]

Purpose

Display alphanumeric field attribute assignments.

Remarks

When ON, a reverse video field displaying the attribute number appears in place of the alphanumeric field(s) on the drawing.

When OFF, alphanumeric fields appear in their normal manner.

Specifying 'D without any parameters toggles the setting.

Related Commands

'A
'CH A
'S

.D — Delete Pin Stub

Format

.D

Purpose

Delete an existing pin stub.

Remarks

The graphics cursor must be positioned between the symbol cell and the outline to delete a stub. This is the same location used when the stub is created or replaced. If the cursor is not in an appropriate location, the command is ignored.

Only block symbols created with the .A, .B, or .F commands, and edited with the standard pin stub commands have the internal symbol definition instruction sequences that can be processed by this command.

Related Commands

.-
.-AI
.A
.B
.F

.D — Delete Target Line

(Symbol Definition Mode)

Format .D

Purpose Delete the symbol definition instruction currently in the target line.

Remarks *Not supported by FutureNet OEM products.*

When a symbol definition instruction is deleted, the next symbol definition instruction in sequence moves into the target line.

Related Commands .I
.CLR

.DCON — Enable/Disable Display of Direct Connections

Format

.DCON [ON | OFF]

Purpose

Displays direct connections across a symbol, instead of all other graphic data in the symbol.

Remarks

Use of the .DCON command is the only way to view direct connect lines.

This command updates the setting in the current user profile.

If no parameter is specified, the setting in the current user profile is toggled.

When turned on, .DCON turns off the display of all other items in a symbol and instead displays any direct connections drawn across the symbol.

Related Commands

/C
PROFILE

.DEL — Delete Symbol from Update Library

Format

.DEL *symbolname*

Purpose

Delete the specified symbol from the update symbol library.

Remarks

.DEL only deletes symbols from the update library. The update library is specified using the .LIB command.

Related Commands

.LIB
.DIR
.SAVE

.DIR — Display Symbol Library Directory

Format

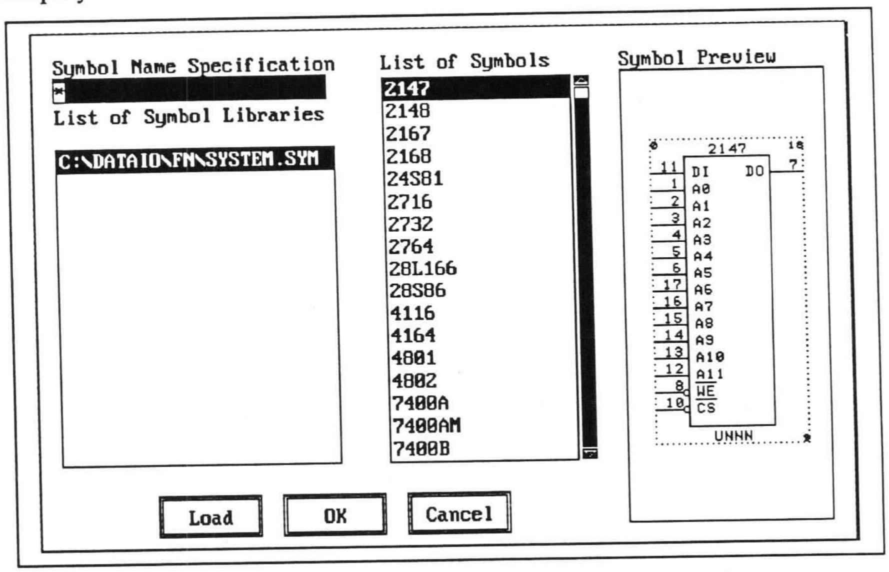
.DIR

Purpose

Display a list of library contents.

Remarks

When the .DIR command is entered, the symbol directory dialog box is displayed.



The **Symbol Name Specification** field filters the list of symbol names. To change the filter, edit the field and press ↵ . Wildcards are valid: * for any number of characters; ? for a single character.

The **List of Symbol Libraries** data box lists all open libraries, one of which is selected (highlighted). The **List of Symbols** data box displays the symbols in the selected library that match the symbol name specification. Select a different library by clicking on it with the mouse.

Use the scroll bars in the List of Symbols data box to browse the various symbol names available. To load a symbol, highlight it on the list and click on the LOAD action button. If there are no boundary conflicts, the symbol is loaded at the drawing cursor, tagged for moving. If a symbol or line is already at that location, the symbol is not loaded, and you receive the message: `Symbol boundary conflict`

Exit the .DIR dialog box without selecting a symbol by clicking on the OK action button, by clicking on the CANCEL action button, by pressing the left mouse button outside of the dialog box, or by pressing Esc .

Related Commands

.L
.DEL
LIB
.LIB
.SAVE

.DIRPR — Print Symbol Library Directory

Format

.DIRPR

Purpose

Print a list of the symbols in the libraries specified by the last .LIB command, the LIB command, and the system.sym library.

.DIRPR uses the relevant print options as set up in the PRINTOPT command.

Related Commands

.LIB
LIB
.DIR

/D — Insert/Delete Interconnect Dot

Format

/D

Purpose

Insert or delete an interconnect dot at the graphics cursor location.

Remarks

An interconnect dot is required when two lines which cross are connected. Interconnect dots are allowed, but not required at T intersections.

If there is an interconnect dot at the intersection located by the graphics cursor, /D deletes it. The location of the graphics cursor is not changed.

Interconnect dots cannot be placed within symbol cells. They can only be placed on intersections of line types /1 and /2. No other line types will accept interconnect dots.

Interconnect dots can be drawn and displayed in any zoom level less than or equal to full zoom (1.00).

The graphics cursor must be located on a line segment when /D is entered; if not, the command is ignored.

The /D command is equivalent to the middle mouse button when in FAST drawing mode.

[D — Define Area

Format

[D

Purpose

Define an area.

Remarks

An area needs to be defined before it can be moved, copied, erased, or saved.

When [D is entered, area definition is initiated; the area definition point appears at the graphics cursor location and the MODE status field displays AREA. The area definition point marks the fixed corner of the area to be defined. Define the area by moving left or right and up or down from that fixed point. Re-enter [D or press the left mouse button to save the newly defined area. The finished area boundary is a dotted rectangle.

Area definition can be initiated anytime. However, only one area at a time can be defined.

When an area is already defined and the graphics cursor is outside of that area, entering [D cancels the currently defined area and begins a new area definition at the cursor location. If this area is canceled before [D is re-entered, that is, if it is canceled while still in area definition mode, area definition is canceled and the previous area is restored.

When an area is already defined and the cursor is inside that area, entering [D restarts area definition. The corner of the area closest to the cursor "snaps" to the cursor location. This corner becomes the movable corner of the area rectangle and the corner opposite becomes the fixed corner. Area definition can then proceed as described above.

Area definition can be initiated from the menu or from the command line using the mouse or keyboard.

The cursor movement commands, and Pan and Zoom can be entered from the keyboard without canceling area definition. Entering any other command from the keyboard automatically cancels area definition and returns to the previous mode.

Related Commands

[C
[ERASE
[K
[LOAD
[M
[R
[RE
[SAVE

DA — Draw Arc in Display Units

(Symbol Definition Mode)

Format

DA *radius,center,eighths*

Purpose

Draw the specified arc in display units.

Remarks

Not supported by FutureNet OEM products.

The current *x,y* coordinates mark the beginning point of the arc.

radius specifies the radius of the circle that the arc is taken from and is expressed in display units. There are four dot units per display unit.

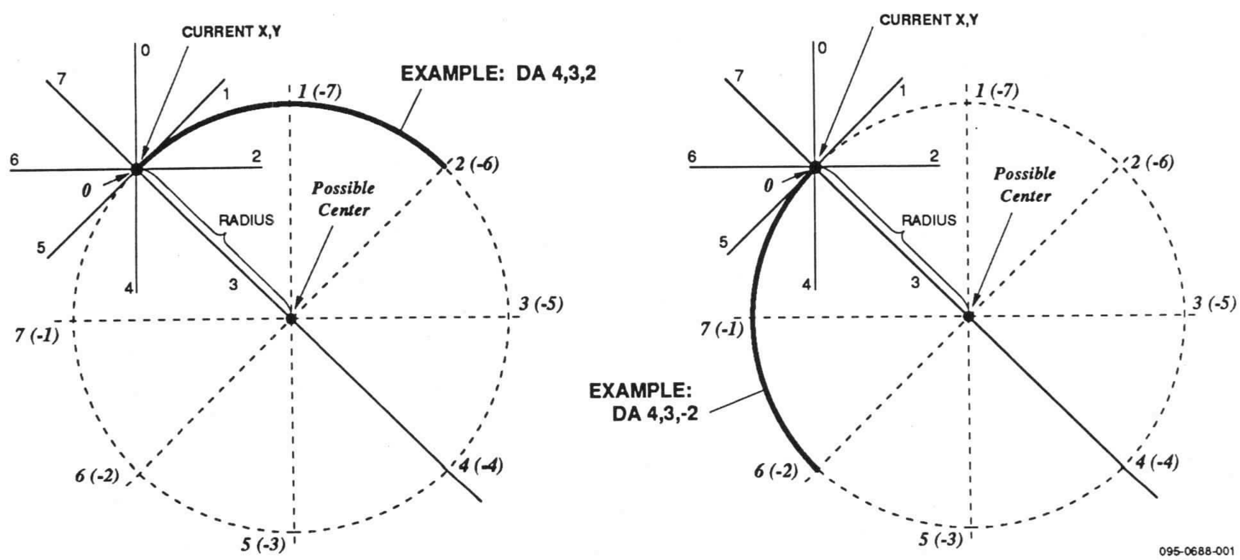
center specifies the line on which the center of the arc will fall, relative to the current *x,y* position (see the figure accompanying this entry). The center will be *radius* display units away from the start point of the arc. *center* can take values from 0 to 7.

eighths specifies the length of the arc in eighths of a circle and the direction of travel: clockwise or counterclockwise. Positive numbers from 1 to 8 draw the arc in a clockwise direction; negative numbers from -1 to -8 draw the arc in a counterclockwise direction. 1 draws 1/8 of a circle, or a 45 degree arc; 8 draws an entire circle. A value of 0 is not valid.

The symbol editor coordinate marker moves to the end of the arc drawn.

Related Commands

DAD
DC



DAD — Draw Arc in Dot Units

(Symbol Definition Mode)

Format

DAD *radius,center,eighths*

Purpose

Draw the specified arc in dot units.

Remarks

Not supported by FutureNet OEM products.

The current *x,y* coordinates mark the beginning point of the arc.

radius specifies the radius of the circle that the arc is taken from and is expressed in dot units. There are four dot units per display unit.

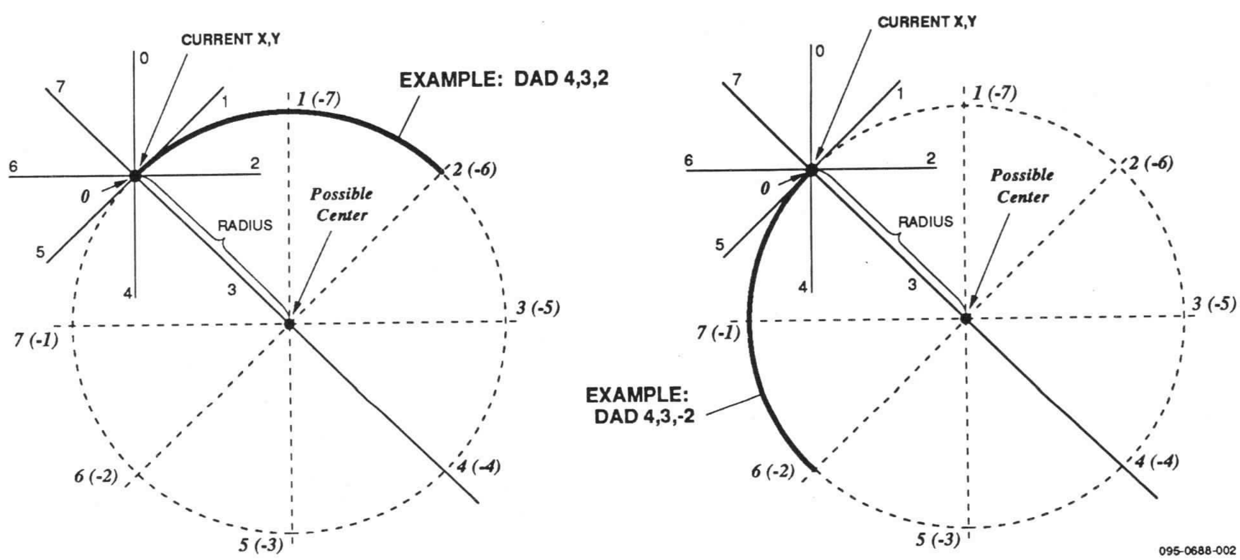
center specifies the line on which the center of the arc will fall (see the figure accompanying this entry). The center will be *radius* dot units away from the start point of the arc. *center* can take values from 0 to 7.

eighths specifies the length of the arc in eighths of a circle and the direction of travel: clockwise or counterclockwise. Positive numbers from 1 to 8 draw the arc in a clockwise direction; negative numbers from -1 to -8 draw the arc in a counterclockwise direction. 1 draws 1/8 of a circle, or a 45 degree arc; 8 draws an entire circle. A value of 0 is not valid.

The symbol editor coordinate marker moves to the end of the arc drawn.

Related Commands

DA
DCD



DASH

See fn.

DC — Draw Circle in Display Units

(Symbol Definition Mode)

Format DC ***radius***

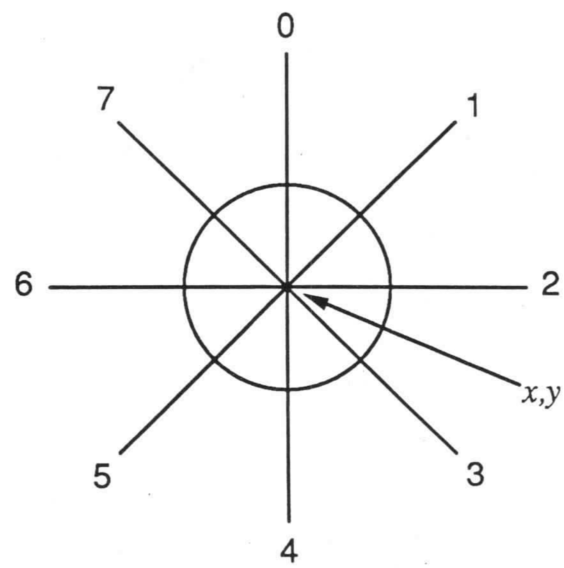
Purpose Draw a circle with the radius specified in display units.

Remarks *Not supported by FutureNet OEM products.*

radius is a required parameter and gives the radius of the circle in display units. *radius* may have values from 0 to 255.

The center of the circle is at the current *x,y* coordinate location, which is not changed by this instruction.

Related Commands DCD
DA



DCD — Draw Circle in Dot Units

(Symbol Definition Mode)

Format

DCD *radius*

Purpose

Draw a circle with the radius specified in dot units.

Remarks

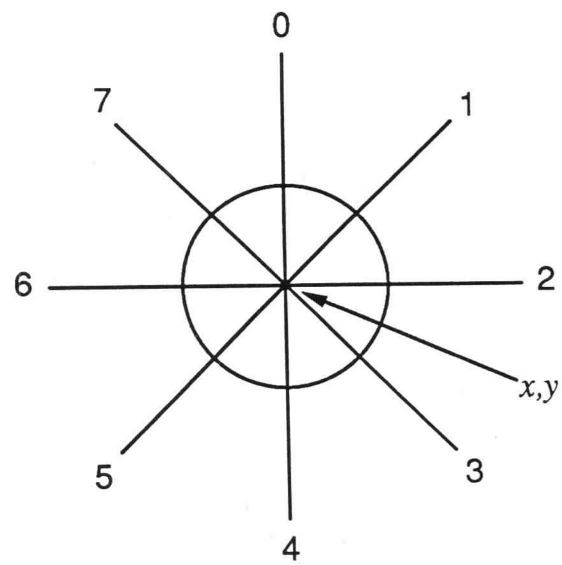
Not supported by FutureNet OEM products.

radius is a required parameter and gives the radius of the circle in dot units. *radius* may have values from 0 to 255.

The center of the circle is at the current *x,y* coordinate location, which is not changed by this instruction.

Related Commands

DC
DAD



DD — Draw Dots

(Symbol Definition Mode)

Format

DD *a,b*

Purpose

Draw two dots.

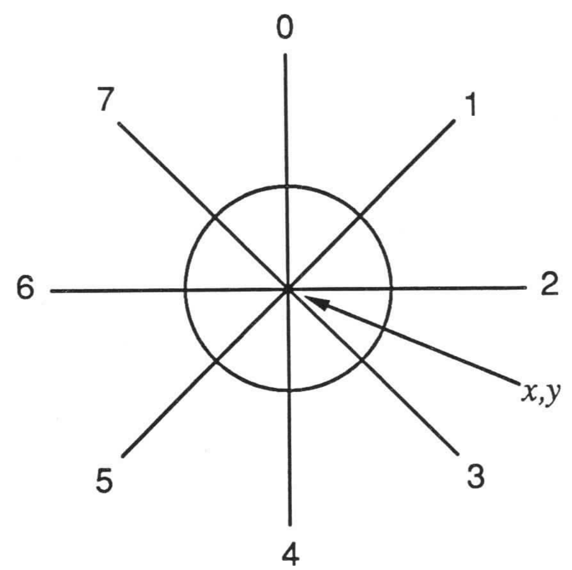
Remarks

Not supported by FutureNet OEM products.

The first dot is placed 1 dot unit from the current *x,y* coordinate, along the line indicated by parameter *a*, according to the figure below. The second dot is placed 1 dot unit from the first dot along the line indicated by parameter *b*, relative to the first dot.

This instruction updates the *x,y* value to the location of the second dot drawn.

The possible values of *a* and *b* are 0 to 7. Relative to an *x,y* coordinate they are as follows:



DEL — Delete File

Format

DEL [*drive*][*path*]*filename*

Purpose

Delete the specified file from the current directory.

Remarks

See your system documentation regarding the DEL command. A drive, path, filename and extension can follow the DEL command. The global file characters * and ? **cannot** be used.

Related Commands

CD
DIR
RM

DIR — List Directory

Format

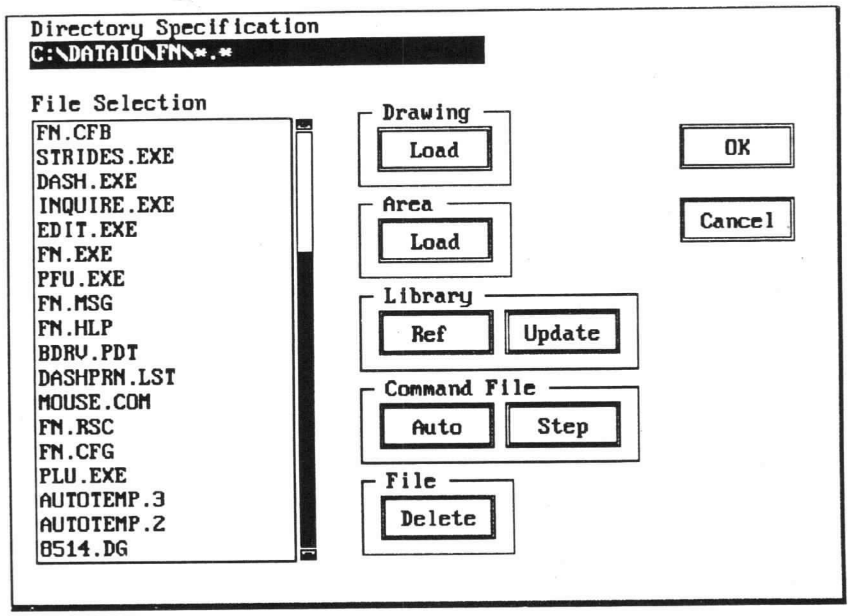
DIR *command line arguments* (PC)
ls *command line arguments* (SUN)

Purpose

List all directory entries or only specified files.

Remarks

When this command is entered, the directory dialog box is displayed. The dialog box contains a list of the files in the current directory that satisfy the file specification given as an argument for the command. This list is placed in a data window, which can be scrolled.



The **Directory Specification** field filters the list of files. To change the filter, edit the field and press ↵ . Wildcards are valid: use * for any number of symbols; ? for a single character. Drawing, Area, Library, Command File and File commands can be performed by selecting the desired file, then clicking on the desired action button.

Use the scroll bars in the file data box to display the filenames available.

Related Commands

ls

DISPGRP*n* — Display Group Setup

Format

DISPGRP1 ***comp1color,comp2color,...comp7color***
DISPGRP2 ***comp8color,comp9color,...comp15color***
DISPGRP3 ***comp16color,comp17color,...comp23color***
DISPGRP4 ***comp24color,comp25color,...comp31color***

Purpose

Change the color of the components that make up a drawing.

Remarks

There are 31 drawing components arranged in one group of seven and three groups of eight components. Each component has an assigned position within its group. The number assigned to a position specifies the color from the palette (see COLOR) assigned to the component associated with that position. To change the color of a component, assign a different color from the color palette to that position.

DISPGRP*n* is used to specify the display group and the color from the color palette. For example

DISPGRP3 ,7

will change the cursor, which is in position 2 of display group 3, to the color assigned to position 7 of the palette. Other colors are unchanged.

Similarly, entering

DISPGRP4 ,,1,8,,

with the default color palette will cause direct connection lines to be yellow, and the pin cursor to be magenta.

The following tables list the components in each display group, their position numbers, and the default color assignments. The items that are marked with an asterisk (*) are fixed colors and cannot be changed.

DISPGRP1:

Pos#	Component Name	Comp#	Default Color
1	Border	1	4
2	Status headings	2	4
3	MODE status field value	3	3
4	Other status field values	4	6
5	Signal and bus lines	5	1
6	Symbol graphics	6	1
7	Symbol boundary	7	1

DISPGRP2:

Pos#	Component Name	Comp#	Default Color
1	Alphanumeric text	8	1
2	Command line*	9	White
3	Message line*	10	White
4	Symbol reference number	11	4
5	Unused	12	
6	Tagged object	13	3
7	Unused	14	
8	Menu text*	15	Black

DISPGRP3:

Pos#	Component Name	Comp#	Default Color
1	Menu headings*	16	Black
2	Cursor	17	2
3	Line target	18	2
4	Area target	19	2
5	Rubberbanded line	20	3
6	Area definition outline	21	2
7	Full scale window	22	4
8	Preset WINDOWs	23	1

DISPGRP4:

Pos#	Component Name	Comp#	Default Color
1	Alphanumeric cursor	24	2
2	Alpha mode insert cursor	25	3
3	Direct connection line	26	3
4	Pin cursor	27	4
5	Grid	28	8
6	Unused		
7	Symbol edit instructions	30	2
8	Symbol edit target line	31	3

Note: The background color is hard-coded to palette position number 5 in the COLOR command.

Related Commands

COLOR
PALETTE

DOS — Enter DOS Command

Format

DOS [*command*]

Purpose

To run a system command or enter a DOS shell without leaving FutureNet.

Remarks

On the Sun, specify *command* with the DOS command; the results of the specified command will be sent to the window from which FutureNet was run. If no command is specified, the DOS command does nothing.

On the PC, the DOS command operates as follows:

1. FutureNet will temporarily close and a DOS shell will be opened. Within the DOS shell, commands should function normally with the following exceptions.

   - The Non-Extended Memory executable uses a "swapping" mechanism which removes FutureNet from memory, and writes it to a special file created on the hard disk named ZTC*xxxxx*.TMP (where *xxxxx* are unique characters). This "swapping" provides more memory for the command to be run. However, at least 500KB of disk space is required. If insufficient disk space is available, the DOS command will not run. (No error message is given when you are returned to FutureNet.)

   - Programs that permanently allocate memory cannot be run in the DOS shell. This includes all TSR (Terminate Stay Resident) programs and DOS functions such as MODE, PRINT, ASSIGN, GRAPHICS, SHARE, and GRAFTABL. Using these programs can cause FutureNet to fail (losing changes done during the FutureNet session).

Note: As a precaution, you may want to save your drawings before performing the DOS command.

   - Using the SET command will only set environment variables for the DOS shell—it will not effect FutureNet.

2. If a command is specified, the command will run in the DOS shell, displaying its results on the screen. When the command has completed, the message

   `Press any key to continue`

   will appear. Pressing any key will return to FutureNet.

3. If no command is specified when the DOS shell was entered, the DOS prompt will appear. The DOS prompt should include a `FutureNet` string. To return to FutureNet, use the DOS EXIT command.

Note: If the DOS prompt does not include the string `FutureNet` *in its entirety, your system's environment space is too small. You may need to enlarge the environment space; see your DOS manual.*

Do not run FutureNet from within a DOS shell. This will cause loss of logging files and cause disk clusters to be lost.

Related Commands

!

DR — Draw Rectangle in Display Units

(Symbol Definition Mode)

Format

DR ***width,length***

Purpose

Draw a rectangle of the specified length and width in display units at the current *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

width and *length* are specified in display units.

width and *length* can take any value from +127 through -128.

If the width is specified using positive numbers, the rectangle extends right from the *x,y* coordinates; if the width is specified with negative numbers, the rectangle extends left.

If the length is specified using positive numbers, the rectangle extends down from the *x,y* coordinates; if the length is specified using negative numbers, the rectangle extends up.

The symbol editor coordinate marker remains at the current *x,y* coordinates.

Related Commands

DRD

DRD — Draw Rectangle in Dot Units

(Symbol Definition Mode)

Format

DRD *width,length*

Purpose

Draw a rectangle of the specified length and width in dot units at the current *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

width and *length* are specified in dot units.

width and *length* can take any value from +127 through -128.

If the width is specified using positive numbers, the rectangle extends right from the *x,y* coordinates; if the width is specified with negative numbers, the rectangle extends left.

If the length is specified using positive numbers, the rectangle extends down from the *x,y* coordinates; if the length is specified using negative numbers, the rectangle extends up.

The symbol editor coordinate marker remains at the current *x,y* coordinates.

Related Commands

DR

DX, DY — Draw Line in Display Units

(Symbol Definition Mode)

Format

DX *length*
DY *length*

Purpose

Draw a line along the *x* or *y* axis, specifying the length in display units.

Remarks

Not supported by FutureNet OEM products.

length is in display units.

length may take any value from +127 through -128. If *length* is between +15 and -16, use the DXS and DYS (Draw Line Short) instructions to save memory.

DX *length* draws a line of the specified length along the *x* axis. If *length* is specified using negative numbers, the line is drawn left; if *length* is specified using positive numbers , the line is drawn right. The *y* coordinate is not changed by the DX instruction.

DY *length* draws a line of the specified length along the *y* axis. If *length* is specified using negative numbers, the line is drawn up. If *length* is specified using positive numbers, the line is drawn down. The *x* coordinate is not changed by the DY instruction.

After a DX or DY instruction, the symbol editor coordinate marker is located at the end of the line drawn.

Related Commands

DXD
DYD
DXY
DXS
DYS

DXD, DYD — Draw Line in Dot Units

(Symbol Definition Mode)

Format

DXD *length*
DYD *length*

Purpose

Draw a line along the *x* or *y* axis, specifying the length in dot units.

Remarks

Not supported by FutureNet OEM products.

length is in dot units

length may take any value from +127 through -128.

DXD *length* draws a line of the specified length along the *x* axis. If *length* is specified using negative numbers, the line is drawn left; if *length* is specified using positive numbers , the line is drawn right. The *y* coordinate is not changed by the DXD instruction.

DYD *length* draws a line of the specified length along the *y* axis. If *length* is specified using negative numbers, the line is drawn up. If *length* is specified using positive numbers, the line is drawn down. The *x* coordinate is not changed by the DYD instruction.

After a DXD or DYD instruction, the symbol editor coordinate marker is located at the end of the line drawn.

Related Commands

DX
DY
DXYD
DXS
DYS

DXL, DXR — Draw x to Symbol Cell Boundary

(Symbol Definition Mode)

Format

DXL
DXR

Purpose

Draw a line along the current *y* coordinate to the left or right symbol cell boundary.

Remarks

Not supported by FutureNet OEM products.

This command is particularly useful in drawing line stubs to the symbol cell boundary.

DXL draws a line along the *x* coordinate to the left symbol cell boundary.

DXR draws a line along the *x* coordinate to the right symbol cell boundary.

The symbol editor coordinate marker location is not changed by these instructions.

Related Commands

BXL
BXR
DYU
DYL

DXS, DYS — Draw Line Short in Display Units

(Symbol Definition Mode)

Format

DXS *length*
DYS *length*

Purpose

Draw a short line (+15 through -16 display units in length) along the *x* or *y* axis.

Remarks

Not supported by FutureNet OEM products.

This instruction is used for lines between +15 and -16 display units in length.

length is in display units.

For lines between +15 and -16 display units in length, using this instruction will save memory. For longer lines, use the DX or DY instruction.

DXS *length* draws a line along the *x* axis beginning at the current *x,y* coordinates. If *length* is negative, the line is drawn left; if length is positive, the line is drawn right.

DYS *length* draws a line along the *y* axis beginning at the current *x,y* coordinates. If length is negative, the line is drawn up; if *length* is positive, the line is drawn down in the symbol cell.

After a DXS or DYS instruction, the symbol editor coordinate marker is located at the end of the line drawn.

DXS and DYS are also used with Increment and Decrement modes, which are described under the SD and SI commands.

Related Commands

DX
DY
DXD
DYD
SD
SI

DXY — Draw Diagonal in Display Units

(Symbol Definition Mode)

Format

DXY *x,y*

Purpose

Draw a diagonal line, in display units, from the current *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

x and *y* are in display units, and can take any value from +127 through -128.

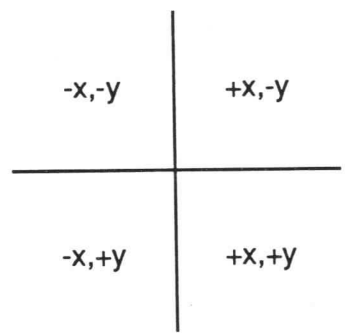
The DXY *x,y* instruction draws a line in any direction, beginning at the current *x,y* coordinates and ending the specified *x* and *y* distance from the original *x,y* coordinates.

Positive numbers draw *x* to the right; negative numbers draw *x* to the left.

Positive numbers draw *y* down; negative numbers draw *y* up.

Positive and negative *x,y* value combinations will draw lines into the quadrants indicated:

Following the DXY instruction, the symbol editor coordinate marker is



located at the end of the line drawn.

Note that this instruction can be used to draw lines along the current *x* or *y* axis.

Related Commands

DXYD
MXY

DXYD — Draw Diagonal in Dot Units

(Symbol Definition Mode)

Format

DXYD *x,y*

Purpose

Draw a diagonal line, in dot units, from the current *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

x and *y* are in dot units, and can take any value from +127 through -128.

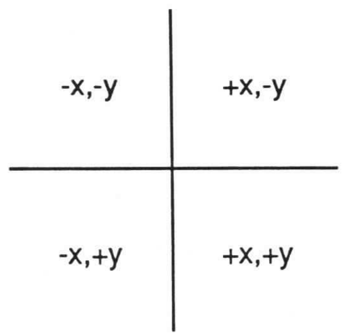
The DXYD *x,y* instruction permits a line to be drawn in any direction, beginning at the current *x,y* coordinates and ending the specified *x* and *y* distance from the original *x,y* coordinates.

Positive numbers draw *x* to the right; negative numbers draw *x* to the left.

Positive numbers draw *y* down; negative numbers draw *y* up.

Positive and negative *x,y* value combinations will draw lines into the quadrants indicated:

Following the DXYD instruction, the symbol editor coordinate marker is located at the end of the line drawn.



Note that this instruction can be used to draw lines along the current *x* or *y* axis.

Related Commands

DXY
MXYD

DYU, DYL — Draw y to Symbol Cell Boundary

(Symbol Definition Mode)

Format

DYU
DYL

Purpose

Draw a line along the current *x* coordinate to the upper or lower symbol cell boundary.

Remarks

Not supported by FutureNet OEM products.

This command is primarily used to draw line stubs to the symbol cell boundaries.

DYU draws a line along the *y* coordinate to the upper symbol cell boundary.

DYL draws a line along the *y* coordinate to the lower symbol cell boundary.

The symbol editor coordinate marker location is not changed by these instructions.

Related Commands

DXL
DXR
BYU
BYL

'E — Erase Alphanumeric Field

Format

'E

Purpose

Erase the current alphanumeric field.

Remarks

An alphanumeric field can be erased by placing the graphics cursor on the desired field and entering the 'E command or by entering the 'E command while a field is tagged, which erases the tagged field and cancels the tagging.

The entire field is erased.

Related Commands

'C
'M
'R
'I
Esc

.E — Erase Symbol

Format

.E

Purpose

Erase a symbol from the drawing.

Remarks

A symbol does not have to be tagged to be erased, as long as the cursor is located somewhere within the symbol boundary.

The entire symbol cell, including the alphanumeric fields and reference number, is erased from the drawing. Interconnects to the symbol are not changed.

When no symbol is tagged or the cursor is not inside a symbol when this command is entered, the command is ignored and the system displays: `Cursor not in symbol`

The graphics cursor location does not change when the symbol is erased.

Related Commands

.C
.M
.L

/E — Erase Line

Format

/E

Purpose

Erase the line segment or the portion of a line segment overlapped by a new line segment.

Remarks

The /E command initiates line erase from either MENU or FAST drawing.

/E works like /L except that it removes lines rather than draws them.

/E erases any portions of a permanent line that are overlapped by a routing segment. The line start point is relocated to the end of the routing segment when the command is entered, so additional connecting segments may be erased.

The line being erased may be of any type, not necessarily the line type displayed in the LINE status field.

Note that routing segments drawn by the /E command, whether drawn over existing line segments or not, can be made permanent by terminating the /E operation with the /L command. That is, terminating /E with /L cancels the erase operation and makes all routing segments permanent lines.

The mode status field shows LINE.

Related Commands

/L
/LE
/EL
/EN
/ES

/EL — Erase Line Segments

Format

/EL

Purpose

Erase all interconnected line segments between two points of connection.

Remarks

/EL traces a line and erases all segments until a connection with a pin, a junction, an interconnection, or another line is encountered. A line interconnection is indicated by an interconnect dot where two lines meet or by a "T" connection. The "T" may or may not have an interconnect dot. The interconnect dot is not erased.

Temporary lines will also be deleted if encountered.

The line being erased may be of any type, not necessarily the line type displayed in the LINE status field.

The location of the graphics cursor does not change.

Related Commands

/E
/LE
/EN
/ES

/EN — Erase Line Network

Format

/EN

Purpose

Erase a line and all interconnecting lines.

Remarks

The graphics cursor is the starting point for the erase line network operation.

The line is traced and erased along with all interconnected lines. Interconnect dots are erased.

Lines are erased along any branch of the network and erasing stops when line ends are reached, including symbol cell boundaries and alphanumeric fields (signal names). Lines emerging from interconnect dots and "T" intersections will be erased.

Temporary lines and direct connect lines (see /C command) will also be deleted.

If the cursor is on an electrically signficant line type (/1 or /2), only electrically significant line types are traced and erased.

If the cursor is on a line type that is not electrically significant, /EN will erase the line segment the cursor is located on, and any electrically significant lines networked to the segment. The line type erased is not necessarily the line type displayed in the LINE status field.

Related Commands

/LE
/E
/EL
/ES

/ES — Erase Line Segment

Format

/ES

Purpose

Erase the permanent or temporary line segment on which the graphics cursor is located.

Remarks

The cursor may be located anywhere on the line segment to be erased.

The line segment is erased to an intersection, that is, a corner, symbol boundary, interconnect dot, or "T" connection.

If the cursor is not on a line segment, the command is ignored. If the cursor is at the intersection of two lines, the command is ignored and the system displays

```
Ambiguous cursor location
```

Temporary lines resulting from rubberbanding will also be erased by the /ES command when the cursor is located on them. If the cursor is located on a temporary line which also overlays a permanent line segment, /ES erases only the temporary line.

When temporary lines are touched, they will become dotted on monochrome systems or change color on color systems.

The line being erased may be of any type, not necessarily the line type displayed in the LINE status field.

The location of the graphics cursor does not change.

The /ES command is equivalent to the right mouse button when using FAST drawing mode.

Related Commands

/E
/EL
/EN
/LE
/ET

/ET — Erase Temporary Lines

Format

/ET

Purpose

Erase all temporary lines in the drawing.

Remarks

The connection information maintained by the temporary lines is lost when this command is entered. Therefore, permanent lines should be drawn to replace temporary lines before this command is entered.

/ET can be used at any time. The temporary lines do not have to be rerouted and made into permanent legal connections before using this command.

When entered, the system will prompt

```
OK to erase all temporary lines (Y/N)?
```

Note: It is not necessary for the cursor to be on a temporary line for this command to take effect.

Related Commands

/ES
/P

[ERASE — Erase Area

Format

[ERASE

Purpose

Erase all objects completely contained within an area.

Remarks

The area must already have been defined, though it does not have to be tagged; see [D. The cursor may be located either inside or outside the area when the [ERASE command is entered.

When the [ERASE command is entered, the system displays

```
OK to erase area (Y/N)?
```

When the response is Y for yes, all objects within the area definition are erased and the area definition is lost.

When the response is N for no, the area is maintained.

[ERASE includes all symbols, alphanumeric fields, and lines completely inside the area boundary, as well as the area definition. Any lines intersecting the area boundary are broken at the boundary and the segments inside the area are erased.

If an area is erased during a MOVE/COPY/ERASE operation, control returns to the previous mode.

Related Commands

[D

ERASE

See CLEAR.

Esc Key — Enter/Exit Alphanumeric Mode

Key

[Esc]

Purpose

Toggle alphanumeric mode.

Remarks

Pressing [Esc] toggles alphanumeric mode.

ALPH appears in the MODE status field when alphanumeric mode is enabled.

If the graphics cursor is located in an existing alphanumeric field when alphanumeric mode is entered, the cursor is positioned on the character at the left end of the field. If the graphics cursor is not located in an existing field when alphanumeric mode is entered, a new alphanumeric field will be created. The graphics cursor marks the lower left corner of the first character position in the new field, and the alphanumeric cursor appears at this character location.

Entering and immediately exiting alphanumeric mode will not affect an existing alphanumeric field nor create a new alphanumeric field.

See the chapter "Mouse, Modes, and Cursors" in the *FutureNet User Manual* for more information on using alphanumeric mode.

Note: Esc is also used to cancel the menus and dialog boxes.

EXEC — Single-step Command Execution

Format

EXEC *filename*

Purpose

Invoke single-step command file execution.

Remarks

Before EXEC can be entered, a filename is required. EXEC will search for command files with the default .cmd command file extension unless a different file extension is specified.

This command can also be used in the Symbol Definition Mode.

EXEC places each command on the command line and waits for a ↵ by the operator to enter it. After the command is entered, the next command in the sequence is placed on the command line.

The command file named must be a standard ASCII text file created by a text editor and must follow the command file format. That is, commands on the same line must be separated by semicolons (;), a line cannot exceed 256 characters, and each line must end with a complete command--they cannot wrap to the next line.

EXEC can be interrupted momentarily at any line by typing a command on the command line and entering it. After the user-initiated command(s) has been entered, the bypassed command file command is again placed on the command line.

Single-step execution invoked by the EXEC command can be changed to automatic execution by typing AUTO without a command filename. See the AUTO command for more information.

A second command file can be initiated from within the first by including the EXEC *filename* command within the first command file. Naming one command file within another command file stops the execution of the first command file and starts execution of the second command file. It does not clear the current drawing work space, and there is no return to the first file.

Any command following an EXEC *filename* command will not be entered.

Related Commands

AUTO
NOTE
PAUSE
STOP
VIEW

EXPORT — Enter Post Processing Menus

Format

EXPORT

Purpose

Enter the post processing menus to create custom post processing scripts and generate reports.

Remarks

The EXPORT menus can also be accessed by selecting **EXPORT Generate Reports** from the main menu.

Complete information on operating the EXPORT menus is given in the *FutureNet Post User Manual,* and the *FutureNet to EDIF 2 0 0 Netlist Writer User Manual.*

The following reports can be generated through the EXPORT menus:

Menu Selection	Report Generated
Connect Design	Drawing Connectivity Model (.dcm)
Check Design Rules	Design Rule Check Report (.drc)
Generate Pinlist	Enhanced or compatible pinlist (.pv4 or .pin)
Generate Netlist	Enhanced or compatible netlist (.nv4 or .net)
Generate Parts List	Parts List (.prt)
EDIF 2 0 0 Netlist Generator	EDIF 2 0 0 netlist

'F — Find Alphanumeric String

Format

'F ["]*string*["]

Purpose

Search the drawing, starting at the graphics cursor location, for any alphanumeric field beginning with the specified character *string*.

Remarks

A search string must be specified. If one is not specified, the system will prompt for one and position the command line cursor at the end of the command, ready to enter the search string. To include leading or trailing spaces in the string, use the double quotations. Otherwise FutureNet will search for the string from the first non-blank character to the last non-blank character.

The drawing is searched from the current graphics cursor position, left to right and top to bottom, for an alphanumeric field starting with the specified character string. If the string is not found below the cursor position, then the search will resume at the top of the drawing.

Only fields starting with the specified string are located.

Uppercase and lowercase occurrences of the specified string are located, as well as partial strings at the beginning of the text field.

The display shifts to center on fields that are located outside the current drawing area.

Since the original command remains on the command line, pressing ↵ again causes the system to search for the next occurrence of the string.

If the requested string is not found, the system displays

```
Not found
```

To search and replace at the same time, use a command list of 'F and either 'R or 'I as follows:

'F ["]*string*["];'R ["]*string*["]

searches for the 'F string and replaces it with the 'R string.

'F ["]*string*["];'I ["]*string*["]

searches for the 'F string and replaces it with the 'I string, incrementing the value specified by 'I.

Note: The entire field is replaced by the 'R or 'I string, not just the string specified with 'F.

For example, when the 'I string is a circuit designator, entering

'F UNNN;'I U1

replaces the first found occurrence of UNNN with U1, and increments the command on the command line to 'F UNNN;'I U2. Pressing ↵ again enters this command, increments the number, and so on.

'F is especially useful when looking for signal names causing errors in the Pinlist Generator or for finding circuit designator fields that need to be filled in.

If a double quote (") or backward slash (\) is to be included in the string, precede the character with a backward slash. For example, to find the string

"status\pin1"

use the following 'F command:

'F "\"status\\pin1\""

Related Commands

'FA
'I
'R

'FA — Find Attribute

Format

'FA *name* | *number* [["]*string*["]]

Purpose

Search the drawing for alphanumeric fields having the specified attribute assignment (name or number) and, if included, beginning with the specified *string*.

Remarks

An attribute (name or number) must always be specified. Specifying a string is only required when the 'FA command is to be used as part of a command list, otherwise it is optional. To include leading or trailing spaces in the string, use the double quotations. Otherwise FutureNet will use the string from the first non-blank character to the last non-blank character.

The drawing is searched from the current graphics cursor position, left to right and top to bottom, for an alphanumeric field with the specified attribute and, if included, the specified character string. If the graphics cursor is within an alpha field, the search begins at the top of that field. If the attribute and/or string is not found below the cursor, then the search will resume at the top of the drawing.

Uppercase and lowercase occurrences of *string* are located, as well as partial strings at the beginning of the text field.

The FutureNet drawing screen shifts to center on fields with the attribute/string combination that are located outside the currently viewed drawing area.

Since the original command remains on the command line, it can be entered as often as required.

If the requested attribute is not found, the message Not found appears. The same message appears when a string is specified and the system finds the specified attribute but not one with a string match.

To search and replace at the same time, use a command list of 'FA *attribute string* (note that in a command list, both the attribute and string are required parameters) and either 'R or 'I to search and replace as follows:

'FA *attribute string;*'R ["]*string*["]

searches for the attribute and string specified in 'FA *attribute string* command and replaces the current string with the string specified in the 'R *string* command.

'FA *attribute string;*'I ["]*string*["]

searches for the 'FA *attribute string* and replaces it with the 'I *string,* incrementing the value specified by 'I.

For example, when the 'I *string* is a circuit designator, entering

'FA LOC UNNN;'I U1

will find the first occurrence of the text field UNNN, with attribute LOC, and replace it with U1. At the same time, the 'I command increments the string specified in 'I *string* command from 'FA LOC UNNN;'I U1 to 'FA LOC UNNN;'I U2. Entering the command list again locates the next occurrence of an alphanumeric field with the LOC attribute and UNNN text field and inserts the incremented U value, and so on.

'FA is also useful when looking for signal names causing errors in the Pinlist Generator or for finding component value fields that need to be filled in.

If a double quote (") or backward slash (\) is to be included in the string, precede the character with a backward slash. For example, to find the string

"status\pin1"

use the following string:

"\"status\\pin1\""

Related Commands

'CH A
'D
'F
'I
'R
'S

.F — Create Functional Block

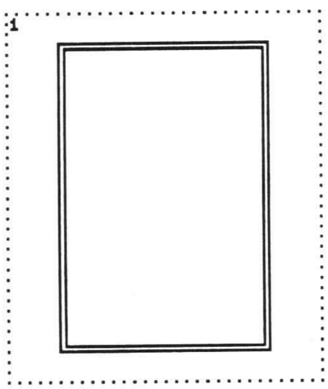
Format

.F *w,h[,ws[,hs]]*

Purpose

Create a functional block symbol with a double outline.

Remarks



w is the width of the symbol rectangle expressed in display units; *h* is the height of the rectangle expressed in display units.

ws is the distance from the sides to the symbol cell. If *ws* is not specified, the system default of 2 display units is used.

hs the top and bottom of the rectangle to the symbol cell. If *hs* is not specified, the system default of 1 display unit is used.

Specifying 0,0 for *ws,hs* causes the symbol boundary to overlap the symbol cell. This means that wires and buses can be drawn right to the symbol cell, eliminating the need for pin stubs. However, if you prefer using pin stubs, the pin stub commands work with the .F command.

The combined numeric value of *w* and two times *ws* (one *ws* per side) cannot exceed 127. Similarly, the combined numeric value of *h* and two times *hs* cannot exceed 127. If the value exceeds 127, or if 0 is entered for any value, the system will display

```
Value out of range
```

and the command will be ignored.

The symbol is created with the top left corner of the symbol cell positioned at the graphics cursor location.

If another symbol, an alphanumeric field, a line, or the drawing edge is in the area to be allocated to the new symbol cell, a symbol boundary conflict will exist, preventing the symbol from being placed. The message

```
Symbol boundary conflict
```

will be displayed. The conflict must be resolved before the new symbol can be entered into the drawing.

A new reference number is assigned to the block symbol.

.F uses symbol definition instructions to generate a symbol. Symbol definition instructions are discussed in the symbol definition chapters in the *FutureNet User Manual.* The list of instructions used to define the block symbol can be viewed and edited by entering Symbol Definition Mode (.S command). However, once a symbol has been edited in Symbol Definition Mode, a bit is set that prevents it from being edited using drawing mode commands, such as the Add/Replace Pin Stub commands. In order to use these commands on symbols that have been edited in Symbol Definition Mode, enter the .SBS command prior to exiting Symbol Definition Mode.

Related Commands

.A
.B
.–
.-AI
.D

FAST

Format

FAST [ON | OFF]

Purpose

To select FAST or MENU mode.

Remarks

FAST ON selects FAST mode, which operates as described below.

FAST OFF selects MENU mode, which allows access to the menus with the right mouse button.

FAST without parameters toggles between the MENU and FAST modes.

The FAST mode makes the most frequently-used drawing functions accessible from the mouse.

The mouse in FAST mode differs from MENU mode in the following ways only:

Left button	No difference
Middle	Add/Remove Interconnect Dot. When the graphics cursor is on a /1 or /2 type line, the middle button toggles an interconnect dot (/D).
Right	Erase Line Segment. The right button erases the line segment at the graphics cursor (/ES).

All other modes are accessible from the FAST mode, allowing you to perform move, copy and erase operations, and enter commands on the command line.

The FAST mode cursor is the graphics cursor.

You can access the menus from FAST mode without exiting FAST mode by entering MENU.

Related Commands

MENU
/D
/ES

FILE — Display Filenames

Format

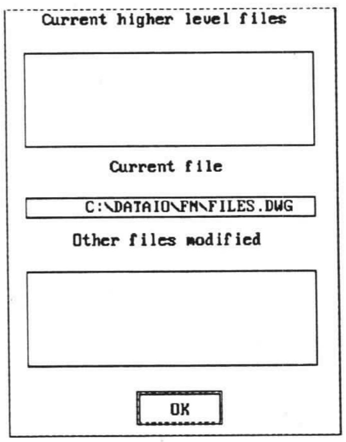
FILE[S]

Purpose

Display the name of the current drawing file.

Remarks

Filenames are displayed in the FILE dialog box:



In a structured design, FILE displays the names of all files accessed in the current session and their status since the last CLEAR, SAVEALL, or LOAD command. The files are displayed in the following order:

- **Current higher level files** — The current path of files is displayed from the root drawing to the current file, excluding any drawing levels that were bypassed with the #D command.
- **Current file** — The current file.
- **Other files modified** — Any other files in the drawing structure that have been changed, but not yet saved.

To the left of each filename the word SAVE is displayed for any file that was modified in this session. Files that have been viewed but not modified have no status next to their name.

If there are no filenames to display, or if the loaded file has a **.bak** extension, the following message appears:

`No current filename`

Exit the FILE dialog box by either selecting the OK button, by pressing the left mouse button outside of the dialog box, or by pressing Esc .

Related Commands

#D
#L
#R
#U
SAVE
SAVEALL
LOAD

fn — Start FutureNet

Format

fn [*drawing_file*]
fn [@][*command file*]

Purpose

Start FutureNet and, optionally, load the specified drawing or run a command file.

Remarks

This command is entered from the operating system command line. If filename extensions are not used, FutureNet first assumes a command file extension (.cmd). If the file does not exist with a .cmd extension, FutureNet assumes a drawing file extension (.dwg). To load files with extensions other than .cmd or .dwg, the extension must be specified explicitly.

If the @ character is used, the specified command file runs in EXEC mode. Otherwise, the command file runs in AUTO mode. See the EXEC and AUTO commands for more details.

Note: You cannot run fn from a DOS shell started from a FutureNet drawing session; you can lose files and autologging.

Related Commands

AUTO
EXEC
NOTE
PAUSE
STOP
VIEW

.G — Change Symbol Type

(Symbol Definition Mode)

Format

.G

Purpose

Change a symbol from a circuit type symbol to a graphic type symbol, or vice versa.

Remarks

Not supported by FutureNet OEM products.

There are two types of symbols, circuit and graphic. Circuit symbols are electrically significant, graphic symbols are not. The symbol cell boundaries and reference numbers are displayed in the drawing for circuit symbols, but not for graphic symbols. Both types of symbols are created and handled the same by FutureNet, but post processors use them differently. Graphic symbols are ignored by post processors.

.G toggles the setting for a symbol.

While the setting may not be immediately obvious in Symbol Definition Mode, when you re-enter drawing mode, cell boundaries and reference numbers are affected.

GRID — Display Grid/Toggle Grid Snap

Format

GRID [ON | OFF]
GRID *x[,y]*
NOGRID
GRID SNAP [ON | OFF]

Purpose

Toggle the grid setting.

Remarks

GRID is valid in both drawing and Symbol Definition modes.

A grid is a framework of horizontal and vertical tic marks. Entering GRID without specifying any parameters toggles the grid display on and off. A default grid of 3 x 3 display units is used when no grid size has been previously specified.

A grid size other than the default can be selected by specifying just the *x* value or both the *x* and *y* values. For example

GRID x

or

GRID x,y

If only *x* is specified, *y* assumes the value assigned to *x*.

The grid specification is saved with a drawing. If on, the drawing will be saved with the grid on; if off, the drawing will be saved with the grid off.

The grid is displayed in full and intermediate zoom levels. The grid can be printed by specifying R (reference parameter) in the PRINTOPT command or selecting the Print Non-Printable Attributes checkbox in the PRINTOPT dialog box.

Entering NOGRID turns off the grid display.

GRID SNAP without any parameters toggles the snap-to-grid setting. GRID SNAP [*ON* | *OFF*] enables or disables grid snap as specified.

The snap-to-grid feature will constrain line drawing, symbol placement, and text placement to the current grid.

Related Commands

PRINTOPT
PROFILE

HELP — Read Online Command Reference

Format

HELP [["]*help_string* ["]]
[F1]

Purpose

Accesses online help.

Remarks

The HELP command provides quick access to an online command reference. The text for the online reference is similar to the text for the written *FutureNet Command Reference*.

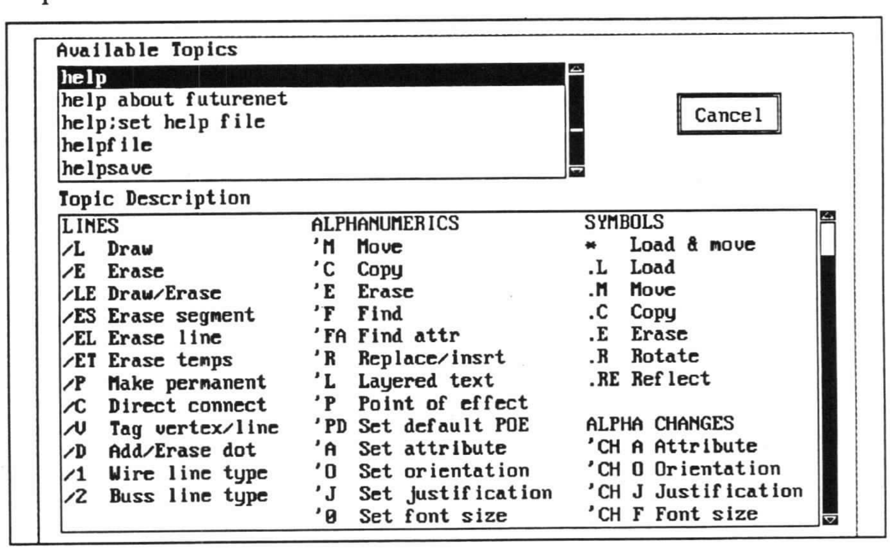
The HELP command is available in MENU, FAST or Symbol Definition modes.

The default command reference file is **fn.hlp**, which is distributed with the FutureNet product. FutureNet accesses this file during initialization, searching the system using the following pattern:

- Current directory
- Directory containing the FutureNet executable
- Directories on the system PATH

The help file can also be set by the HELPFILE command. Only one command reference file can be active at any time.

To invoke the HELP dialog box, type either **HELP** optionally followed by a string on which help is needed, or [F1]. The string can be any FutureNet command as well as some phrases which describe the command. If no string is provided, the HELP dialog box appears with help on the most common commands.



The HELP dialog box contains the following items:

- **Available Topics** — This data window contains a list of all topics which can be accessed using the HELP command. The list contains all FutureNet commands plus some equivalent phrases which describe the commands.

  If a command is selected on the command line, it is highlighted in the topics data window.

  To select a topic, use the scroll bars to the right of the data window to scroll the list and select one of the topics by placing the mouse cursor on the desired item and clicking the left mouse button. The command reference text for the selected command will appear in the lower data window.

  To use the keyboard to select the item, see the *FutureNet User Manual* for information on keyboard equivalents for the mouse.

- **Topic Description** — The lower data window will contain the actual reference text for the chosen topic. Use the scroll bars to the right of the data window to scroll the reference material.

- **Cancel** — The cancel button is used to leave the HELP dialog box and return to previous mode.

  You can also exit the HELP dialog box by pressing ESC or clicking the mouse cursor outside of the dialog box.

Related Commands

HELPFILE

HELPFILE — Open Online Command Reference File

Format

HELPFILE *help_filename*

Purpose

Specifies a help file for use by the FutureNet HELP command.

Remarks

The HELPFILE command can be used to set the current command reference filename for the HELP command.

Note: *Command reference files are not ASCII files. Special tools are required to create a command reference file. These tools can be made available upon request.*

The HELPFILE command requires a file specification. The file specification is searched for using the following pattern:

- File specification as given
- Directory containing the **fn** executable file
- Directories on the system PATH

Only one command reference file can be active at any time.

Related Commands

HELP

HELPSAVE — Save Custom Menu or Help Screen

Format

HELPSAVE *name*

Purpose

Save the current drawing as a custom menu or help screen to the update library.

Remarks

name is a required parameter.

Custom menus and help screens are saved into the update library which is specified with the .LIB command. If update library has not been specified the following error message will appear:

```
Update library not specified
```

The **?** command is used to retrieve the custom menus. When the **?** command is used without a parameter, then the symbol **?HELP** is retrieved. To save this default menu, use the following **HELPSAVE** command:

HELPSAVE ?HELP

Refer to the *FutureNet User Manual* for more information on creating custom menus and help screens.

Related Commands

?
.LIB

HOME — Home Cursor

Format HOME

Keys

PC	SUN
Home	R7

Purpose

Center the work area around the cursor location.

Remarks

The cursor location does not change with this command. In full or intermediate zoom, the display shifts so that the cursor location is centered on the screen.

When the cursor is at, or very close to, the edge of the drawing, the cursor cannot be centered on the screen when HOME is entered, because the system does not move the display beyond the edge of the drawing.

The **HOME** command has no effect in fit zoom.

HOME can be entered in command files.

Related Commands

CURSOR
WINDOW
PAN

HOME

(Symbol Definition Mode)

Format HOME

Keys

PC	SUN
Home	R7

Purpose

Reposition the display window with the current symbol editor coordinate marker location as the center of the graphics portion of the screen.

Remarks

Not supported by FutureNet OEM products.

HOME, and Home or R7 are primarily used with symbols too large to display on one Symbol Definition screen.

'I — Insert Alphanumeric String and Increment Numbering

Format

'I ["]*string*["]

Purpose

Insert an alphanumeric string in the field at the cursor location and automatically increase the numeric part of the *string* by one.

Remarks

A string must be specified. To include leading or trailing spaces in the string, use the double quotations. Otherwise FutureNet will use the string from the first non-blank character to the last non-blank character.

'I replaces an existing alphanumeric field or creates a new one.

The number must be at the end of the string, for example

'I U1

After the command is entered, the command and the incremented numeric value are displayed on the command line, for example

'I U2

Moving the cursor and entering the command again causes the system to insert the incremented string at the new cursor location and repeat the process.

This command is especially useful when labeling pin fields, signal lines, etc.

'I will not increment multiple instances of numerics in strings; only the last instance is found. For example, incrementing U1a1 will result in U1a2, not U2a2.

If a double quote (") or backward slash (\) is to be included in the string, precede the character with a backward slash. For example, to insert the string

"status\pin1"

use the following string:

'I "\"status\\pin1\""

Related Commands

'F
'FA
'R

.I — Enter/Exit Insert Mode

(Symbol Definition Mode)

Format

.I

Purpose

Toggle symbol definition insert mode.

Remarks

Not supported by FutureNet OEM products.

In insert mode, an instruction entered on the command line is inserted above the instruction currently on the target line. When not in insert mode, the current target line instruction moves up and the instruction entered on the command line becomes the new target line instruction.

In insert mode, the horizontal line below the target line appears as a dotted line.

Related Commands

.D

ID — Insert Dot Matrix

(Symbol Definition Mode)

Format

ID ***row1,row2,row3,row4***

Purpose

Insert the specified four-by-four dot matrix at the current *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

The matrix is a four-by-four square of dots, where each dot may be on or off. The parameters, *row1* through *row4*, identify the rows of the matrix; *row1* is the top row, *row2* is the next, etc. The matrix is placed at the current *x,y* coordinates, beginning at row 1 dot 1, and moving to the right and down.

Parameters are specified in hexadecimal (0 through F), but are translated by the instruction into their 4-bit binary equivalents 0000 through 1111, where 0s indicate dots that are not illuminated and 1s indicate dots that are illuminated. Hexadecimal and binary equivalents are provided in the table accompanying this instruction.

Hexadecimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

For example, the following matrix has all dots off:

```
0000
0000
0000
0000
```

In hexadecimal, the command would be coded as

ID 0,0,0,0

The following matrix has all dots on:

```
1111
1111
1111
1111
```

In hexadecimal it would be coded as

ID F,F,F,F

Once the matrix has been defined, you can use it repeatedly during the current design session. The command

IS*xx* 10

inserts a copy of the current matrix at the location given in the IS*xx* command. See the ISxx command for more information. Using the IS command instead of repeated ID commands will save memory.

The symbol editor coordinate marker location is not changed by this instruction.

Related Commands

ISxx

IG — Insert Graphics Element

(Symbol Definition Mode)

Format

IG ***number***

Purpose

Insert the specified graphics element at the current *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

Number is a required parameter. Refer to the appendix in the *FutureNet User Manual* that lists the available graphics elements by index number.

FutureNet includes a built-in table of commonly used graphic elements, such as AND gates, OR gates, etc. They are referenced by an index number. When the IG instruction is entered, the specified symbol is displayed at the cursor location.

Graphic elements are placed at the cursor location beginning at the top of the element and moving down and to the right. The symbol editor coordinate marker location will not be changed by this instruction.

Related Commands

ISxx

Ins and R11 Keys — Insert Alphanumeric Character

Keys

PC	SUN
Ins	R11

Purpose

Toggle alphanumeric insert mode.

Remarks

The default setting is Insert Off.

Pressing Ins or R11 toggles the setting.

The alphanumeric cursor is a solid underscore for Insert Off, and a dotted underscore for Insert On.

Characters are inserted at the alphanumeric cursor location as they are typed. They are inserted according to the justification set for the current alphanumeric field (left, right, or center).

Once the field encounters another drawing element (symbol, alphanumeric field, etc.), or the edge of the drawing or screen, no more characters can be inserted. Move the alphanumeric field to allow the additional characters to be inserted.

ISxx — Insert Symbol Element

(Symbol Definition Mode)

Format

IS*xx number*

Where *xx* is

TC | TR | RC | BR | BC | BL | LC | TL | C

Purpose

Insert the specified symbol element at the current *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

Number is a required parameter. Refer to the appendix in the *FutureNet User Manual* that lists the available graphics elements by index number.

FutureNet includes a built-in table of commonly used dot-matrix symbol elements, mostly arrows of varying shapes and sizes. They are referenced by an index number. When the IS instruction is entered, the specified symbol is positioned at the current *x,y* coordinates according to the arguments specified with the IS instruction.

ISTL	*x,y* at top left of symbol
ISTC	*x,y* at top center of symbol
ISTR	*x,y* at top right of symbol
ISRC	*x,y* at right center of symbol
ISBR	*x,y* at bottom right of symbol
ISBC	*x,y* at bottom center of symbol
ISBL	*x,y* at bottom left of symbol
ISLC	*x,y* at left center of symbol
ISC	*x,y* at center of symbol

See the Select Symbol Element Table (ST) instruction for information on selecting symbol element tables.

The symbol editor coordinate marker location is not changed by the instruction.

Related Commands

ID
IG
ST
RS

'J — Set Justification

Format

'J [L | R | C]

Purpose

Set the default justification to be assigned to future alphanumeric fields.

Remarks

This command sets the default justification that will be assigned to new alphanumeric fields. The justification for existing alphanumeric fields is not affected. The minimum entry is 'J, which cycles through the options, from left to right to center. The justification is shown on the third line of the ALPH status field.

Left, L, enters type to the right of the initial cursor position. Right, R, enters type to the left of the cursor position. Center, C, spaces the type evenly on either side of the initial cursor position.

Note that the 'B command displays the field boundary and the point of justification, which is the dot appearing at the lower left, center, or right of the field, depending on the justification set for the field.

Related Commands

'B
'CH J

/J — Draw Junction Segment

Format

/J [*direction*] [,[*size*] [,*type*]]

Purpose

Draw a curved or diagonal line segment.

Remarks

The /J options must be specified in order with commas (,) used as delimiters. *Direction* can be specified without any other options. If you want to specify *size*, you must either specify *direction* or use a comma; if you want to specify *type*, you must either specify the *direction* and *size* or use commas.

Used to join signal wires to buses, as an alternative to orthogonal (right angle) signal/bus connections.

Direction is the direction the segment is to follow from the current cursor position. The direction can be

DR (or RD)	Down and Right
UR (or RU)	Up and Right
DL (or LD)	Down and Left
UL (or LU)	Up and Left (default)

When drawing diagonal segments, the order of the direction indicators doesn't matter; that is, DR and RD are equivalent.

When drawing curved segments, the order is important because it indicates the segment's initial direction. For example, DR causes the segment to be drawn down and then right, that is, counterclockwise; RD causes the segment to be drawn right, then down, that is, clockwise.

Size is the segment's size in display units. It can be from 1 through 10.

Type is the type of segment you want drawn. D is diagonal; C is curved.

Any unspecified parameters default to the values specified the last time /J was used during the current session. If /J has not been used during the current session, the defaults are

/J UL,3,D

Junctions lines are actually graphic symbols. They have undisplayed square boundaries in the size specified and direct connections between opposite corners. Other objects, such as alphanumeric fields, symbols, and lines cannot be drawn within these boundaries. You should use symbol commands to manipulate them, (for example, .E to erase one), rather than line commands. However, the line erasing commands /ES, /EL, and /EN usually work to erase the segments.

Related Commands

.E
/ES
/EL
/EN

'K — Cancel Alphanumeric Field Tag

Format

'K

Purpose

Cancel an alphanumeric field's tag and return to the previous mode.

Remarks

The 'K command can be performed anytime during tag and drag, and is equivalent to pressing the right mouse button.

The location of the graphics cursor will be where the tag cursor was located at the time 'K was entered. The tag cursor will disappear and the graphics cursor will appear in its place.

Related Commands

'C
'M
'E

.K — Cancel Symbol Tag

Format

.K

Purpose

Cancel a symbol tag and return to previous mode.

Remarks

The .K command can be performed any time during tag or drag of a symbol. The tag will be canceled and the drawing will remain unchanged. The tag cursor will disappear and the graphics cursor will appear in its place.

Related Commands

.C
.E
.M

/K — Escape Line Drawing

Format

/K

Purpose

Escape line drawing and return control to the previous mode.

Remarks

When /K is entered, the line drawing cursor is replaced by the graphics cursor. The cursor location does not change.

All routing segments disappear.

When line drawing is terminated, the system returns to the mode that was in effect when line drawing was initiated.

The /K command is equivalent to the right mouse button during line drawing.

[K — Cancel Area Definition or Area Tag

Format

[K

Purpose

Erase the area definition or cancel tag and drag, depending on the system mode.

Remarks

The area must already have been defined; see [D. If not, the command is ignored.

[K erases the existing area definition boundary.

When [K is entered or the right mouse button is pressed while in MOVE/COPY/ERASE mode, the mode is canceled and the system is returned to MENU or FAST mode. Entering [K again cancels the area definition.

Since only one area can be defined at a time, the cursor can be located anywhere in the drawing when [K is entered.

Related Commands

[C
[D
[M

KEY — Assign a Function to a Function Key

Format

KEY <*key*> *command(s)*

Purpose

Assign a function to one of the "F" function keys (F1 - F10).

Remarks

<*Key*> is the name of the function key, for example, **<F2>**. Angle brackets are required.

Command(s) is the function the key is to perform; any commands that can be used in a command file or command list can be assigned to a function key. *Command(s)* can be up to 127 characters long.

For example,

KEY <F1> .=;down 3

will allow the **<F1>** key to add bus pin stubs to a functional block and move down 3 display units, ready to add the next bus pin stub.

If *command(s)* is omitted, **KEY** prompts with the current key definition, leaving the cursor at the start of the command string.

This command updates the current user profile.

Function key assignments are for the current editing session only, unless they are entered into an **fnpro.cmd** file.

Related Commands

PROFILE

!LB, !MB and !RB— Emulate a Mouse Button Click

Format

!LB
!MB
!RB

Purpose

Allow emulation of a mouse click from the keyboard.

Remarks

!LB emulates the left mouse button click.

!MB emulates the middle mouse button click.

!RB emulates the right mouse button click.

The commands are all valid in MENU or FAST drawing modes. !RB brings up the Symbol Definition menus in Symbol Definition Mode.

These commands are not operational in dialog boxes, such as PROFILE, PALETTE or DIR. Use the key bindings described in the *FutureNet User Manual* for operation of a dialog box from the keyboard.

Note: Early versions of FutureNet on the Sun used the commands LB, MB and RB, instead of !LB, !MB and !RB. These commands are treated as synonyms.

#L and #R— Move Left or Right in Design Hierarchy

Format

#L
#R

Purpose

Move between neighboring sheets in a drawing set.

Remarks

#L and **#R** only allow movement between adjacent drawings in the same functional block drawing set, and only after the following sequence of events has happened:

1. The cursor is located on one of the alphanumeric fields associated with the drawing set (these fields must be in a functional block symbol at the drawing level immediately above the drawing set and the alphanumeric fields must have a FILE or FILN attribute assignment).
2. One of the files in the functional block drawing set was accessed by entering [Ctrl] – [PgDn], [Ctrl] – [R15] or **#D** without a filename. If the functional block drawing set was accessed using **#D** *filename,* **#L** and **#R** will not work.

Once one of the drawings in the functional block drawing set has been accessed, **#L** and **#R** can be used to move between adjacent files in the set.

If there is no adjacent file to move to, the higher level drawing is redisplayed with the graphics cursor at the first or last file name in the list.

To move to a file in the functional block drawing set that is not adjacent to the current file, enter **#L** or **#R** until the desired file is reached or return to the next higher level using **#U** and specify the desired file.

These commands are provided for moving between adjacent files in a multifile drawing set.

Related Commands

#D
#U

'L — Create, Edit or View Layered Text

Format

'L

Purpose

Create a new property sheet or display an existing one for a single alphanumeric field in order to add, edit, or view layered text, or view all layered text entries of the same class for an entire symbol.

Remarks

Not supported by FutureNet OEM products.

Layered text is used to add drawing information required by design tools and translators. Layered text is stored on what are called property sheets that are attached to displayed alphanumeric fields.

Layered text may be added to the text associated with a symbol. When the symbol is saved into a symbol library the layered text is saved as well.

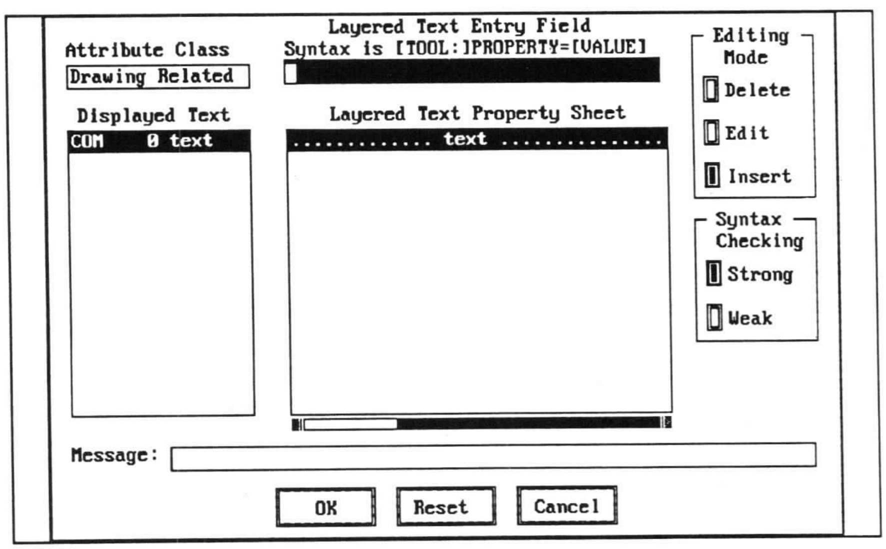
The purpose, use of, and syntax for layered text are explained in the chapter "Understanding FutureNet" in the *FutureNet User Manual.*

Exiting the layered text dialog box saves the edits made to the property sheet. The drawing must be saved before ending the session or loading another drawing in order for the new property sheet to be saved to the drawing file.

Pressing Esc will cancel the layered text dialog box, ignoring changes made during the editing session.

The Layered Text Dialog Box

When the 'L command is entered, the layered text dialog box appears.



What appears in the dialog box depends upon the location of the cursor at the time the 'L command is entered:

- If the cursor is on a text field, the layered text for the class of objects defined by the text attribute class is displayed.

- If the cursor is not on a text field, but the cursor is within a symbol boundary, then the object is assumed to be SYMBOL and all symbol-related layered text for the current symbol is displayed.
- If the cursor is not on a text field and is not in a symbol, then the object is assumed to be DRAWING and all drawing-related text is displayed.

The operation of the dialog box is discussed below.

Attribute Class

Attributes are divided into five classes. (See the chapter "Understanding FutureNet" in the *FutureNet User Manual* for complete explanations of the classes). Depending upon the location of the cursor, the layered text entries displayed belong to one of these classes:

Pin — Lists layered text for all text with pin-related attributes that share the point of effect of the selected text and are unambiguously associated with that pin.

Signal — Similar to PIN except that the text has been assigned signal-related attributes.

Symbol — Lists layered text for all text with symbol-related attributes whose point of effect falls within the symbol boundary, according to OVERLAP rules (see OVERLAP command). Note that text with attribute COM(0) that has its point of effect within the symbol boundary is considered to be symbol related.

Drawing — Lists layered text for all text with drawing-related attributes. Note that text with attribute COM(0) that has its point of effect outside of symbol boundaries is considered to be drawing related.

Circuit — Lists layered text for all text in the drawing that has a circuit (design) related attribute. Note that the property sheet does not list text from other sheets in the drawing set.

Displayed Text

Lists all displayed text that has attributes in the given attribute class. If the cursor was in a text field when 'L was entered, that text is highlighted in the data window. Otherwise, the first displayed text in the list is highlighted. The syntax for the display is

```
attribute_name attribute_number text_string
```

You may select (highlight) any entry in this list, and the associated text in the Layered Text Property Sheet is selected.

Layered Text Property Sheet

Lists all layered text associated with the listed displayed text. Entries in this list are either headings or layered text. Headings (text in a row of dots) cannot be edited. Unless changed, selected text is the first layered text associated with the current Displayed Text. When a line in this list is selected, the associated text in the Displayed Text list is also selected.

Layered Text Entry Field

The Layered Text Entry Field is used to enter new layered text or edit existing layered text. In insert mode, text is entered below the selected entry in the property sheet. In edit mode, the text that is selected in the property sheet is copied to the entry field for editing.

Editing Mode

Editing modes that are available are Delete (delete text), Edit (edit text) and Insert (add new text). Default setting is Insert.

To edit text, select the Edit mode button, then select the text to edit. The text appears in the entry field. Edit the text in the entry field, and press [↵] when finished. The text is updated on the list.

To insert new text, select the Insert mode button. Select the layered text item after which the new text should be inserted. Enter the text in the entry field, and press [↵] when finished.

To delete text, select the Delete mode button, then select the layered text to be deleted. The selected text is removed from the list.

Syntax Checking

Two-state mode button that allows you to adjust the sensitivity of syntax checking. **Strong** does not allow incorrect syntax to be inserted into the Layered Text Property Sheet; **Weak** allows incorrect syntax, but warns you of the problem.

The Message Line and Syntax Errors

The **Message** line gives you information on how to correct the layered text entry if there is a syntax error. Some common syntax errors are shown below:

- Layered text syntax requires a colon after tool specification if the tool is specified. A colon is not allowed anywhere else, with the exception of the value field.

```
Misplaced colon.  Entry ignored.  (strong)
Possible misplaced colon.  (weak)
```

- Syntax requires a property string.

```
Missing property.  Entry ignored.  (strong)
Possible missing property.  (weak)
```

- Syntax requires an equal sign following the property string.

```
Missing equal sign.  Entry ignored.  (strong)
Possible missing equal sign.  (weak)
```

OK Button

Accepts all changes and returns to the drawing.

Reset Button

Abandons all changes made during the current layered text editing session.

Cancel Button

The Cancel button cancels the layered text modifications and returns to the drawing.

To cancel the dialog box without saving the changes, use [Esc] or click the left mouse button outside of the dialog box.

Related Commands

'LE
'LR

'LE — Erase Layered Text

Format

'LE ["]*string*["]

Purpose

Erase the specified layered text entry in the current alphanumeric field or symbol.

Remarks

Not supported by FutureNet OEM products.

The purpose, use of, and syntax for layered text are explained in the chapter "Understanding FutureNet" in the *FutureNet User Manual.*

The graphics cursor must be on the alphanumeric field or symbol containing the target entry. The **'LE** string needs only the information necessary to uniquely identify the entry. If the layered text entry to be erased is tool specific, begin the string with the tool, then add property and value information as necessary to uniquely identify the entry. If the layered text entry is not tool specific, start with the property information and add the value if needed to uniquely identify the entry.

'LE searches for the specified string and erases it. If no string is specified or the string is not found, no text will be erased. To include leading or trailing spaces in the string, use the double quotations. Otherwise FutureNet will use the string from the first non-blank character to the last non-blank character.

If the cursor is on a symbol and is either on a symbol-related alphanumeric field or on no alphanumeric field, then all symbol-related text in the symbol will be searched. Otherwise, if the graphics cursor is on an alphanumeric field, then only layered text associated with that field will be searched.

To use a double quote (") or backward slash (\) in the string, precede the character with a backward slash. For example, to erase the string

"status\pin1"

use the following string:

'LE "\"status\\pin1\""

This command will not affect the displayed text field information.

Note: Be careful, especially when entering this command within a symbol, since only partial strings are necessary to allow a match.

Related Commands

'L
'LR

'LR — Replace/Insert Layered Text

Format

'LR ["]*string*["]

Purpose

Replace an existing value within a string in a layered text entry or insert a new layered text entry consisting of the string in the current alphanumeric field or symbol.

Remarks

Not supported by FutureNet OEM products.

The purpose, use of, and syntax for layered text is explained in the chapter "Understanding FutureNet" in the *FutureNet User Manual.*

String must be in the form of one of the following syntaxes:

tool:property=*value*
tool:property#*value*
property=*value*
property#*value*

In order for the *value* to be replaced, an exact match must be made up to and including the = or # characters:

1. If the cursor is on a symbol and is either on a symbol-related alphanumeric field or on no alphanumeric field, then all symbol-related text in the symbol is searched.
2. If the graphics cursor is on an alphanumeric field, then only layered text associated with that field will be searched.

If *string* (not including the value) is found, the current value is replaced with the value specified in the string. For example, if the field you are searching has the string **tool:property=*xx*** and you enter the command **'LR tool:property=*yy***, the value *xx* will be replaced with *yy*.

If *string* is not found, then the entire string is appended to the list of layered text entries, according to these guidelines:

1. If the cursor is in a symbol but not on an alphanumeric field, the string is appended to the last symbol-related text field.
2. If the cursor is on an alphanumeric field, *string* is appended as the last entry for the alphanumeric field.

To include leading or trailing spaces, use double quotation marks. Otherwise, FutureNet uses the first non-blank character to the last non-blank character. If a double quote (") or backward slash (\) is to be included in the string, precede the character with a backward slash. For example, to replace the string **"status\pin1"** use the string **'LR "\"status\\pin1\""**

Related Commands

'L
'LE

.L — Load Symbol from Library (*)

Format

.L *symbolname* [,[*d:*] *filename*]
* *symbolname* [,[*d:*] *filename*]

Purpose

Load the specified symbol at the graphics cursor location.

Remarks

.L loads the symbol at the graphics cursor location. * loads a symbol and initiates tag and drag.

When *filename* is omitted, the libraries are searched for *symbolname* in the following order:

1. Update library specified in the last **.LIB** command.
2. Reference libraries specified by a **LIB** command.
3. System symbol library, system.sym.

Note: The system.sym library is loaded when FutureNet is initialized. This library is either in the directory specified by the FNLIB environment variable or in the current directory. This library contains a large subset of FutureNet symbols which are supplied with the FutureNet products.

When *filename* is specified as a symbol library, only that library is searched.

The symbol is placed in the drawing so that the upper left corner of the symbol cell is located at the graphics cursor.

The inserted symbol cannot overlap other symbol cells, alphanumeric fields, or lines, or the edge of the drawing. The area must be clear before the symbol can be inserted into the drawing. If there is any conflict, the symbol is not inserted into the drawing and the system displays

```
Boundary conflict
```

Either the graphics cursor must be moved, or the interfering drawing elements must be moved or erased.

Symbols inserted into the drawing are automatically assigned a reference number. This number is displayed in the upper left corner of the symbol cell. The reference number is used to identify the symbol to the system and may be used to find the symbol on the display screen using the N command (Move to Symbol Using Reference Number).

Related Commands

.DIR
.LIB
LIB

.L — Print the Symbol Definition List

(Symbol Definition Mode)

Format **.L**

Purpose Print the instructions for current symbol.

Remarks *Not supported by FutureNet OEM products.*

This command is entered while in Symbol Definition Mode to print the symbol instruction list. It is not entered as part of the instruction list.

.L uses the print relevant options as set up in the **PRINTOPT** command.

.LIB — Update Symbol Library

Format

.LIB [*d:*]*filename*
.NOLIB

Purpose

Specify the update symbol library file into which new symbol definitions are to be saved.

Remarks

The filename extension .sym is automatically added unless an extension is explicitly specified.

If the file already exists, it is assumed to be a library to which additional symbols will be added; otherwise, a new file is created.

Entering the **.LIB** command supersedes use of a library previously specified by **.LIB**, if any.

Once an Update Symbol Library has been specified, use the **.SAVE** command to place symbol definitions in the library.

Entering **.NOLIB** terminates use of the update symbol library.

Note: (For PC Users) If a symbol library is specified on a diskette using the .LIB command, the diskette must not be removed unless access to the library is properly terminated with the .NOLIB command. If this is not done, further access to the library may read random data from a subsequent diskette that could damage the drawing in the FutureNet memory workspace.

Related Commands

LIB
.DIR
DIR
.SAVE
.DEL
.L

/L — Draw Lines

Format /L

Key Left mouse button in MENU and FAST modes.

Purpose Begin and end line drawing in MENU or FAST mode.

Remarks

When line drawing is initiated, the graphics cursor is replaced by the line drawing cursor, a large cross hair. The line drawing cursor is used to route line segments. Up to two routing segments can be defined at one time, the first of which must be set before another can be started. Routing segments are dynamic and, until set, can be rerouted or abandoned altogether. Setting a line segment requires re-entering the /L command or pressing the left mouse button.

Routing segments cannot start or end inside a symbol or alphanumeric field, nor pass through either drawing element. If the graphics cursor is located in an invalid line drawing location, inside a symbol or alphanumeric field, the /L command is ignored.

If there are any boundary conflicts between routing segments and other drawing elements, the routing segments cannot be set.

The MODE status field shows LINE.

Lines can be drawn at any zoom level.

The type of line that will be drawn is shown in the LINE status field.

/L can be used to connect or extend existing lines, provided the line types are the same.

/L is equivalent to the left mouse button in line drawing mode.

If line drawing is initiated but no line is drawn and the cursor is not moved, entering /L again will cancel line drawing mode.

Related Commands

/V
/LE
/E

/LE — Draw/Erase Line

Format

/LE

Purpose

Begin or end line drawing or erase those portions of existing line segments (up to two segments) that coincide with new routing segments.

Remarks

The Draw/Erase Line command can be used to draw new lines or erase existing line segments. /LE operates like the /E command if the routing segment coincides with an existing line, but draws a new line like /L if the routing segment does not coincide with an existing line.

/LE is useful when modifying existing drawings because it allows lines to be drawn and erased with the same command.

Related Commands

/E
/EL
/L

[LOAD — Load Area

Format

[LOAD ***filename*****[.ara]**

Purpose

Insert a previously saved area into the drawing at the graphics cursor.

Remarks

The area must first have been saved using the **[SAVE** command.

Filename specifies the area file to be loaded into the drawing. The system adds the extension **.ara** unless an extension is explicitly specified.

The area is inserted into the drawing so that the upper left corner is positioned at the graphics cursor.

The area loaded is automatically defined (marked by an area boundary), canceling any previously defined areas.

The **[LOAD** command completes only if the graphics cursor is located in a portion of the drawing that is at least the same size as the area to be loaded, that is, clear of symbol cells, alphanumeric fields, and lines.

If the saved area cannot be loaded due to a boundary conflict, the command is ignored and the system displays

```
Boundary conflict
```

The conflict must be resolved before the command can be entered.

Area definition can be canceled using **[K.** Loading another area or creating a new one will also cancel the current area definition, as will saving an area as a file.

Load Drawings as Areas

A drawing file with a .dwg extension can also be specified for the **[LOAD** command. However, the whole drawing in the file must fit into the area in the current drawing for this command to work.

If a drawing is loaded, instead of an area file, then the system-assigned reference numbers will be from bottom up, rather than top down.

If a drawing-sized area is loaded, the area definition boundary may be difficult to see because of its proximity to the drawing edge; nevertheless, areas are loaded with their area definition boundary displayed.

Related Commands

[D
[SAVE

LIB — Specify Symbol Library for Reference NOLIB

Format

LIB [*filename*[,*filename*][...]]
NOLIB

Purpose

Specify symbol libraries to be accessed in addition to **system.sym**.

Remarks

LIB allows symbols from libraries other than **system.sym** to be loaded with the * or **.L** command. **NOLIB** cancels the library specification.

Entering the **LIB** command without specifying a library causes the system to prompt for the name of the library.

Note: FutureNet allows you to specify up to ten reference libraries. However, if you are going to specify more than one symbol library simultaneously, an adequate number of files must be specified in CONFIG.SYS first with the FILES=n command. See the DOS manual for instructions on how to change the number of accessible files.

Related Commands

.DIR
.LIB
.L

LOAD — Load Drawing

Format

LOAD ***filename***

Purpose

Load the specified drawing file into the system workspace.

Remarks

Filename specifies the name of the drawing file to be loaded.

FutureNet will automatically look for a file with the **.dwg** file extension unless a different file extension is given.

When a drawing is loaded, an automatic **CLEAR** command is performed, and the drawing is displayed in fit zoom. Various line, symbol, and area tags are reset and the graphics cursor is positioned at the upper left corner of the drawing.

The size of the drawing is reset to the size stored with the drawing. If a grid was displayed when the drawing was stored, the same grid appears when the drawing is reloaded. Note that the grid does not display in fit zoom.

If the workspace contains a drawing that has been modified and not saved when you load a new drawing, the system displays

```
OK to discard changes that have not already been saved
(Y/N)?
```

Y clears the workspace and loads the drawing. N cancels the **LOAD** command.

When the workspace contains a drawing that has not been changed since it was loaded, verification is bypassed.

In a structured design, the **LOAD** command ends the previous session with a **CLEAR** command before loading the requested root drawing. It displays the same message as the **CLEAR** command. This means that drawings modified during a structured design session can be lost if they are not saved before the **LOAD** command is entered.

If a structured drawing is already loaded, you can use the **#U, #D, #L** and **#R** commands to traverse the drawing structure represented by that root.

If no file was previously loaded, or if **LOAD** is immediately preceded by **SAVEALL** or **CLEAR**, no message is displayed, and the file is loaded as requested.

Related Commands

SAVE
DEL
CLEAR

ls — List Directory

See DIR.

'M — Move Alphanumeric Field

Format

'M

Purpose

Move an alphanumeric field.

Remarks

Place the cursor within the target field's boundary, and enter **'M** or press the left mouse button until the field is tagged. When entered, the MODE status field changes from the current mode to MOVE and the graphics cursor becomes the tag cursor.

When the tag cursor is moved, a dotted boundary detaches from the field and can be moved around the drawing using the mouse or cursor movement commands. The tagged field retains a highlighted boundary. The ghosted boundary aids in locating the text field so that it can be positioned without causing a boundary conflict. You can also use **'B** to display boundaries of all alphanumeric fields. On monochrome systems, BLINK causes the tagged field to blink so that it is easier to see. On color systems, the tagged field changes color.

When **'M** is entered again, the field is moved into the new cursor location, assuming there are no boundary conflicts with other symbol cells, alphanumeric fields, lines, or the edge of the drawing. To resolve boundary conflicts, move the copy to a valid location, or move or erase conflicting objects.

After the field is moved, tag and drag is canceled, control returns to the previous mode, and the graphics cursor is restored.

Related Commands

'B
'C
'K
BLINK

.M — Move Symbol

Format **.M**

Purpose Move the tagged symbol to the current cursor location.

Remarks

When the symbol to be moved is not tagged, first move the cursor into the symbol and enter the **.M** command or press the left mouse button as required to tag the symbol and enable MOVE/COPY/ERASE mode.

When the symbol to be moved is tagged and the cursor is moved, the boundary detaches from the symbol and moves around the drawing with the mouse or cursor movement commands. The tagged symbol boundary and cursor blink on monochrome systems or change color on color systems so that the symbol and its new location are obvious. Entering the **.M** command again or pressing the left mouse button moves the symbol to the current cursor location assuming there are no boundary conflicts.

A symbol cannot be moved to a location where its cell overlaps another symbol, the edge of the drawing, an alphanumeric field, or a line. If the move results in interference, the command is ignored and the system displays

```
Boundary conflict
```

The boundary conflict must be resolved before the move can be successfully completed.

The symbol reference number moves with the symbol.

Whenever a symbol with connections is moved, the connections are maintained only if connection maintenance is enabled with the **CONNECT** command. If a two-segment line can be drawn connecting the points without conflicts, then it is drawn as a permanent line. Where there are conflicts, the connections are drawn as temporary lines.

The area previously occupied by the symbol is left blank. When the move is complete, MOVE/COPY/ERASE mode is canceled, and control returns to the previous mode.

Related Commands

.C
.K
.E
.L
CONNECT

[M — Move Area

Format

[M

Purpose

Move the tagged area.

Remarks

The area must already have been defined; see **[D**.

When **[M** is entered, the area is tagged and a boundary detaches from the area. The boundary can be moved using the mouse or cursor movement commands to the desired location. Entering **[M** again or pressing the left mouse button sets the area in its new location, assuming there are no boundary conflicts.

Boundary conflicts are caused by other symbol cells, alphanumeric fields, lines, or the edge of the drawing.

If the tagged area cannot be moved due to a boundary conflict, the command is ignored and the system displays

```
Boundary conflict
```

Resolve the conflict by moving or erasing the conflicting drawing elements or repositioning the area.

Connection Maintenance

When an area is moved with the **CONNECT** command enabled, existing signal lines that connect to elements outside the area boundary are maintained. Signal lines are maintained from the point at which they cross the area boundary to their point of connection outside the area. When the area is moved and set in place, connections that can be drawn without boundary conflicts are made permanent; connections that conflict with other drawing elements are drawn as temporary lines.

When an area is moved with the **CONNECT** command disabled, existing signal lines that connect to elements outside the area boundary are broken at the area boundary when the area is moved.

Symbols and alphanumeric fields are treated as part of the tagged area only if they are completely within the area boundary. Symbols and alphanumeric fields intersecting the area boundary are excluded from the move.

Related Commands

[C
[D
[K
CONNECT

MENU

Format

MENU

Purpose

The MENU command brings up the menus.

Remarks

The **MENU** command brings up the menus in MENU mode, FAST mode and Symbol Definition Mode. If you are in FAST mode, the **MENU** command brings up the menus without exiting FAST mode.

Related Commands

FAST

MOVEPOEE — Enable/Disable Point of Effect Movement for Electrical Items

Format

MOVEPOEE [ON | OFF]

Purpose

Move alphanumeric fields and their points of effect independently of each other or keep them together.

Remarks

This command applies to alphanumeric fields that have electrical significance, for example, pin and signal name fields.

MOVEPOEE applies when tagging and moving

- an alphanumeric field
- a symbol that an alphanumeric field's point of effect is located in
- an alphanumeric field or symbol as an area.

If **MOVEPOEE** is ON, the point of effect returns to its default location within the alphanumeric field.

If **MOVEPOEE** is OFF, the point of effect does not change, as long as the field is within 127 drawing units of the point of effect location. If the alphanumeric field is more than 127 drawing units from the point of effect, the point of effect will return to its default location.

The default setting is ON.

Entering **MOVEPOEE** without specifying a parameter toggles the setting.

When moving a symbol, electrically significant fields located within the symbol whose points of effect are also located within the symbol move with the symbol regardless of the **MOVEPOEE** setting. If a field's point of effect is outside the symbol, or the alphanumeric field is outside the symbol, the **MOVEPOEE** command works as described above. If on, and either the field or symbol is moved, the point of effect returns to its default location within the alphanumeric field. If off, the point of effect remains at its assigned location.

For non-electrically significant items, see the **MOVEPOEG** command.

Related Commands

MOVEPOEG
'P
'PD
POEDISP
POER

MOVEPOEG — Enable/Disable Point of Effect Movement for General Items

Format

MOVEPOEG [ON | OFF]

Purpose

Move non-electrically significant alphanumeric fields and their points of effect independently or keep them together.

Remarks

This command applies to alphanumeric fields that have nonelectrical significance, for example, comment and filename fields.

MOVEPOEG applies when tagging and moving

- an alphanumeric field
- a symbol that an alphanumeric field's point of effect is located in
- an alphanumeric field or symbol as an area.

When **MOVEPOEG** is ON, the point of effect returns to its default location within the alphanumeric field.

When **MOVEPOEG** is OFF, the point of effect does not change, as long as the field is within 127 drawing units of the point of effect location. If the alphanumeric field is more than 127 drawing units from the point of effect, the point of effect will return to its default location.

The default setting is ON.

Entering **MOVEPOEG** without specifying a parameter toggles the setting.

When moving a symbol, non-electrically significant fields located within the symbol whose points of effect are also located within the symbol move with the symbol regardless of the **MOVEPOEG** setting. If a field's point of effect is outside the symbol, or the alphanumeric field is outside the symbol, the **MOVEPOEG** command works as described above. If on, and either the field or symbol is moved, the point of effect returns to its default location within the alphanumeric field. If off, the point of effect remains at its assigned location.

For electrically significant items, see the **MOVEPOEE** command.

Related Commands

MOVEPOEE
'P
'PD
POEDISP
POER

MX, MY— Move Coordinate in Display Units

(Symbol Definition Mode)

Format

MX ***distance***
MY ***distance***

Purpose

Change the cursor location by specifying the distance to move the *x* or *y* coordinates in display units.

Remarks

Not supported by FutureNet OEM products.

Distance is in display units.

Subsequent instructions originate at the new cursor location.

Distance may be assigned any value from +127 through -128. If distance is between +15 and -16, use the **MXS** and **MYS** (Move Short instructions) to save memory.

MX *distance* changes the *x* coordinate cursor value. If distance is negative, the *x* coordinate moves left. If distance is positive, the *x* coordinate moves right. The *y* coordinate is not changed by the **MX** instruction.

MY *distance* changes the *y* coordinate cursor value. If distance is negative, the *y* coordinate moves up. If distance is positive, the *y* coordinate moves down. The *x* coordinate is unchanged by the **MY** instruction.

The symbol editor coordinate marker moves in the direction and the distance specified by the instruction.

Related Commands

MXS
MYS
MXD
MYD

MXD, MYD— Move Coordinate in Dot Units

(Symbol Definition Mode)

Format

MXD *distance*
MYD *distance*

Purpose

Change the cursor location by specifying the distance to move the *x* or *y* coordinates in dot units.

Remarks

Not supported by FutureNet OEM products.

Distance is in dot units.

Subsequent instructions originate at the new cursor location.

Distance may be assigned any value from +127 through -128. If distance is between +15 and -16, use the **MXS** and **MYS** (Move Short instructions) to save memory.

MX *distance* changes the *x* coordinate cursor value. If distance is negative, the *x* coordinate moves left. If *distance* is positive, the *x* coordinate moves right. The *y* coordinate is not changed by the **MX** instruction.

MY *distance* changes the *y* coordinate cursor value. If *distance* is negative, the *y* coordinate moves up. If distance is positive, the *y* coordinate moves down. The *x* coordinate is unchanged by the **MY** instruction.

The symbol editor coordinate marker moves in the direction and the distance specified by the instruction.

Related Commands

MX
MY
MXS
MYS
MXY

MXS, MYS — Move Short in Display Units

(Symbol Definition Mode)

Format

MXS ***distance***
MYS ***distance***

Purpose

Change the cursor location by specifying the distance to move the *x* or *y* coordinate in display units (use for short moves, between +15 and -16 display units.

Remarks

Not supported by FutureNet OEM products.

Distance is in display units.

If *distance* is between +15 and -16, use this instruction to save memory. For larger *distance* values, use the **MX** and **MY** instructions.

MXS *distance* changes the *x* coordinate cursor value. If *distance* is negative, the *x* coordinate moves left. If *distance* is positive, the *x* coordinate moves right. The *y* coordinate is not changed by the **MXS** instruction.

MYS *distance* changes the *y* coordinate cursor value. If *distance* is negative, the *y* coordinate moves up. If *distance* is positive, the *y* coordinate moves down. The *x* coordinate is unchanged by the **MYS** instruction.

The symbol editor coordinate marker moves in the direction and the distance specified by the instruction.

Related Commands

MX
MY
MXD
MYD
MXY

MXY — Move to New Coordinates in Display Units

(Symbol Definition Mode)

Format

MXY *x,y*

Purpose

Move the cursor in display units the specified distance from the *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

Both *x* and *y* must be specified.

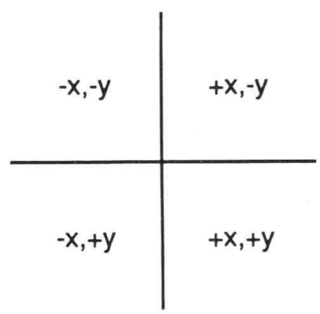
X and *y* are specified in display units. They may be assigned any value from +127 through -128.

MXY moves the *x* and *y* coordinates relative to the current *x,y* coordinates.

Specifying *x* in negative numbers moves the *x* coordinate left; specifying *x* in positive numbers moves the *x* coordinate right.

Specifying *y* in negative numbers moves the *y* coordinate up; specifying *y* in positive numbers moves the *y* coordinate down.

The figure accompanying this instruction shows which direction the possible combinations of positive and negative numbers will move the *x,y* coordinates.



The symbol editor coordinate marker moves in the direction and the distance specified by the instruction.

Related Commands

MXYD
MXYA

MXYA — Move to Absolute *x,y* Coordinates

(Symbol Definition Mode)

Format

MXYA *x,y*

Purpose

Move the cursor to the specified *x,y* coordinates.

Remarks

Not supported by FutureNet OEM products.

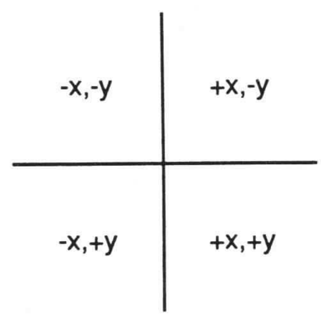
X and *y* are specified in display units. They may be assigned any value between +127 and -128.

MYXA moves the cursor to the specified *x,y* coordinates. It does not move relative to the current location, but the specified distance from the upper left corner of the editing area: x,y equal to 0,0.

Specifying *x* in negative numbers moves the *x* coordinate left; specifying *x* in positive numbers moves the *x* coordinate right.

Specifying *y* in negative numbers moves the *y* coordinate up; specifying *y* in positive numbers moves the *y* coordinate down.

The figure accompanying this instruction shows which direction the possible combinations of positive and negative numbers will move the *x,y* coordinates.



The symbol editor coordinate marker moves in the direction and the distance specified by the instruction.

Related Commands

MXY
MXYD

MXYD — Move to New Coordinates in Dot Units

(Symbol Definition Mode)

Format

MXYD *x,y*

Purpose

Move the cursor in dot units to the specified *x,y* coordinates, starting from the current cursor location.

Remarks

Not supported by FutureNet OEM products.

Both *x* and *y* must be specified.

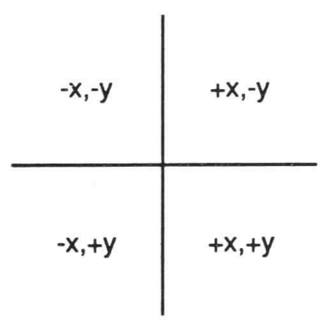
X and *y* are specified in dot units. They may be assigned any value from +127 through -128.

MXYD moves the *x* and *y* coordinates relative to the current *x,y* coordinates.

Specifying *x* in negative numbers moves the *x* coordinate left; specifying *x* in positive numbers moves the *x* coordinate right.

Specifying *y* in negative numbers moves the *y* coordinate up; specifying *y* in positive numbers moves the *y* coordinate down.

The figure accompanying this instruction shows which direction the possible combinations of positive and negative numbers will move the *x,y* coordinates.



The symbol editor coordinate marker moves to the *x,y* location specified by the instruction.

Related Commands

MXY
MXYA

'NAME — Assign a Name to an Unused Attribute

Format

'NAME ***number name***

Purpose

Assign names to unassigned attributes.

Remarks

Both *name* and *number* are required parameters.

Note that this command is for users who are writing custom translators. You cannot take an unassigned attribute and define it to be a signal or pin attribute and have FutureNet and the post processors treat it as such. This command allows entry of information that FutureNet and the post processors do not use, but will pass on through to a custom post processor via the pinlist.

For example, an unassigned attribute, such as a cost field, can be defined by the user. Any text entered in the field would specify the cost for the item the field was associated with. A custom post processor would read these fields in the pinlist and total the cost for the item represented by the schematic.

Number is the number of the attribute you wish to assign a name to. *number* must be an unused attribute. Attribute numbers range from 0 to 255. Unassigned attribute numbers are:

- Unused signal-related attributes:
  15 to 19
- Unused pin-related attributes:
  30 to 49
- Unused unassigned attributes:
  55 to 99
  108 to 113
  116 to 127
  131, 133, 135, 140, 142, 143
  170 to 255

Name is the name (or mnemonic) you wish to assign the specified attribute number. A name can be 1 to 4 characters long and cannot match any of the existing attribute names. See the chapter "Understanding FutureNet" in the *FutureNet User Manual* for more information on attributes.

Unless this command is set up in the **fnpro.cmd** file, the setting is only active during the current editing session.

Related Commands

'A
'S
'CH A

N — Move to Symbol Using Reference Number

Format

N *number*

Purpose

Move the cursor to the specified symbol.

Remarks

Each symbol is automatically assigned a reference number when it is loaded in the drawing. **N** *number* moves the cursor to the upper left corner of the specified symbol.

In fit zoom, the cursor moves to the symbol. In full and intermediate zoom, if the symbol is located off the screen, the display is updated so that the cursor at the symbol is centered on the screen.

If there is no symbol with the specified reference number, the cursor is not moved, and the message line displays

```
Not Found
```

Note that only electrically significant symbols have reference numbers. Graphic Symbols, which can be created using the **.G** symbol definition command, are not found with this command.

Related Commands

RENUM
.G

NOTE — Insert Prompt

Format

NOTE [["]*text* ["]]

Purpose

Display a prompt or comment on the command line while a command file is running (with either AUTO or EXEC).

Remarks

Command file only.

NOTE is used in command files to add user prompts or to display comments, like information about the previous or next command.

Comments are limited to 80 characters, including **NOTE** and any spaces. To include leading or trailing spaces in the text, use the double quotations. Otherwise FutureNet uses the string from the first non-blank character to the last non-blank character.

If a double quote (") or backward slash (\) is to be included in the string, precede the character with a backward slash. For example, to display the string

"status\pin1"

use the following string:

NOTE "\"status\\pin1\""

Related Commands

AUTO
EXEC

'O — Set Default Orientation for Alphanumeric Fields

Format

'O [H | V]

Purpose

Set the default orientation, horizontal (H) or vertical (V), for future alphanumeric fields.

Remarks

The orientation can be either horizontal or vertical.

Horizontal text reads from left to right.

Vertical text reads from bottom to top.

Entering **'O** without specifying an orientation toggles the current setting.

Related Commands

'CH O

'OVER — Add/Delete Overbar to/from Alphanumeric Field (Signal Inversion)

Format

'OVER [ON | OFF]

Purpose

Add an overbar to the current alphanumeric field to indicate signal inversion or delete an existing overbar.

Remarks

Entering **'OVER** without a parameter adds an overbar to the current alphanumeric field if one is not present and deletes one if it is present.

Position the graphics cursor on the target text field and enter the command. Use in base mode on existing alphanumeric fields.

Spaces within the text field will not cause a break in the overbar.

'OVER can only be entered on an existing field.

While in alphanumeric mode, an overbar can be added or deleted by using the [Ctrl] – [O] keystroke sequence.

When an existing field that already has an overbar is edited, the overbar will be added or deleted as text is added or deleted.

Related Commands

'UNDER

OVERLAP — Allow Alphanumeric Fields to Overlap Symbol Boundaries

Format

OVERLAP [ON | OFF]

Purpose

When ON, alphanumeric fields may overlap symbol boundaries.

Remarks

Entering **OVERLAP** toggles the current setting; entering **OVERLAP ON** or **OVERLAP OFF** sets the value directly. OFF is the default.

When overlap is ON, a symbol "owns" any alphanumeric field whose point of effect is inside or on the border of the symbol boundary, even when the alphanumeric field lies partially or entirely outside the symbol boundary. The following are also true:

- Alphanumeric fields associated with the symbol are saved (**.SAVE**) or loaded (**.L**) with the symbol, and are moved (**.M**), copied (**.C**), rotated (**.R**), or reflected (**.RE**) with the symbol.
- The relationship of the alphanumeric field and the symbol remain constant during all operations involving the symbol, regardless of the **MOVEPOEE/MOVEPOEG** state.
- Alphanumeric fields are only permitted to overlap symbol boundaries, **not** lines or other alphanumeric fields.
- When symbols are tagged for any operation, boundaries are checked for any alphanumeric field that is outside the symbol boundary.

The only time that alphanumeric fields are not "owned" is when one or more of the following are true:

- Both the point of effect and the text origin of the alphanumeric field are shared by two or more symbols.
- The point of effect resides in two or more symbols and the text origin resides in another.
- When the point of effect is shared by two symbols, and the alphanumeric field is outside both symbols, the field will be considered to be owned by the tagged symbol.

The above rules do not apply to AREA manipulations, even in **OVERLAP** mode.

'P — Set Point of Effect

Format

'P

Purpose

Move an alphanumeric field's point of effect in order to associate it with an electric or graphic item.

Remarks

To move an alphanumeric field's point of effect, place the graphics cursor on the target field and enter the **'P** command. The MODE status field will display PTR. Move the point of effect cursor using any method of cursor movement and re-enter the **'P** command to set the point of effect at the new location.

Related Commands

MOVEPOEE
MOVEPOEG
'PD
POER

'PD — Reassign a Point of Effect to Its Default Location

Format

'PD

Purpose

Restore the current alphanumeric field's point of effect to its default location.

Remarks

'PD will return the point of effect for the current alphanumeric field to its default location.

The default location for the point of effect is determined by the class of the attribute assigned to the text field, using the following rules:

For Signal-related Attributes

If **POER** is enabled, then the point of effect defaults to the nearest line within 10 display units.

If **POER** is disabled or no line is within 10 display units of the text field, then the point of effect defaults to the justification point of the field.

For Pin-related Attributes

If the alphanumeric field is within a symbol, then the point of effect defaults to the side of the symbol which is closest to the alphanumeric field, and is in the same orientation as the field. For horizontal text, the appropriate sides are the left and right. For vertical text, the appropriate sides are the top and bottom.

If the text field is outside of a symbol, then the default point of effect location is the justification point of the field.

For Other Attributes

The point of effect defaults to the justification point of the field.

The graphics cursor must be on the alphanumeric field that owns the point of effect that is to be returned to its default location.

This command is probably most useful for returning the point of effect to its alphanumeric field when the alphanumeric field has been moved or when a graphic item has been placed between an alphanumeric and its point of effect.

Related Commands

'CH J
'J
MOVEPOEE
MOVEPOEG
'P
POER

'PRINT — Print All Text for Alphanumeric Field

Format

'PRINT [*opt1*][,*opt2*]...[,*optn*]

Purpose

The **'PRINT** command allows you to output all the text for a particular alphanumeric field, including layered text, to an ASCII file.

Remarks

Move the graphics cursor onto the field to be printed and enter the **'PRINT** command. This command uses the print options specified in the **PRINTOPT** command except the Strip option. You can override any options by specifying them in the **'PRINT** command.

See the **PRINTOPT** command for the options available.

Use the **PRINTOPT** command to check your current print defaults.

A minus sign (-) following any text field in the printout indicates an overbar.

Related Commands

.PRINT
PRINT
PRINTOPT

.PRINT — Print All Text for Symbol

Format

.PRINT [*opt1*][,*opt2*]...[,*optn*]

Purpose

The **.PRINT** command allows you to output all the text for a particular symbol, including layered text, to an ASCII file.

Remarks

Move the graphics cursor onto the symbol to be printed and enter the **.PRINT** command. This command uses the print options specified in the **PRINTOPT** command except the Strip option. You can override any options by specifying them in the **.PRINT** command.

See the **PRINTOPT** command for the options available.

Use the **PRINTOPT** command to check your current print defaults.

A minus sign (-) following any text field in the printout indicates an overbar.

Related Commands

'PRINT
PRINT
PRINTOPT

/P — Convert Temporary Lines to Permanent

Format /P

Purpose Convert the temporary line the cursor is on to a permanent line.

Remarks The cursor must be on the temporary line that is to be converted to a permanent line. The /P command will trace and convert up to three temporary segments, assuming no illegal line routing.

Related Commands /ET

PALETTE — Assign FutureNet Colors

Format

PALETTE

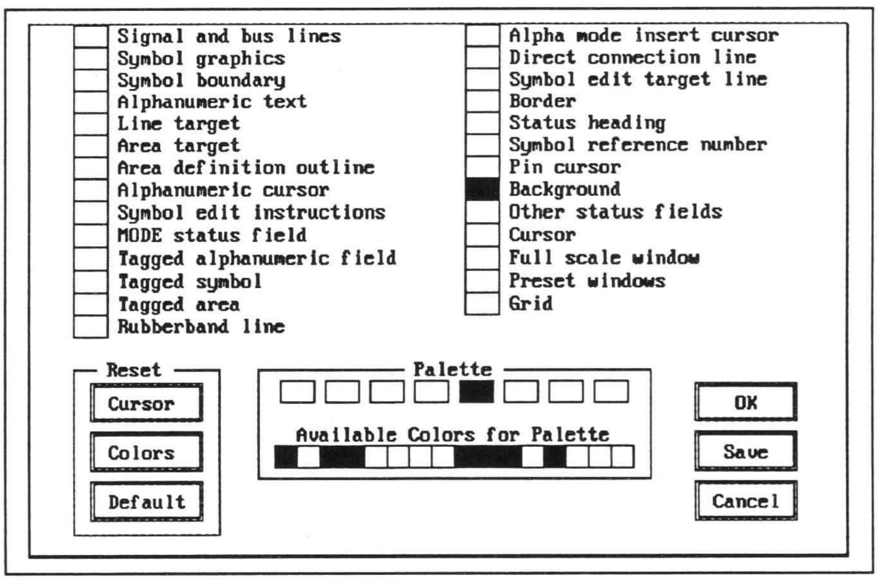
Purpose

Select the colors for the FutureNet display; assign colors to selected components of the FutureNet environment.

Remarks

Note: If you use a monochrome monitor you have no choices to make. FutureNet Group Color 5 and the background must be white. All other choices must be black.

Entering the **PALETTE** command causes a dialog box to appear that lets you choose 8 FutureNet colors from a palette of 16, and then assign any one of those 8 colors to 28 components of the FutureNet display. The dialog box appears below.



The Palette area contains the range of available colors.

The **Palette** color boxes contain the colors that are used for all FutureNet display items. Up to 8 colors may be used. To assign a **Palette** color, click the left mouse button on the color of your choice in the **Available Colors for Palette**. The cursor changes to a hollow box in that color. Place the cursor on the appropriate **Palette** box and press the left mouse button again. All items assigned to that Palette group change color. Alternatively, you can press the left mouse button when the pointing cursor is located inside of a **Palette** box. The box and its associated items cycle through the colors in the Palette. To return the cursor to the pointing cursor, select the **Reset Cursor** button.

Display components may be assigned to a color group in two ways. The first method is to pick up a **Palette** color and assign it to an item. Do this by pressing the left mouse button when the pointing cursor is in the appropriate **Palette** box. The cursor changes to solid block in that color. Place the cursor in the color box of the item to be assigned and press the left mouse button again. The item's color changes to that of the new Group. To return the cursor to the pointing cursor, select the **Reset Cursor** button. The second method is to place the pointing cursor on the color box of the item itself. Press the left mouse button to cycle through the colors available in the **Palette.**

If you attempt to place a color from an **Available Colors** box in an item color box, the cursor changes to a stop sign. Click on Reset cursor to return to the pointing cursor.

The Background box cannot be assigned a different **Palette** color. You can only change the background color by changing the color of the fifth **Palette** box.

Action Button Choices

Reset Cursor — Returns any **Palette** cursor to the pointing cursor.

Reset Colors — Returns **Palette** colors to their previous assignments.

Reset Default — Returns **Palette** colors to the program default colors.

OK — Saves the changes made, exits the **Palette** dialog box, updates graphics if necessary, and returns to the drawing screen. If your changes aren't reflected on the screen, refresh the screen by pressing [PgUp] (or [R9]) then [PgDn] (or [R15]).

Save — Saves changes made to the palette to the current **fnpro.cmd** file. Depending upon your FNPRO variable, this may standardize your color choices regardless of the directory from which you start FutureNet. The changes you have made are appended to the end of the command file using **DISPGRP** and **COLOR** command syntax. You must have write permission to **fnpro.cmd.**

Cancel — Leaves the **Palette** dialog box, cancelling any changes done.

You can also cancel the **Palette** dialog box and discard the changes, by pressing the [Esc] key or clicking the cursor outside the dialog box.

Related Commands

DISPGRP*n*
COLOR

PAN — Pan to Window Locations

Format

PAN 1 | 2 | 3 | 4

Purpose

Move to the specified window.

Remarks

Note: Window locations must first be saved using the WINDOW command.

A number is required.

PAN rapidly displays a preset window location. Preset window locations will appear in full zoom, regardless of the zoom level in effect at the time **PAN** is entered.

If **PAN** is used and no window with the specified number exists, the command is ignored and the message line displays

```
Pan not set
```

Related Commands

WINDOW

PAUSE — Change from Automatic to Single-step Execution

Format

PAUSE

Purpose

Change command file execution from automatic (**AUTO**) to single-step (**EXEC**).

Remarks

To enter, type **PAUSE** on the command line and press ⏎ while **AUTO** execution is taking place.

The **PAUSE** command can also be entered in a command file.

The **PAUSE** command suspends **AUTO** execution. The system goes into single-step **EXEC** mode, displaying each command on the command line.

Resume automatic file execution by entering the **AUTO** command without specifying a filename.

Related Commands

AUTO
EXEC
STOP

PgUp or R9 Keys— Move to Beginning of Command List
PgDn or R15 Keys — Move to End of Command List

(Symbol Definition Mode)

Keys

PC	SUN
PgUp	R9
PgDn	R15

Purpose

Move the beginning or end of the symbol definition instruction list to the target line.

Remarks

Not supported by FutureNet OEM products.

PgUp or R9 moves the first symbol definition instruction line into the target line.

PgDn or R9 makes the target line the one after the last symbol definition instruction line. When the line after the last symbol definition instruction is positioned as the target line, additional symbol definition instructions entered are added to the end of the list.

Note that PgUp , PgDn , R9 and R15

have a different function when not in Symbol Definition Mode. See **ZIN/ZOUT** commands.

PINSNAP — Snap Line to Pin

Format PINSNAP [ON | OFF]

Purpose Enable or disable pinsnap.

Remarks Pinsnap is the automatic connection of lines that come within two display units of a pin-related text point of effect.

If no parameter is specified, the command toggles the current condition.

Enabled is the default setting.

Related Commands PROFILE

POEDISP — Display Points of Effect

Format

POEDISP [ON | OFF]

Purpose

Display points of effect.

Remarks

POEDISP displays the points of effect for all alphanumeric fields in the drawing. A line will connect the point of effect and the field.

Points of effect can be moved while **POEDISP** is ON and the line that associates a point of effect with its alphanumeric field will track the point of effect as you move it.

The default setting is OFF.

Entering **POEDISP** without any parameters toggles the setting.

Related Commands

'P
'PD
MOVEPOEE
MOVEPOEG
POER
PROFILE

POER — Point of Effect Range

Format

POER [ON | OFF]

Purpose

Toggle the snap-to feature for signal-related alphanumeric field points of effect.

Remarks

When ON, the points of effect for signal-related alphanumeric fields will snap to a signal line from up to ten display units away, or snap to the nearest of two signal lines.

When OFF, the signal-related alphanumeric field boundary must be touching the signal line in order to affix the point of effect to that line.

If no parameter is specified, the command toggles the current condition.

This command only applies to alphanumeric fields which have been assigned signal-related attributes.

Refer to the chapter "Understanding FutureNet" in the *FutureNet User Manual* for a list of signal-related attributes.

In areas where signal lines are crowded together, it may be necessary to use the **'P** command to place point of effect in the exact location desired.

Enabled is the default setting.

Related Commands

MOVEPOEE
MOVEPOEE
'P
'PD
PROFILE

PRINT — Print Drawing

Format

PRINT [*opt1*][,*opt2*]...[,*optn*]
NOPRINT

Purpose

Print the current drawing.

Remarks

Print the drawing currently in the work space.

PRINT uses the relevant print options as set up in the **PRINTOPT** command, or options can be specified on the command line. Options specified in this command temporarily override any conflicting options in the **PRINTOPT** command.

Printing in progress can be canceled by pressing any key during direct output to the printed, or by typing **NOPRINT** during spooled printing. In either case, FutureNet will ask for verification

```
Halt printing (Y/N)?
```

See the **PRINTOPT** command for valid options.

Use the **PRINTOPT** command to see current settings.

Related Commands

'PRINT
.PRINT
PRINTOPT

PRINTOPT — Set Print Options (Sun)

Format

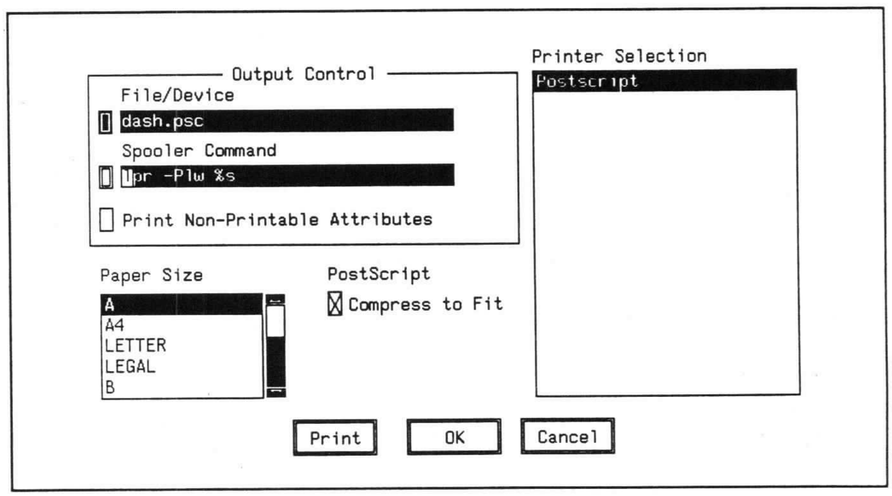
PRINTOPT [*opt1*][,*opt2*]...[,*optn*]

Purpose

Select the printer, width and other print characteristics to be used when you enter a **PRINT** command.

Remarks

You can list options in any order (using commas as delimiters is optional). A conflicting option in a **PRINT** command will override the one specified with **PRINTOPT** for that print job. Entering the PRINTOPT command with no parameters brings up the Printer Setup dialog box shown below.



The available print options are listed below. The dialog box selection is on the left, and the option syntax for the PRINTOPT command line entry is given across from it on the right.

Output Control

File/Device
☐ ▮__

This option specifies the name of the file or device to which the printer output will be directly written. (FILE=*filename/device*)

To choose the direct print option, select the box to the left of the file/device name. (F)

A printer output file that has been created can later be printed by using the spooler (see "Spooler Command" below). The default filename for direct printer output is **fn.psc**.

The device that you are sending output to must be directly connected to the local workstation.

Spooler Command
☐ ▮__

This option specifies the system command string that will invoke the system spooler. The command string must contain the name of the spooler and a filename place holder (**%s**). Each time the system spooler is invoked, FutureNet will replace the **%s** place holder with the appropriate filename. The default command string is **lpr -Plw %s**. (SPOOLER="*spooler_command_string*")

To choose the spooled print option, select the box to the left of the spooled command string. (S)

☐ **Print Non-printable Attributes** — Prints all non-printing attributes, the symbol boundaries, the symbol reference number, and the grid defined with the **GRID** command. The default is do not print non-printing attributes. (R | RNO)

Strip

☐ ALL	Prints all strips. If your drawing is wider than one strip, then multiple strips will be sent to the printer with each strip starting on a new page.
☐ 1 – 5	Selects a specific horizontal strip of a drawing to be printed. The strip width is the paper width specified by the paper width minus 0.5-inches (13 mm) for laser printers. 1 is the default. (ALL | 1 | 2 | 3 | 4 | 5)

Laser Options

These options are used by laser printers only.

Paper Size

LETTER	Denotes printer paper size of 8.5-inches by 11 inches.
LEGAL	Denotes printer paper size of 8.5-inches by 14 inches.
A,B,C,D,E	Denotes English standard drawing sizes. See the **SIZE** command for size of drawings.
A0,A1,A2, A3,A4,A5	Denotes metric standard drawing sizes. See the **SIZE** command for size of drawings.

☐ **Compress to Fit** — Fits entire drawing on one page, if printer has the capability. (FIT | NOFIT)

Printer Selection

Sets the printer type with a name that uniquely identifies a particular printer. The name must be exactly as shown in the list of printers in the **PRINTOPT** dialog box (printers are also listed when you enter **Help printer list** on the FutureNet command line). All supported printers are listed in the **PRINTOPT** dialog box. (PRINTER="*name*")

To pick a printer from the **Printer Selection** list, position the cursor over the printer of choice and click the left mouse button to highlight it.

[Print] — This button prints the current drawing with the selected options.

[OK] — This button accepts the changes made in the dialog box.

[Cancel] — This button exits the dialog box cancelling changes. You can also press [Esc] or click outside the dialog box with the left mouse button.

See the *FutureNet User Manual* for information on how to use dialog boxes.

Related Commands

'PRINT
.PRINT
PRINT
.L (Symbol Definition Mode)

PRINTOPT — Set Print Options (PC)

Format

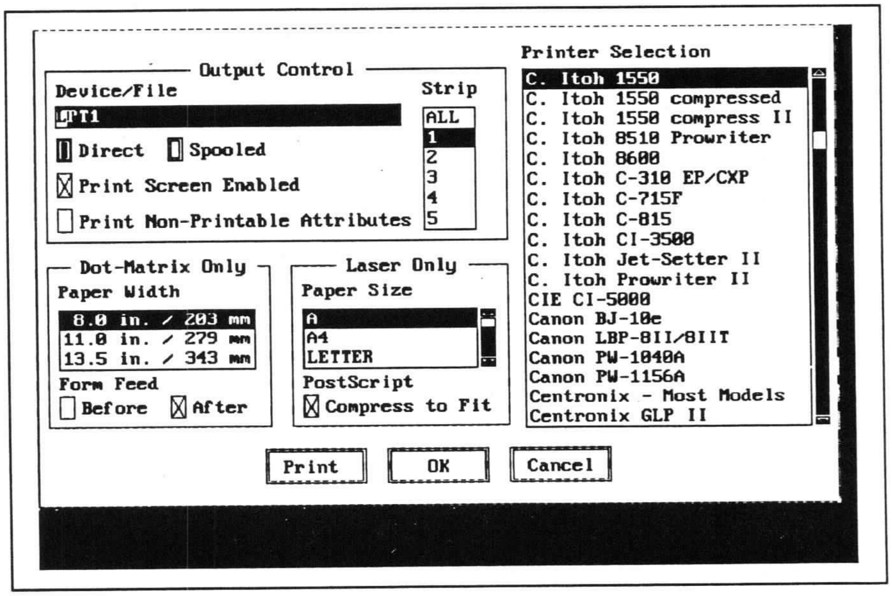
PRINTOPT [*opt1*][,*opt2*]...[,*optn*]

Purpose

Select the printer, width and other print characteristics to be used when you enter a **PRINT** command or Shift – PrtSc .

Remarks

Options can be listed in any order and the use of commas as delimiters is optional. A conflicting option in a **PRINT** command will override the one specified with **PRINTOPT** for that print job. Entering the PRINTOPT command with no parameters brings up the Printer Setup dialog box shown below.



The available print options are listed below. The dialog box selection is on the left, and the option syntax for the PRINTOPT command line entry is given across from it on the right.

Output Control

Device/File

This option specifies the name of the device or file to which the printer output will be written. (**FILE=***device/filename*)

A printer output file that has been created by direct print can later be printed from DOS using the following command:

COPY *filename port* /b

where *port* is the port connected to the printer (LPT1, COM2, etc.) and /b (required) specifies binary mode for DOS.

Do not use the DOS PRINT command to print a direct print output file. The **PRINT** command acts as a filter, removing essential escape sequence information.

☐ Direct/Spooled Options	Direct print writes directly to the device or filename you specify in the FILE= parameter at the end of the command. The default device is LPT1. Spooled printing first writes to a temporary file in the root directory (during this time you cannot edit), then prints the temporary file (releasing the system so that you can continue to edit while the printer is still printing). (F I S)
☐ Print Screen Enabled	This option allows you to enable the normal PC Shift – PrtSc function. (S$YES I S$NO)
☐ Print Non-printable Attributes	Prints all non-printing attributes, the symbol boundaries, the symbol reference number, and the grid defined with the **GRID** command. The default is do not print non-printing attributes. (R I RNO)

Strip

	☐ ALL	Prints all strips. If your drawing is wider than one strip, then multiple strips will be sent to the printer with each strip starting on a new page.
	☐ 1 – 5	Selects a specific horizontal strip of a drawing to be printed. The strip width is the paper width specified by the N, M, or W setting for dot-matrix printers or the paper width minus 0.5-inches (13 mm) for laser printers. 1 is the default. (ALL I 1 I 2 I 3 I 4 I 5)

Dot--matrix Only

These options are used by dot-matrix printers only.

Paper Width

8.0 in./203 mm	Narrow (N). Specifies printing on an 80-character, 8-inch (203 mm) wide printer.
11.0 in./279 mm	Medium (M). Specifies printing on a 110-character, 11-inch (279 mm) wide printer.
13.5 in./343 mm	Wide (W). Specifies printing on a 132-character, 13.5-inch (343 mm) wide printer.
	Narrow is the default value. (N I M I W)

Form Feed

☐ Before	Form feeds to a new page at the beginning of a print operation. (PAGETOP I PAGETOPNO).
☐ After	Form feeds to a new page at the end of a print operation. (PAGEBOT I PAGEBOTNO)

Laser Only Options

These options are used by laser printers only.

Paper Size

LETTER	Denotes printer paper size of 8.5-inches by 11 inches.
LEGAL	Denotes printer paper size of 8.5-inches by 14 inches.
A,B,C,D,E	Denotes English standard drawing sizes. See the **SIZE** command for size of drawings.
A0,A1,A2, A3,A4,A5	Denotes metric standard drawing sizes. See the **SIZE** command for size of drawings.

☐ **Compress to Fit**

Fits entire drawing on one page, if printer has the capability (PostScript printers only). (FIT | NOFIT)

Printer Selection

Sets the printer type with a name that uniquely identifies a particular printer. The name must be exactly as shown in the list of printers in the **PRINTOPT** dialog box (printers are also listed when you enter **Help printer list** on the FutureNet command line). All supported printers are listed in the **PRINTOPT** dialog box. (PRINTER="*name*")

To pick a printer from the **Printer Selection** list, position the cursor over the printer of choice and click the left mouse button to highlight it.

[Print]

This button prints the current drawing with the selected options.

[OK]

This button accepts the changes made in the dialog box.

[Cancel]

This button exits the dialog box cancelling changes. You can also press [Esc] or click outside the dialog box with the left mouse button.

See the *FutureNet User Manual* for information on how to use dialog boxes.

Related Commands

'PRINT
.PRINT
PRINT
.L (Symbol Definition Mode)

PROFILE — Display Profile Options

Format

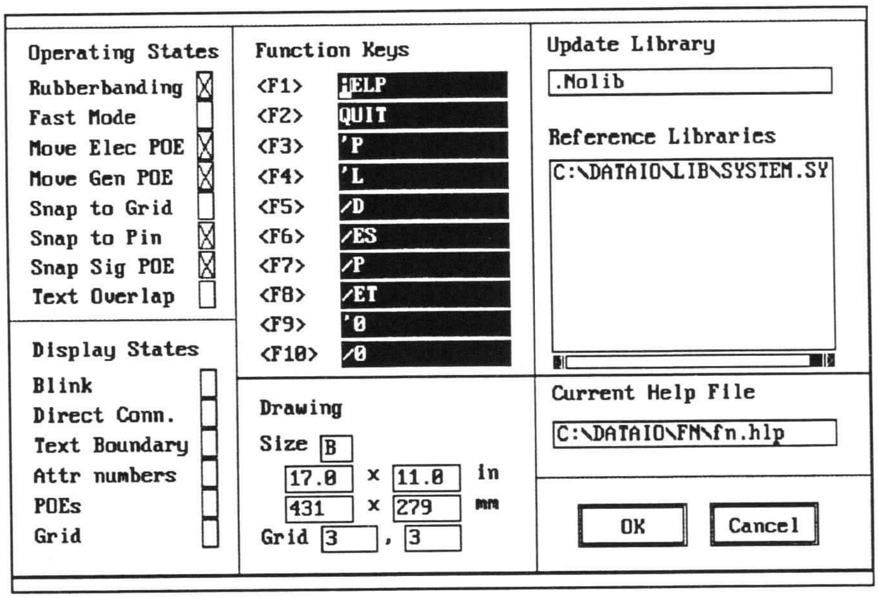
PROFILE

Purpose

Display system settings.

Remarks

The profile screen is a dialog box that, from one location, lets you review or modify a selected set of FutureNet environment parameters that would otherwise have to be reviewed or modified individually.



See the FutureNet User Manual for basic operation in dialog boxes.

Each section of the **PROFILE** screen is described below. Capitalized words in parentheses, such as (CONNECT), identify the command to which the profile selection is mapped.

Operating States

☐ Rubberbanding

Default: ON. Indicates whether or not connections should be maintained by rubberbanding lines when moving areas or symbols. (**CONNECT**)

☐ Fast Mode

Default: OFF. Indicates whether the drawing screen is in **MENU** or **FAST** mode. (**FAST**)

☐ Move Elec POE

Default: ON. Indicates whether points of effect for electrical fields (signals and pins) should return to default locations or be left where they are, when the associated field is moved. ON indicates the point of effect moves with alphanumeric fields. (**MOVEPOEE**)

☐ **Move Gen POE**

Default: ON. Indicates whether points of effect for non-electrical fields (not signals or pins) should return to default locations or be left where they are, when the associated field is moved. ON indicates the point of effect moves with alphanumeric fields. (**MOVEPOEG**)

☐ **Snap to Grid**

Default: OFF. Indicates whether the cursor will snap to the closest grid point in the left and upward direction when a grid is displayed. When ON, the cursor moves only between grid points. (**GRID SNAP**)

☐ **Snap to Pin**

Default: ON. Indicates whether a line that is being drawn will snap to a symbol pin that is within two display units. When ON, lines that are being drawn snap to symbol pins. (**PINSNAP**)

☐ **Snap Sig Poe**

Default: ON. Indicates whether the extended point of effect range is active. When ON, points of effect for alphanumeric fields with signal attributes snap to the closest type /1 (signal) or /2 (bus) line that is within 10 display units. (**POER**)

☐ **Text Overlap**

Default: OFF. Indicates whether alphanumeric fields can overlap symbol boundaries. See OVERLAP for more information. (**OVERLAP**)

Display States

☐ **Blink**

The **BLINK** command controls blinking for the following:

- Zoom window (visible when in fit zoom)
- Alphanumeric fields at the graphics cursor
- Tagged alphanumeric fields, symbols and areas
- The direct-connection cursor (/C command)
- Temporary lines

Blinking is the default value on monochrome systems. Non-blinking is the default value on color systems. (**BLINK**)

☐ **Direct Conn.**

Default: OFF. Indicates whether direct connections will be displayed in place of other graphical elements. (**.DCON**)

☐ **Text Boundary**

Default: OFF. Indicates whether the boundaries of all alphanumeric fields will be displayed. When on, the dotted boundaries and justification points for all alphanumeric fields are displayed. (**'B**)

☐ **Attr Numbers**

Default: OFF. Indicates whether the attribute for each alphanumeric field will be displayed instead of the alphanumeric text. When on, the attributes are displayed in reverse video. (**'D**)

☐ **POEs**

Default: OFF. Indicates whether points of effect will be displayed for all alphanumeric fields. When ON, the points of effect for all alphanumeric fields are displayed. Points of effect are connected to their corresponding alphanumeric field by a line in order to show which point of effect belongs to which text field. (**POEDISP**)

☐ **Grid**

Default: OFF. Selects whether a grid is displayed in the workspace. When ON, the grid is displayed. (**GRID**)

Function Keys ▮__	Displays the function currently assigned to each function key. See the chapter "Using FutureNet" in the *FutureNet User Manual* for information on setting up and using the function keys. These fields can be edited by clicking the left mouse button on the desired field. You can display commands to the right using [Ctrl] – [→]. Complete editing instructions are given in the *FutureNet User Manual*.
Drawing	
Size	Shows the size of the current drawing. You cannot directly edit this field. It must be changed by command. **(SIZE, SIZED, SIZEM)**
Grid	Shows the grid defined in the xy plane. The grid appears as dots on the drawing screen. You cannot directly edit this field. It must be changed with the GRID command. **(GRID)**
Libraries	
Update Library	Displays the name of the Update Library, if one has been opened, to which you can save symbols or from which you can load symbols. You cannot directly edit these fields. They must be changed by command. **(.LIB** and **.NOLIB)**
Reference Libraries	Displays the names of the reference libraries from which you can read symbols. Use the scroll bars to reveal long names, or to view additional reference libraries. You cannot directly edit this field. It must be changed by command. **(LIB, NOLIB)**
Current Help File	Display path and filename of help file. You cannot directly edit this field. It must be changed by command. **(HELPFILE)**
[OK]	This button saves the changes made in the dialog box.
[Cancel]	Quit the **PROFILE** screen without saving any changes. Press [Esc] or click the left mouse button outside of the dialog box to cancel the **PROFILE** dialog box and discard any changes made.

.Q — Exit Symbol Definition Mode

(Symbol Definition Mode)

Format

Q
QUIT
.Q
END

Purpose

Exit Symbol Definition Mode and return control to the drawing screen.

Remarks

Not supported by FutureNet OEM products.

If Symbol Definition Mode was entered with the graphics cursor located in an existing symbol, the edited symbol definition replaces the original definition of that symbol in the drawing.

If a new symbol was created, it is placed in the drawing with the upper left corner of the symbol cell at the location of the graphics cursor.

Related Commands

.S

QUIT — Quit Editing Session

Format

QUIT
Q
END
SYSTEM

Purpose

End the FutureNet editing session and return to the operating system. Or, if you are in Symbol Definition Mode, exit that mode and return to the drawing screen.

Remarks

If the drawing in the work space or a drawing in a structured design was modified and not saved, the system displays

```
OK to discard changes that have not already been saved
(Y/N)?
```

Y erases the contents of the system work space. N cancels the command, allowing schematic editing to continue. In a structured design, you can use the **FILE** command to see the status of all files accessed in this session and decide whether or not to save them. Use the **SAVE** command to save a specific file. Note that the drawing being saved must be in the current workspace to be saved using the **SAVE** command. Use the **SAVEALL** command to save all changed files.

When quitting FutureNet with a drawing that has been viewed but not changed, this verification is bypassed. Commands that affect only how the drawing is viewed do not change the drawing. These commands include cursor moves, Zoom In, Zoom Out, the **Home** command, and setting default attributes, character size, and line type.

If a drawing is being printed using spooling, the **QUIT** command will issue the following warning:

```
Halt printing (Y/N)?
```

If you answer Y for yes, printing stops. If you answer N for no, printing and schematic editing continue.

'R — Replace Alphanumeric String

Format

'R ["]*string*["]

Purpose

Insert the specified alphanumeric string at the cursor location.

Remarks

String is a required parameter. To include leading or trailing spaces in the string, use the double quotations. Otherwise FutureNet will use the string from the first non-blank character to the last non-blank character.

'R either replaces an existing alphanumeric field or creates a new one.

The original command remains on the command line, making it possible to move the cursor to a new location and enter the command again.

If *string* replaces an existing alphanumeric string, it will retain the font type, attribute, orientation, justification, and printability values of the text it is replacing. If it creates a new field, it will assume the default font, attribute, orientation, justification, and printability values which are displayed in the status fields.

If a double quote (") or backward slash (\) is to be included in the string, precede the character with a backward slash. For example, to replace the string

"status\pin1"

use the following string:

'R "\"status\\pin1\""

Related Commands

'F
'FA
'I
Esc

.R — Rotate a Symbol

Format

.R [- | +]
.R [90 | 180 | 270 | -90 | -180 | -270]
.R [0]

Purpose

Rotate a symbol in increments of 90 degrees around the upper left corner of the symbol and tag the symbol so that it can be moved.

Remarks

.R and **.R +** without any parameters both rotate the symbol 90 degrees clockwise; **.R -** without any parameters rotates the symbol 90 degrees counterclockwise. Note that there is a space between the **.R** and the plus (+), minus (-) or number. Positive numbers rotate the symbol clockwise; negative numbers rotate the symbol counterclockwise. **.R 0** will return the symbol to non-rotated position.

Move the cursor to the desired symbol and enter the **.R** command. .R rotates the current symbol and also tags it so that it can be moved using the mouse or arrow keys; MOVE appears in the MODE status field.

If the symbol has already been tagged using **.M**, entering **.R** and any optional parameters will rotate the symbol accordingly. If rotating or placing the symbol causes a boundary conflict, the command is ignored and the system displays

```
Boundary conflict
```

To complete the rotation operation, use the **.M** command or the left mouse button.

Any connections to the symbol are broken at the symbol edge when it is rotated.

Related Commands

.RE

.RE — Reflect a Symbol

Format

.RE [H | V]

Purpose

Reflect a symbol horizontally (H) or vertically (V) about its axis, creating a mirror image of the symbol.

Remarks

Symbols can be reflected 180 degrees horizontally (H) or vertically (V). If no direction is specified, the symbol is reflected horizontally 180 degrees.

Symbols are tagged during reflection and MOVE appears in the MODE status field. Until the symbol is set in place, it can be moved using the mouse or arrow keys. If placing the reflected symbol causes a boundary conflict, the command is ignored and the system displays

```
Boundary conflict
```

To complete the reflection operation, use the .M command or the left mouse button.

Any connections to the symbol are broken at the symbol edge when it is reflected.

Related Commands

.R

/R — Change Line Routing

Format

/R

Purpose

Switch line routing from horizontal/vertical to vertical/horizontal.

Remarks

Normal line drawing draws horizontal routing segments first and vertical routing segments second. This order can be reversed so that vertical segments are drawn first and horizontal segments second by entering the **/R** command or using the middle mouse button to change the order of the routing segments. This command only works in line drawing mode (line drawing cursor is present). It can be entered before the routing segments have been drawn, or after they have been drawn but before they have been set. Once line segments have been set, they cannot be changed using this command.

The middle mouse button is equivalent to **/R** in **MENU** and **FAST** drawing modes.

Related Commands

/E
/L
/LE
/V

[R — Rotate an Area

Format

[R [- | +]
[R [90 | 180 | 270 | -90 | -180 | -270]

Purpose

Rotate an area in increments of 90 degrees in the x,y plane and tag it so it can be moved.

Remarks

The area must already have been defined; see [D.

[R and **[R +** without any parameters both rotate the area 90 degrees clockwise; **[R -** without any parameters rotates the area 90 degrees counterclockwise. Note that there is a space between the **[R** and the plus (+), minus (-), or number. Positive numbers rotate areas clockwise; negative numbers rotate areas counterclockwise.

Areas are tagged during rotation and MOVE appears in the MODE status field. Until the area is set in place by entering the command again, it can be relocated using the mouse or arrow keys, as well as rotated. If placing the rotated area causes a boundary conflict, the command is ignored and the system displays

```
Boundary conflict
```

To complete the rotation operation, use the **[M** command or the left mouse button.

Any connections to the area are broken at the area boundary when it is rotated.

All text fields, whether inside or outside of a symbol, will be oriented so that horizontal text is read from left to right and vertical text is read from bottom to top, regardless of the number of rotations.

Related Commands

[D
[M
[RE

[RE — Reflect an Area

Format

[RE [H | V]

Purpose

Reflect an area about its horizontal (H) or vertical (V) axis, creating mirror image of the area.

Remarks

The area must already have been defined; see **[D**.

Areas can be reflected 180 degrees horizontally (H) or vertically (V). If no direction is specified, the area is reflected horizontally 180 degrees.

Reflecting an area horizontally means that it will pivot on a vertical axis. Reflecting an area vertically means that it will pivot on a horizontal axis.

Areas are tagged during reflection and MOVE appears in the MODE status field. Until the area is set in place, it can be moved using the mouse or arrow keys. If placing the reflected area causes a boundary conflict, the command is ignored and the system displays

```
Boundary conflict
```

To complete the reflection operation, use the **[M** command or the left mouse button.

Any connections to the area are broken at the area edge when it is reflected.

All text fields, whether inside or outside of a symbol, will be oriented so that horizontal text is read from left to right and vertical text is read from bottom to top.

Related Commands

[D
[M
[R

REFRESH — Refresh Screen

Format REFRESH

Key [Ctrl] – [L]

Purpose Repaint the current FutureNet screen.

Remarks Entering **REFRESH** or pressing [Ctrl] – [L] causes a repaint of the screen, clearing any text from the message area. The current command on the command line is maintained.

RENUM — Resequence Symbol Reference Numbers

Format

RENUM

Purpose

Resequence symbol reference numbers in consecutive order following changes to the drawing.

Remarks

Each symbol in a drawing is assigned a reference number as it is loaded into the drawing.

This command numbers the symbols left to right, top to bottom throughout the drawing.

Note: *This command should be used with care because some translators and back-end tools use the symbol reference number to uniquely identify a symbol. Changing symbol reference numbers with RENUM can interfere with the operation of these tools.*

rm — Remove File

Format

rm ***command line arguments***

Purpose

Remove the specified file from the current directory (SUN only).

Remarks

See your system documentation regarding the **rm** command.

Global file characters * and ? can be used with the **rm** command.

Related Commands

CD
DEL
ls

RS — Reset to Default Symbol Element Set

(Symbol Definition Mode)

Format

RS

Purpose

Select the default symbol element table.

Remarks

Not supported by FutureNet OEM products.

RS resets the default symbol element table shown in an appendix in the *FutureNet User Manual*. The symbol element table can be modified by the **ST** command.

The symbol editor coordinate marker is unaffected by this instruction.

Related Commands

ISxx
ST

'S — Set Attribute for Existing Field

Format

'S name | number

Purpose

Set the default attribute to be assigned to future alphanumeric fields or change the attribute setting for the alphanumeric field the cursor is located in.

Remarks

An attribute number or name must be specified.

To change the default attribute setting, move the graphics cursor to an unoccupied area of the work space and enter the **'S** command along with the new attribute setting. The new default setting will appear in the ATTR status field. This mode of **'S** is similar to the **'A** command.

To change the attribute setting for an existing alphanumeric field, move the graphics cursor to that field and enter the **'S** command along with the new attribute setting. This mode of **'S** is similar to the **'CH A** command.

The **'D** command can be used to display all attributes for all alphanumeric fields. For a particular field, the attribute is shown in ATTR status field when the graphics cursor is located on the alphanumeric field.

See the chapter "Understanding FutureNet" in the *FutureNet User Manual* for more information on attributes.

Related Commands

'A
'CH A
'D

.S — Enter Symbol Definition Mode

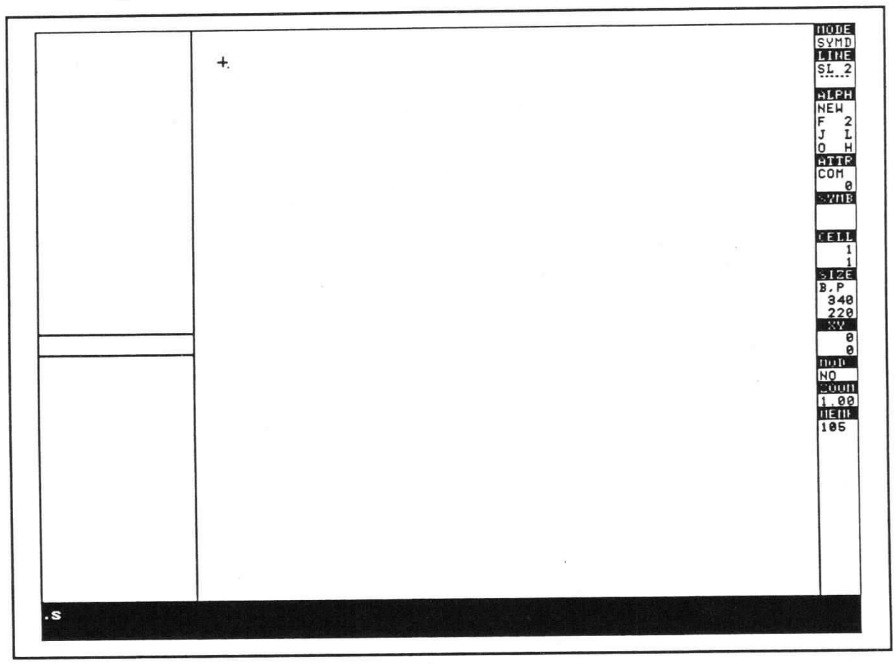
(Symbol Definition Mode)

Format .S

Purpose Enter Symbol Definition Mode.

Remarks *Not supported by FutureNet OEM products.*

The Symbol Definition Mode screen is shown below. The symbol definition list appears in the column on the left, with the definition area on the right.



The symbol editor coordinate marker will be positioned in the upper left corner of the symbol definition area.

When creating a new symbol (the graphics cursor was not located in a symbol cell when Symbol Definition Mode was entered), the symbol definition list will be empty. Before a symbol cell is defined, the symbol cell is one display unit high by one display unit wide.

When editing an existing symbol (the graphics cursor was located in a symbol cell when Symbol Definition Mode was entered), the symbol definition instruction list is copied into the symbol editor and is displayed for viewing or editing. The symbol cell is displayed in the upper left corner of the graphics area.

Related Commands .Q

.SAVE — Save Symbol in Current Update Library

Format

.SAVE ***symbolname***

Purpose

Save the specified symbol definition in the current update library.

Remarks

.SAVE saves the symbol on which the graphics cursor is located. The graphics cursor can be located anywhere within the symbol boundary.

Symbolname specifies the name of the symbol and can include any ASCII characters except commas and spaces. *Symbolname* should be no more than 16 characters in length.

The symbol and any alphanumeric data is stored in the current update library (update libraries are specified using the **.LIB** command). Any existing symbols with the same name will be overwritten.

Related Commands

.LIB
.DEL
.DIR
.DIRPR

.SBS — Set Block Symbol Bit

(Symbol Definition Mode)

Format

.SBS

Purpose

Allow use of pin commands in drawing mode to edit block symbols.

Remarks

Not supported by FutureNet OEM products.

.SBS should only be used if Symbol Definition Mode was inadvertantly entered when the cursor was on a block symbol.

.SBS must be entered prior to exiting Symbol Definition Mode, if pin stub commands are to be used in the drawing mode.

Once a symbol has been edited in Symbol Definition Mode, a flag is set that prevents it from being edited using the symbol definition commands available outside Symbol Definition Mode. Entering the **.SBS** command while in Symbol Definition Mode resets the flag, permitting symbols to be edited with pin stub commands.

Note: Symbol definition instructions are normally performed relative to the end point of the previous instruction, creating an instruction list in which the instruction graphics are drawn in the order of the instruction list. However, the instructions for the pin stub commands are inserted at the beginning of the existing instruction list. So, if the ***MXYA*** *instruction has not been entered before the first symbol definition instruction, all following instructions could be performed from different locations than expected, producing a symbol that does not appear as intended.*

Related Commands

MXYA
.Q
.S

[SAVE — Save Area

Format

[SAVE ***filename***

Purpose

Save the contents of the defined area in a file.

Remarks

The area must already have been defined; see **[D**.

[SAVE creates a file with the specified filename if one does not already exist. The filename extension .ara is automatically added unless a different extension is explicitly given.

If a file with the filename given already exists, the system asks

```
OK to save into existing file (Y/N)?
```

If the response is Y for yes, the file is overwritten with the current information.

If the response is N for no, the **[SAVE** command is ignored.

Symbols and alphanumeric fields are treated as being within the area to be saved only if they are completely within the area boundary. Drawing elements intersecting the area boundary are not saved. Lines are broken at the area boundary and the segments inside are saved.

Related Commands

[D
[LOAD

SAVE — Save Drawing

Format

SAVE [*filename*]

Purpose

Write the current drawing to the specified file.

Remarks

If the file specified already exists when the Save Drawing command is entered, the system displays

```
OK to save into existing file (Y/N)?
```

Y overwrites the file. N cancels the command.

When no file with the specified *filename* exists, one is created. The filename extension .dwg is automatically added unless an extension is explicitly given. When *filename* is omitted, the Save Drawing command writes the drawing to the filename from the last Load Drawing command. Use the **FILE** command to check the name.

Note: *If the loaded file has a .bak extension and you enter the SAVE command, the system will prompt you for a filename to avoid overwriting your backup file.*

When *filename* is omitted and the drawing is a new one which has not yet been saved, the system will prompt for *filename*.

When used without *filename*, **SAVE** copies the original file to a file with a .bak extension and saves the drawing in the work space to the old *filename*.

The Save Drawing command can be used during an editing session to save working drawings in order to prevent significant loss of work due to power failure or editing errors.

The drawing in memory is unchanged after the Save Drawing command is completed so you can continue editing.

SAVE with a filename specification saves a copy of the file in the name specified. This form of the command can be useful in saving copies of a file to different filenames while continuing to make additional changes to the original file.

SAVE operates on one file only, not on a whole drawing structure.

Related Commands

SAVEALL
AUTOSAVE

SAVEALL — Save All Changed Drawing Files

Format

SAVEALL

Purpose

Save all changes made in all drawing files accessed and modified since the editing session began or since the last **SAVEALL** was entered.

Remarks

As you move between files in a design hierarchy, the drawings and any changes are saved in temporary drawing files. These temporary files include information as to whether the drawing was changed or just viewed. When the **SAVEALL** command is entered, it determines which files have been modified and saves those temporary files.

The **SAVEALL** command can be entered from any drawing level of the design. It performs the save function on every modified drawing in the hierarchy. You can still move up and down in the hierarchy after entering the **SAVEALL** command.

If you do not want to update all of the drawings accessed and modified, move through the structure to the drawings you do want to update and enter the **SAVE** command at each one.

Note: This command is SAVEALL, one word, with no space between SAVE and ALL. If the command SAVE ALL is entered instead of SAVEALL, you will save the current file to a file named all.dwg and all other accessed drawings will remain in temporary files with potential for loss unless SAVEALL is entered properly prior to quitting the editing session.

Related Commands

SAVE

SD — Select Decrement Mode

(Symbol Definition Mode)

Format

SD

Purpose

Decrement the opposite coordinate by one dot unit (x if y is drawn short or y if x is drawn short) following any Draw Line Short in Display Units (**DXS** or **DYS**) instruction.

Remarks

Not supported by FutureNet OEM products.

The **SD** assists in drawing curves by making it easier to draw a stepped set of varying length line segments.

Once initiated, decrement mode remains in effect until turned off by the **SN** or **SI** instruction described in this chapter.

In decrement mode, following entry of the **DYS** (Draw y Short) instruction, the x coordinate is decremented (moved left) by one dot unit while the y coordinate is extended by one dot unit in the direction of the line segment drawn. Similarly, in decrement mode, following entry of the **DXS** (Draw x Short) instruction, the y coordinate is decremented (moved up) by one dot unit while the x coordinate is extended by one dot in the direction of the line segment drawn.

The **DXS** and **DYS** instructions are altered to interpret the length of the line drawn in dots instead of display units.

The line drawn is one dot longer than the length specified in the draw short instruction; that is, a line of length zero is one dot long, while a line of length one is two dots long, etc.

While in decrement mode, use the **DX, DY, DXD,** and **DYD** instructions to draw a line of any length (-128 to +127), including short lengths, without decrementing the opposite coordinate.

Related Commands

DX
DY
DXD
DYD
DXS
DYS
SI
SN

SI — Select Increment Mode

(Symbol Definition Mode)

Format

SI

Purpose

Increment the opposite coordinate by one dot unit (x if y is drawn short or y if x is drawn short) following any Draw Line Short in Display Units (DXS or DYS) instruction.

Remarks

Not supported by FutureNet OEM products.

The **SI** instruction assists in drawing curves by making it easier to draw a stepped set of varying length line segments.

Once initiated, increment mode remains in effect until turned off by a Select No Increment/Decrement or Select Decrement Mode instruction described in this chapter.

In increment mode, following entry of the **DYS** (Draw y Short) instruction, the x coordinate is incremented (moved to the right) by one dot unit while the y coordinate is extended by one dot unit in the direction of the line segment drawn. Similarly, in increment mode, following entry of the **DXS** (Draw x Short) instruction, the y coordinate is incremented (moved down) by one dot unit while the x coordinate is extended by one dot in the direction of the line segment drawn.

The **DXS** and **DYS** instructions are altered to interpret the length of the line in dot units instead of display units.

The line drawn is one dot longer than the length specified in the draw short instruction; that is, a line of length zero is one dot long, while a line of length one is two dots long, etc.

While in increment mode, use the **DX, DY, DXD,** and **DYD** instructions to draw a line of any length (-128 to +127), including short lengths, without incrementing the opposite coordinate.

Related Commands

DX
DY
DXD
DYD
DXS
DYS
SD
SN

SIZE — Set Drawing Size

Format

SIZE [A | B | C | D | E], [B | H | E | I | F | O | M | P | T]
SIZE *width,height*
SIZED *width,height*
SIZEM [A5 | A4 | A3 | A2 | A1 | A0]
SIZEM *width,height*

Purpose

Specify the drawing size.

Remarks

Drawings can be specified in standard drawing sizes A - E, metric drawing sizes A5 - A0, inches, display units, and millimeters. The **SIZE** status field always gives the drawing size in display units; plus, if one of the standard sizes A - E or metric drawing sizes A5 - A0 have been selected, those are also given. Knowing the drawing dimensions in display units is useful since the cursor location is shown in display units in the XY status field and all cursor movement is calculated in display units.

The default drawing size is B, which is 17 inches wide by 11 inches high.

There are approximately 20 display units per inch on both x and y axes.

The printer parameter is provided to help compensate for physical limitations of some printers. For more information on this parameter, see the discussion on printer compensation below.

SIZE

Standard drawing sizes A - E, and their equivalent in inches and display units are shown below. The range for width in inches is 7.5 to 102.4. The range for height in inches is 3.8 to 102.4. Indicate tenths of inches using a decimal point.

Following is a table of the standard drawing sizes and their display unit equivalents. Note that this table assumes that no printer option is specified in the SIZE command. This table reflects true sizes without compensation for a particular printer. See the discussion below on printer compensation for more information.

Size	Inches (wxh)	Display Units (wxh)
A	11 x 8.5	220 x 170
B	17 x 11	340 x 220
C	22 x 17	440 x 340
D	34 x 22	680 x 440
E	44 x 34	880 x 680

SIZE is used to specify the width and height of a drawing in standard drawing sizes A - E or in inches.

SIZED

SIZED is used to specify the width and height of a drawing in display units. The range for width in display units is 150 to 2048. The range for height in display units is 75 to 2048.

SIZEM

Following is a table of the standard metric drawing sizes and their display unit equivalents. Note that this table assumes that no printer option is specified in the SIZE command. This table reflects true sizes without compensation for a particular printer. See the discussion below on printer compensation for more information.

Size	Millimeters (wxh)	Display Units (wxh)
A5	210 x 147	165 x 116
A4	297 x 210	234 x 165
A3	420 x 297	330 x 234
A2	594 x 420	467 x 330
A1	840 x 594	662 x 467
A0	1190 x 840	936 x 662

Metric drawing sizes A5 - A0 and their equivalents in millimeters and display units are shown below. The range for width in millimeters is 191 to 2602. The range for height in inches is 96 to 2602.

Changing Drawing Size

To reduce the size of a drawing after drawing has started, use the following method:

1. Save the drawing as an area using the Area Editing Commands.
2. Open a new drawing and select a size using any of the SIZE commands.
3. Load the area into the new drawing.

Note: This procedure will not scale a drawing down to a different size. For example, if you have a D size drawing, which is 680 x 440 display units, and you want to put the drawing into a B size drawing, which is 340 x 220 display units, you can only copy a B sized portion of the D drawing.

Printer Compensation

Most printers cannot print on an entire sheet of paper. Compensation for this physical limitation is accomplished by limiting the drawing area within FutureNet to match the physical limitations of the printer. The following compensation factors are provided using the printer parameter on the size commands:

Value	Compensation Factors
B or H	Reduces the drawing space by one inch on each size, two inches in height and two inches in width. Use when drawing will be plotted with DPlot on a plotter device.
E or I	Reduces the drawing space by 10% on the x axis and 25% on the y axis. Use with printers similar to the Epson MX family or Epson FX family in IBM emulation.
F or O	Reduces the drawing space by 10% on both the x axis and the y axis. Use with printers similar to the Epson FX family or Okidata printers.

M Reduces the drawing space by 20% on both the x axis and the y axis. Specifically designed to create Milspec drawings when scaled by 125% in DPLOT.

T Reduces the drawing space by 10% on the x axis. The y axis is not affected by this option. Use with printers similar to the C. Itoh 1550.

P No compensation for either axis. This is the default setting.

Use the option that is most appropriate for your printer. Some experimenting may be necessary to find the correct option for your particular printer. The SIZED command can also be used to specify exactly the size of the drawing area in display units.

SL — Select Broken Line Type

(Symbol Definition Mode)

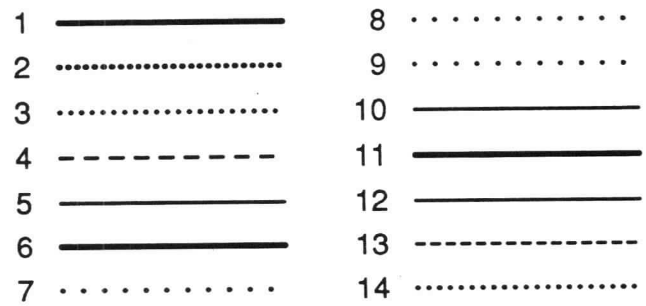
Format **SL 0 | 1 | 2 | 3 | 4 | ... | 14**

Purpose Select line type.

Remarks *Not supported by FutureNet OEM products.*

Select the type of line that will be drawn by BX and BY instructions.

The current line type is shown in the Line Status box.



This command has no effect on the symbol editor coordinate marker.

Related Commands BX
BY

SN — Select No Increment/Decrement Mode

(Symbol Definition Mode)

Format

SN

Purpose

Restore the normal operation of the Draw Short instructions, terminating the specialized operation of the **SD** and **SI** instructions.

Related Commands

DXS
DYS
SI
SD

ST — Select Symbol Element Table

(Symbol Definition Mode)

Format

ST 0 | 1 | 2 | 3 | 4 | 5 | 6

Purpose

Select the ASCII symbol element table and specify the font size of the characters used as symbol elements when using the IS*xx* command.

Remarks

Not supported by FutureNet OEM products.

The ASCII symbol element table (an appendix in the *FutureNet User Manual*) is an alternative to the default symbol element table (which is selected by entering the **RS** command).

This command allows addition of text in various fonts to a symbol at a specific pixel location, rather than at a display unit location.

Command	Selects ASCII Character Size
ST 0	'1
ST 1	'2
ST 2	'3
ST 3	'4
ST 4	'5
ST 5	'6
ST 6	'7

The **ST** command must precede the Insert Symbol Element (ISxx) commands that are to be affected.

Once **ST** is selected, the Insert Symbol Element command uses the selected character set until another **ST** command, or until an **RS** command resets the symbol element table to the default shapes.

Related Command

RS
ISxx

STOP — Stop Command File Run

Format

STOP
NOAUTO
NOEXEC

Purpose

Stop automatic or single-step execution of a command file.

Remarks

STOP, NOAUTO, and **NOEXEC** can be entered anytime to stop **AUTO** or **EXEC** command file execution. **NOAUTO** and **EXEC** are synonyms for **STOP.**

To enter, type **STOP** on the command line while the current command file is being entered in either **AUTO** or **EXEC** mode.

Related Commands

AUTO
EXEC
PAUSE

SXY, RXY— Save/Restore x and y Coordinates

(Symbol Definition Mode)

Format

SXY
RXY

Purpose

Save the current *x,y* values and then restore them after they have been changed by other drawing instructions.

Remarks

Not supported by FutureNet OEM products.

SXY saves the current *x,y* coordinate values. This instruction does not affect the symbol editor coordinate marker location.

RXY restores the *x,y* coordinate values saved by the last **SXY** instruction. The symbol editor coordinate marker is restored to the old *x,y* location.

Note: If an RXY command occurs in the symbol definition list, before an SXY command, then the default restored location is (0,0).

Tab, Shift – Tab Keys— Move Cursor Down/Up a Field

Keys

Tab	(Down)
Shift – Tab	(Up)

Purpose

When in alphanumeric mode, move the cursor to the beginning of the next or previous alphanumeric field within a symbol cell.

Remarks

Tab moves the cursor to the beginning of the next alphanumeric field in the symbol cell.

Shift – Tab moves the cursor to the beginning of the previous alphanumeric field in the symbol cell.

Tab fields are found moving left to right, top to bottom.

Shift – Tab fields are found moving right to left, bottom to top.

These keys work only within a symbol cell, and will not move the cursor between symbol cells.

If the next alphanumeric field to be reached is not entirely visible on the screen, the display shifts so that the cursor is centered on the drawing screen.

Related Commands

Esc

#U — Move Up in Design Hierarchy

Format #U[*filename*]

Key

PC	SUN
Ctrl – PgUp	Ctrl – R9

Purpose

Move back up in the design hierarchy and show a previously loaded drawing file.

Remarks

#U[*filename*] saves the current drawing file to a temporary file and reloads a previously loaded drawing file that is on the current drawing path but at a higher level in the hierarchy. The current drawing file is saved in a temporary file with a unique filename based on the root filename with a numbered extension. If the filename exists, the numbered extension is incremented until the filename is unique.

Ctrl – PgUp, Ctrl – R9 or #U without a filename reloads the file above the current file in the hierarchy.

Pressing Ctrl – PgUp or Ctrl – R9 repeatedly traverses the drawing structure up to the root. Once in the root drawing, the Move Up in Design Hierarchy command has no effect.

Related Commands

#D
#L
#R
SAVEALL

'UNDER — Add/Delete Underscore to Alphanumeric Field

Format

'UNDER [ON | OFF]

Purpose

Adds or deletes a field underscore.

Remarks

Position the graphics cursor on the desired alphanumeric field and enter the command.

If no parameter is specified, the command toggles the current setting.

Spaces within the text field will not cause a break in the underscore.

'UNDER can only be entered on an existing field.

While in alphanumeric mode, an underscore can be added or deleted by using the Ctrl – U keystoke sequence.

When an existing field that already has an underscore is edited, the underscore will be added or deleted as text is added or deleted.

Related Commands

'OVER

UNDO/REDO — Reverse/Restore Commands

Format

UNDO [ON | OFF]
REDO [ON | OFF]

Purpose

UNDO allows step-by-step reversal of actions done during the current drawing session. **REDO** reverses the **UNDO** feature, redoing actions that have been undone.

Remarks

UNDO and **REDO** are available on most configurations supported by FutureNet. The following configurations are supported.

- All Sun systems
- PC systems with Extended Memory (using the Extended Memory executable)
- PC systems with Expanded Memory or Expanded Memory drivers (using the Non-Extended Memory executable)

UNDO/REDO works from a stack of actions that is built as a drawing is edited. **UNDO** is ON by default if you have EMS. Entering **UNDO OFF** or **REDO OFF** disables this feature, and clears the stack.

You are able to move forward or backward in the stack at will. The stack has the following limitations:

- Only actions which affect drawing graphics are placed on the stack. For instance, accesses of the **PROFILE** screen or menus are ignored, as are file commands such as **SAVE**.
- Failed operations are not stacked. For instance, if you attempt to draw a line that results in an "Invalid line routing" error and do not reroute the line correctly, the entire operation is ignored.
- Entering the **LOAD, CLEAR, #L, #R, #U** and **#D** commands clear the stack.

/V — Move Vertex

Format

/V

Purpose

Tag any point along a line or the intersection of two line segments and move the line without moving the end points, or move one end point of a line without moving the other end point.

Remarks

Position the graphics cursor at the desired location, either a point on a line or the intersection to two line segments and enter the /V command. A two display unit square will appear, with its center at the tagged point, indicating that the point has been tagged. Move the tagged vertex to the desired location and re-enter the /V command to complete the operation.

Lines moved using the /V command will be rubberbanded where necessary to maintain connectivity. Depending on where the lines have been moved to, it may not be possible to reroute them without creating temporary lines.

The left mouse button can also initiate this command. Move the graphics cursor to desired location on the line and press the left mouse button twice. The first press puts the system in LINE mode and the second press puts the system in tag mode. Assuming the cursor coincides with a point on a line or a vertex, the line will be tagged. If other graphic elements (symbols, text fields, or areas) are in close proximity, they may be tagged first. Continue pressing the left mouse button, moving through the tagging hierarchy, until the line or vertex is tagged.

This command works with permanent and temporary lines. It does not work with direct connect lines.

Related Commands

/L

VERSION — Display Software Version

Format

VERSION
VER

Purpose

Display the version of FutureNet software presently installed in the system.

Remarks

The version of software is also displayed in the message area when FutureNet initializes.

This information may be helpful in any communication with a Data I/O Customer Support office.

VIEW — View Drawing

Format

VIEW [ON | OFF]

Purpose

Toggles screen update ON/OFF.

Remarks

ON is the default setting. **VIEW** without any parameters toggles the setting.

When ON, the screen is updated as the commands are entered.

When OFF, screen updating is suspended.

VIEW can be entered on the command line or as part of a command file. It can be entered more than once, either on the command line or in command files. In command files, it can appear anywhere in the list of commands.

Command files will run to completion faster with **VIEW OFF** rather than ON because there will be no need to display graphics. The lines, symbols, areas, and text fields are created in the same manner whether **VIEW** is ON or OFF, but you don't see them occurring when **VIEW** is OFF.

Type **VIEW ON** at any time to see the results of the commands that were run since **VIEW OFF** was used.

Related Commands

AUTO
EXEC

.w,h — Define Symbol Cell Size

Format

.width,height

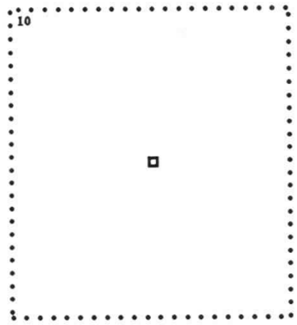
Purpose

Define or redefine the size of the symbol cell.

Remarks

Width and *height* are defined in display units.

This command is also available in the Symbol Definition Mode.



In drawing mode, the cursor must be inside a symbol cell before this command can function. Symbol size should be specified before starting the symbol definition instruction list.

An existing symbol cell can be made larger or smaller by specifying different width and height dimensions. .0,0 will remove the boundary from the screen.

If the symbol cell defined does not fit in the drawing at the graphics cursor location, the system displays

```
Symbol boundary conflict
```

Before the symbol cell can be set to the desired size at the current location, all boundary conflicts must be resolved by moving or erasing the other drawing elements that interfere with the new symbol. The size of the symbol boundary is restricted only by symbol boundary conflicts.

If no symbol cell size has been previously specified, a default of w=1 and h=1 is provided for a newly created symbol. Symbols may extend beyond the limits of the symbol cell; however, no boundary checks beyond the symbol cell size are done.

Related Commands

.S

WINDOW — Set Window Locations

Format

WINDOW 1 | 2 | 3 | 4
WINDOW OFF
NOWIN

Purpose

Specify up to four window locations that can be searched to using the **PAN** command.

Remarks

Assign a number from 1 through 4 to a window location.

The **WINDOW** command can be entered at any zoom level.

Fit (**ZIN** ***n***) zoom is the best zoom mode in which to assign window numbers because the entire drawing, including the window position relative to other windows and elements, is displayed.

The full-scale window (the rectangular box which appears at fit zoom level) frames the area of the drawing that will be assigned a window number.

Window locations saved with the **WINDOW** command are displayed in fit zoom as dotted rectangles with the window number in the upper left corner.

NOWIN or **WINDOW OFF** cancels all the saved window locations.

Related Commands

PAN
ZIN
ZOUT

ZIN/ZOUT — Change Zoom Level

Format

ZIN [0 | 1 | 2 | n]
ZOUT [0 | 1 | 2 | n]

Keys

PC	SUN	
PgUp	R9	(ZIN)
PgDn	R15	(ZOUT)

Purpose

Select half, full, intermediate, or fit zoom.

Remarks

Depending on the current zoom level, repeatedly entering **ZIN** or pressing R9 cycles from fit to half; **ZOUT** or R15 , from half to fit. To zoom to a specific level, use the appropriate parameter.

The ZOOM status field displays the value of the zoom level where full zoom (**ZIN 1** or **ZOUT 1**) is considered as 1.

0	Half (objects appear largest, roughly twice actual size)
1	Full (objects appear actual size)
2	Intermediate (midway between full and fit)
n	Fit (entire drawing visible)

Full zoom is the default zoom level.

ZIN and **ZOUT** can be entered on the command line and in macros.

Half Zoom

Half zoom displays a small portion of the drawing, at roughly twice actual size. .70 is displayed in the ZOOM status field. All drawing editing commands operate in full zoom.

All symbols, their alphanumeric text, and all connecting lines that fit in the window are displayed.

Full Zoom

Full zoom displays a portion of the drawing, with all objects appearing nearly actual size. 1.00 is displayed in the ZOOM status field. The full range of editing commands are available, just as in half zoom.

Intermediate Zoom

Intermediate zoom displays the drawing in a scale that is the average of the drawing area in fit and full zoom. The ZOOM status field displays a value midway between 1 and the zoom value of fit zoom.

Outlines of the symbols and their connecting lines that fit on the screen are displayed, but alphanumeric text is not. Alphanumeric field boundaries display when the **'B** command is entered.

Fit Zoom

In fit zoom, the entire drawing is displayed in the scale required for it to fit exactly on the screen. Outlines of the symbols and their connecting lines are displayed, but not alphanumeric text. A rectangular boundary, called a full scale window, outlines the portion of the drawing that would be displayed in full zoom. This window is centered around the cursor and moves as it moves. Also shown are any saved windows and their window numbers. Alphanumeric field boundaries appear when the **'B** command is entered.

In fit zoom, the **HOME** command has no effect on the cursor. The graphics cursor can be moved using the various move commands (mouse, Move to Coordinates, arrows, etc.) to anywhere in the drawing.

Lines can be drawn and symbols manipulated in fit zoom and the full scale window can be relocated using the mouse and command line.

Related Commands

ZOOM

ZOOM — Dynamic Zoom Level

Format

ZOOM

Purpose

Define and display an area at a custom zoom level.

Remarks

When the **ZOOM** command is entered, the zoom cursor appears at the current cursor location, and ZOOM appears in the MODE status field. Using the mouse or with cursor control, a zoom box may be stretched in any direction to enclose an area with one corner at the original cursor location. When the command is reentered, the area within the zoom box grows to fill the entire window, and the mode returns to the previous mode.

The ZOOM status field is updated to show the level of zoom compared against full zoom=1. The smallest zoom level possible is .1.

When a zoom box is being defined, ZOOM mode may be cancelled by pressing the right mouse button.

Use the **ZIN/ZOUT** commands or keys to move to a pre-defined zoom level once you are finished with dynamic zoom.

Related Commands

ZIN
ZOUT

Index

! (exclamation)

(pound)

' (apostrophe)

* (asterisk)

. (period)

/ (slash)

[(bracket)

? (question mark)

A

FutureNet®

Schematic Designer

Installation Guide (DOS)

DATA I/O

980-0010-006

January 1993 980-0010-006

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA
(206) 881-6444

Acknowledgments:

1 Installation

This Installation Guide details the procedures required to install the FutureNet® 6.10 Schematic Designer software on IBM PC/XT/AT, IBM PS/2, or compatible systems.

To install part or all of FutureNet 6.10, the following steps are required.

1. Install software.
2. If necessary, edit your **autoexec.bat** and **config.sys** files (see Chapter 2, "Installation Options").
3. Reboot computer.

System Requirements

To use FutureNet on your computer system, you need at least the following:

- 640K RAM.
- Fixed disk drive (40MB minimum). About 20MB of space on your fixed disk are required to install all the FutureNet programs and libraries on your system. (It is not necessary to install all the programs and libraries.)
- High density (1.2MB) 5.25-inch disk drive or a high density (1.44MB) 3.5-inch disk drive.
- DOS version 3.3 or later.

- One of the following or compatible graphics displays and drivers.

Extended Memory	**Non-extended Memory**
EGA Enhanced Color Display	8514/A Color Display
EGA Monochrome Display	Compaq Portable Plasma Display
Everex VGA High Resolution	EGA Enhanced Color Display
Hercules Monochrome Graphics	EGA Monochrome Display
Toshiba 3100 Monochrome Display	FutureNet Monochrome Graphics
Paradise VGA High Resolution	GRiD Laptop Monochrome
VGA Monochrome Display	Hercules Monochrome Graphics
VGA Color Display	Microfield T4 Color Display
Video 7 VEGA VGA High Resolution	Toshiba 3100 Monochrome Display
VGA High Res, per VESA specifications (various)	VGA Color Display
Orchid Pro Designer VGA Hires	VGA Monochrome Display
ATI Wonder VGA–EGA emulation	
ATI Wonder VGA Standard Resolution	
ATI Wonder VGA High Resolution	
Diamond Flower Instruments VGA	
Compaq Portable Plasma Display	
TsengLab ET4000 SuperVGA ChipSet (example: Orchid ProDesigner VGA)	

- A mouse (recommended). A three-button mouse (such as the LOGITECH C9 LOGIMOUSE) is preferable, but a two-button mouse (such as the Microsoft Mouse) will perform satisfactorily. (The action of the middle button of a three-button mouse can be duplicated by pressing both mouse buttons simultaneously.)

  Any mouse interface that is compatible with the Microsoft Mouse driver, version 5.03 or later, will be compatible. The mouse interface is a software interface.

- (Optional for better DPLOT performance) an 8087, 80287, 80387, or 80487 math coprocessor. Some DPLOT drivers require a math coprocessor (see the *DPLOT User Manual* for more information).

Software Installation

The FutureNet software is provided on both high density (1.2MB) 5.25-inch disks and high-density (1.44MB) 3.5-inch disks.

Note: *You cannot use the DOS COPY command to install FutureNet. The installation program must be run to install the FutureNet program files.*

Note: ***Update Customers:*** *FutureNet 6.1 does not require a security device. If you will be using Version 6.1 exclusively, you can remove any FutureNet security devices from your computer's parallel port.*

1. To begin installation, insert Disk 1 of the master distribution disks into drive A.
2. At the system (DOS) prompt, enter

   a:\install

   and follow the instructions on your display.

Note: *If you want to install the software from a drive other than A, at the system prompt, enter*

***disk_drive*:\install**

*where **disk_drive** is the name of the disk drive where Disk 1 of the master distribution disks is located.*

This concludes the software installation.

If the installation program has made all needed changes to your **autoexec.bat** and **config.sys** files, you can run FutureNet by rebooting your computer and, from the system prompt, entering

fn

Otherwise, see Chapter 2, "Installation Options," for information on editing your **autoexec.bat** and **config.sys** files.

For information on FutureNet directories and environment variables, see Chapter 2, "Installation Options."

Readme Files

The readme files (which can be found in the **fnreadme** directory) contain additional information about this release of FutureNet. The format of the readme files is

prefix*_READ.*ext

You can print a copy of the readme files using the DOS PRINT command.

2 *Installation Options*

This chapter discusses each of the environment variables and system configuration options.

Note: The installation program does not update many environment variables including FNLOG, FNELOG, and FNPRO. If you wish to use these environment variables, you will have to set them in your **autoexec.bat** *file. See the "Editing Your autoexec.bat File" section later in this chapter.*

Editing Your config.sys File

If you did not have the FutureNet 6.10 Installation program update your **config.sys** file, you will have to update this file yourself.

For information on the use of the **config.sys** file, refer to the DOS Reference manual.

Ensure System Performance

Configure DOS version 3.3 or later for improved performance by adding or editing the following commands in your **config.sys** file:

BUFFERS=40
FILES=40

DPLOT Environment Requirements

If DPLOT was installed, your system configuration file (**config.sys**) must be modified to use the DPLOT package. The modifications necessary depend on which driver you will be using. The following drivers are included with the standard FutureNet package:

- Hewlett Packard 7400 and 7500 series and HPGL compatible plotters
- Houston Instruments DMP 41, 42, 51, and 52 plotters

Note: For detailed information about configuring your system for DPLOT, see the "Troubleshooting" chapter in the DPLOT User Manual.

Setting the Serial Port

Both the Hewlett Packard and Houston Instruments drivers require installation of a serial port device driver which is provided with the DPLOT installation. The device driver must be added to your **config.sys** file using the following entry:

DEVICE=*drive*:\dplot\serial*x*.sys

where *drive* is the drive letter where DPLOT was installed, and the *x* in serial*x*.sys is the serial port the plotter is attached to. For example, if the plotter is attached to COM1 then *x*=1; if COM2 then *x*=2.

Setting the Baud Rate

The computer must communicate with the plotter at a known baud rate. The following command must be executed to set the correct communication protocol:

MODE COM*x*: *baud_rate*, N, 8, 1

where *x* represents the serial port the plotter is attached to and *baud_rate* represents the appropriate baud rate for the plotter. Set *baud_rate* to 9600 for Hewlett-Packard plotters and 2400 for Houston Instruments plotters.

Placing the MODE command in your **autoexec.bat** file will cause the correct protocol to be set automatically when your machine is booted.

Editing Your autoexec.bat File

If you did not have the FutureNet Installation program update your **autoexec.bat** file, you will have to update this file yourself. If you do not have an **autoexec.bat** file, you have to create one.

Note: *The installation program does not update many environment variables including FNLOG, FNELOG, and FNPRO. If you wish to use these environment variables, you have to set them in your* **autoexec.bat** *file.*

Consult the **autoexec.new** file (found in your root directory) for the changes that you need to make to your **autoexec.bat**. (The FutureNet installation program automatically creates the **autoexec.new** file based on the choices you made during installation.)

For information on the use of the **autoexec.bat** file, refer to the DOS Reference manual.

The Search Path

Information about where FutureNet resides can be provided to DOS through the use of the DOS PATH command. The PATH command can be executed from the **autoexec.bat** file. You may either edit a pre-existing **autoexec.bat** file or create one with any standard ASCII text editor.

If you do not already have an **autoexec.bat** file, create one and insert the following command:

PATH=*drive:path*

where *drive* is a disk drive designator, and *path* is the full DOS path specification to the directory where FutureNet resides.

For example, the default directory for the extended memory (XMS) version of FutureNet is c:\dataio\fnx. Add the following path statement:

PATH=c:\dataio\fnx

If you already have an **autoexec.bat** file, add the directory path name where FutureNet resides to the end of any existing PATH statement; for example,

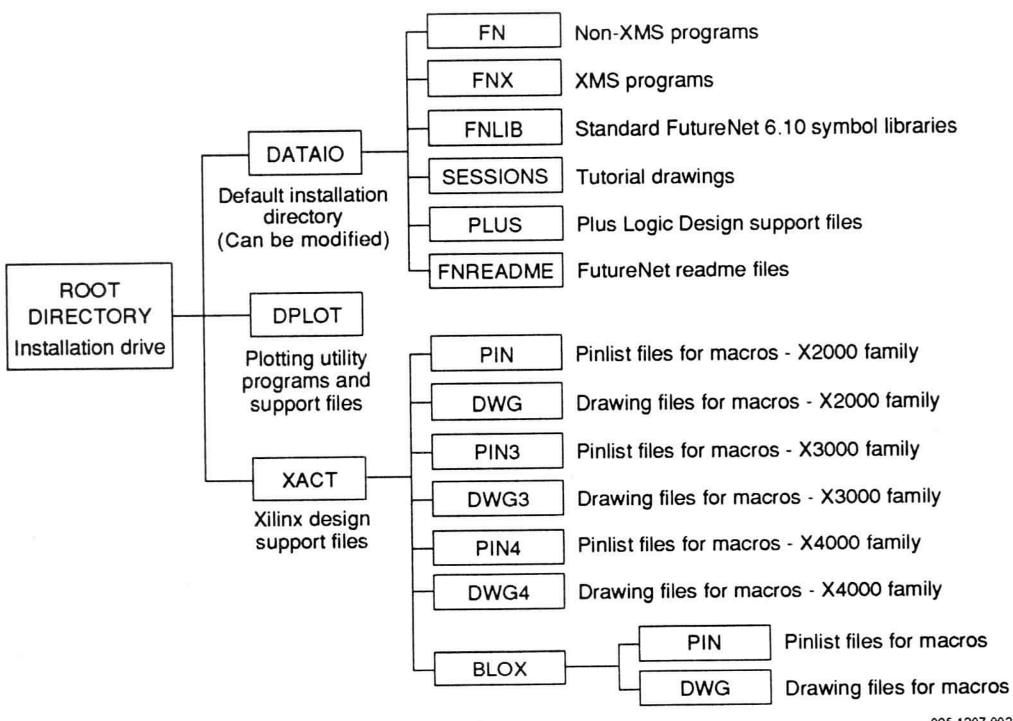
PATH= ***existing_path*****;c:\dataio\fnx**

The installation procedure automatically adds to an existing PATH statement the path specification for the default installation of FutureNet. If you move FutureNet from its default directory, remember to change the PATH statement.

Directories

Any subdirectory can be chosen for installation. If the subdirectory does not exist, it will be created. The default directory is \dataio on the drive chosen for installation. Although the subdirectory for installation can be chosen, the directory tree beneath this subdirectory cannot be manipulated using the install program. If the default directory tree structure is not acceptable, use DOS commands to alter the structure after the install has completed.

The files on the installation disks are copied to various directories. The illustration below shows the directory tree which will be found if a complete installation is done. Some directories may not be created due to choices made during the installation procedure.



Note: The subdirectories \dplot and \xact must be located in the root directory.

Mouse Sensitivity

If you are using the Logitech mouse driver that comes with FutureNet, during installation you can set the mouse sensitivity, or you can add the following statement to your **autoexec.bat** file:

mouse /s*num* /*port*

where *num* is the two digit sensitivity setting, ranging from 01 to 10 (including the leading zero for values less than 10), and *port* is the port where your mouse is located.

Note: Mouse sensitivity in FutureNet may be adjustable depending on the mouse drivers. Most mouse drivers allow you to control the mouse sensitivity. See your mouse documentation if you wish to change the mouse sensitivity.

Environment Variables

FutureNet uses environment variables to allow customized installation and to control contention on networked systems.

These environment variables may be entered at an operating system prompt, or specified in your **autoexec.bat** file. An environment variable specified at the DOS prompt will override one set in the **autoexec.bat** file. If requested, the installation program will modify your **autoexec.bat** file to set the environment variables. Otherwise, you will need to note the setting for the variable in the **autoexec.new** file.

If environment variables will be used with FutureNet, they must be set prior to starting FutureNet.

Checking Environment Variables

You can find out which variables are enabled by entering the following command:

set

A list of the environment variables that are set is displayed. For example, your system might display

```
PATH=c:\; c:\dos; c:\dataio\fn
FNLIB=c:\dataio\fnlib
FG_DISPLAY=VGA12
```

FNDISPLAY and FG_DISPLAY

The FNDISPLAY and FG_DISPLAY environment variables are used to specify which display driver should be used. Specifying a display driver is required. The FNDISPLAY environment variable is used by non-extended memory DOS systems only, and the FG_DISPLAY environment variable is used by extended memory (XMS) DOS systems only.

The installation procedure allows you to automatically add FNDISPLAY and/or FG_DISPLAY to an existing **autoexec.bat** file.

FNDISPLAY

The FNDISPLAY environment variable is used to tell the non-extended memory version of FutureNet which display driver should be used. The syntax for the variable is

set FNDISPLAY=[*drive*:]\[*pathname*]*driver*[.dg]

where *drive* and *pathname* are optional path designators for the location of the desired driver, and *driver* designates which of the display drivers is to be used. For example,

set FNDISPLAY=c:\dataio\fn\egaecd.dg

If an explicit path to a display driver is specified, then FutureNet will look in the designated directory only. If no path is specified, only the driver name, then the following search priority is used, searching for the designated driver.

If FNDISPLAY is not specified, FutureNet will search for the driver ALL.DG using the following search priority:

1. Current directory
2. The directory from which FutureNet was executed
3. The system path

If no driver is found then the message

```
Can't initialize graphics
```

will appear and FutureNet will not run.

During installation of the FutureNet executables, all of the display drivers (.DG files) are copied into the non-extended memory directory. The FNDISPLAY environment variable names are derived from the display driver filename (the .DG extension is optional). For example, EGAECD uses the **egaecd.dg** display driver.

The non-extended memory version of FutureNet supports the displays and drivers shown on the following table.

FNDISPLAY Value	Driver Description	Pixels
8514A	8514/A Color Display	1024 X 768
COMPAQPL	Compaq Portable Plasma Display	640 x 400
EGAECD	EGA Enhanced Color Display	640 X 400
EGAMONO	EGA Monochrome Display	640 X 400
FNET	FutureNet Monocrome Graphics	640 x 400
GRID	GRiD Laptop Monochrome	640 x 400
HERC	Hercules Monochrome Graphics	640 X 400
T4	Microfield T4 Color Display	1024 x 768
TOSHIBA	Toshiba 3100 Monochrome Display	640 X 400
VGA12	VGA Color Display	640 X 480
VGA11	VGA Monochrome Display	640 X 480

FG_DISPLAY

The FG_DISPLAY environment variable is used to tell the extended memory version of FutureNet which display driver should be used.

The syntax for the variable is as follows:

set FG_DISPLAY=*value*

where *value* designates which display driver will be used with FutureNet. For example,

set FG_DISPLAY=egaecd

The extended memory version of FutureNet supports the displays and drivers shown on the following table.

FG_DISPLAY value	Driver Description	Pixels
EGAECD	EGA Enhanced Color Display	640 X 400
EGAMONO	EGA Monochrome Display	640 X 400
EVGAHIRES	Everex VGA High Resolution	800 X 600
HERC	Hercules Monochrome Graphics	640 X 400
TOSHIBA	Toshiba 3100 Monochrome Display	640 X 400
COMPAQPL	Compaq Gas Plasma Display	640 X 400
VGA12	VGA Color Display	640 X 480
VGA11	VGA Monochrome Display	640 X 480
VEGAVGAHIRES	Video 7 VEGA VGA High Res	800 X 600
VESA6A	VGA High Res. VESA mode 0x6A	800 X 600
VESA0	VGA High Res. VESA mode 0x100	640 X 480
VESA1	VGA High Res. VESA mode 0x101	640 X 480
VESA2	VGA High Res. VESA mode 0x102	800 X 600
VESA3	VGA High Res. VESA mode 0x103	800 X 600
VESA5	VGA High Res. VESA mode 0x105	1024 X 768
VESA7	VGA High Res. VESA mode 0x107	1280 X 1024
PARADISEHIRES	Paradise VGA High Resolution	800 X 600
ATI161	ATI Wonder VGA - EGA emulation	640 X 400
ATI162	ATI Wonder VGA Standard Res	640 X 480
ATI163	ATI Wonder VGA High Resolution	800 X 600
DFIHIRES	Diamond Flower Instruments VGA	800 X 600
TSENGHIRES	TsengLab ET4000 VGA ChipSet (replaces Orchid ProDesigner support)	800 X 600

FNMOUSE

The FutureNet software needs to know if you are using a mouse. This is done by adding the following environment variable to your autoexec.bat file:

set FNMOUSE=*value*

where *value* is one of the environment variable options from the following table.

Memory Type	Environment Variable Option	Description
Extended	M	Microsoft-compatible mouse driver installed
	N	No mouse attached.
Non-extended	M	Microsoft-compatible mouse driver installed (recommended setting).
	1, 2, 3	Logitech C7 mouse on serial port COM1, COM2 or COM3, respectively.
	P1,P2,P3	Logitech C9 mouse on serial port COM1, COM2, or COM3, respectively.
	S1, S2, S3	Mouse Systems mouse on serial port COM1, COM2 or COM3, respectively.
	N	No mouse installed.

Refer to the documentation that came with your mouse for driver installation information.

Note: *If you use any FNMOUSE option other than* **M**, *running FutureNet with a mouse driver can cause erratic mouse behavior or program failure.*

If you have changed the value of FNMOUSE from M to some other value and wish to run FutureNet, remove the driver from memory by either rebooting your system without the driver, or clearing the driver from memory.

FNLIB

FNLIB indicates the default directory for FutureNet to search for symbol libraries opened with the .LIB and LIB commands. If a specified library is not in the current directory, then the directory specified by FNLIB is searched, unless the path to the library is specified on the **.lib** or **lib** command line.

Set the variable as follows

set FNLIB= [*drive*:]*path_to_directory*

For example, for the default installation, the symbol libraries are placed in the directory **c:\dataio\fnlib**, so setting FNLIB would look like

set FNLIB=c:\dataio\fnlib

FNPRO

Note: You can use the configuration file, fn.cfg, to configure this feature automatically. See HELP FN.CFG within FutureNet for more information.

FNPRO specifies a command file which will run immediately after FutureNet is loaded. This file can be useful for setting standard configuration items such as libraries to use, program settings, and color settings.

The DOS command to set the FNPRO environment variable is

set FNPRO=[*path*] *filename* [*.ext*]

You can set the FNPRO variable to any filename you choose. If no path is given, only the current directory is searched for the file. If a full path name is included, the same file is executed regardless of the current directory.

Including FNPRO in your **autoexec.bat** file and specifying a full path ensures that FutureNet starts in the same configuration.

If you want FutureNet to perform differently in different directories, leave the FNPRO variable unset and place **fnpro.cmd** (command) files in each FutureNet directory.

FNLOG

FNLOG specifies the file to which FutureNet will log commands for the current editing session. A new log file is created each time FutureNet is started. (This process overwrites the old log file if a complete path name is specified for the variable.) The log file contains the complete contents of the work session, and protects your work in case of a system or power failure. The log file can be used to document system problems to help reproduce the problem. We strongly recommend that you set this variable.

Note: To preserve a previous log file when FNLOG is enabled, you must rename it or temporarily disable FNLOG before restarting FutureNet.

The DOS command to set the FNLOG environment variable is

set FNLOG=[*path*] *filename* [*.ext*]

If the FNLOG variable is not set, or does not contain a filename, autologging is disabled. If no path is included, the log file is created in the directory FutureNet is started in. This makes it possible to have a log available in any directory in which you work.

Note: This environment variable is compatible with the old LOG variable, but will be checked first.

FNELOG

The FNELOG environment variable enables error logging when running a command file and logging of error messages that occur when running commands from the command line. When set, an error file is created each time you run a command file. The error file contains a list of the status and error messages generated while the command file is running, and can help diagnose problems in command files.

The DOS command to set the FNELOG environment variable is

set FNELOG=[*path*]*filename*

The FNELOG variable enables error file creation, sets the path, and gives the filename of the error file.

The error file contains messages from the last command file executed. Each time a command file is executed, messages from the previous file are overwritten.

Note: To preserve a previous error file when FNELOG is enabled, you must rename it or temporarily disable FNELOG before restarting FutureNet.

If the FNELOG variable is not set, or does not contain a filename, error logging is disabled. If no path is included, the error file is created in the current directory from which FutureNet was started. This makes it possible to have an error log available in any directory in which you work.

Note: This environment variable is compatible with the old DASHELOG variable, but will be checked first.

TMP

You may use the TMP environment variable to tell FutureNet where to write the temporary file when it performs the DOS shell command.

Note: The TMP variable is only used by the non-extended memory version of FutureNet.

The DOS command to set the TMP environment variable is

set TMP=*drive*:*path*

The name of the temporary file is called ztc*xxxx*.tmp.

CAUTION: Do not remove this temporary file in a DOS shell or you will not be able to return to FutureNet and may lose any unsaved work.

If TMP is not set, the current directory is used as the location for the temporary file.

XACT

The XACT variable specifies where the Xilinx drawing and pinlist files can be found. This variable is used by Xilinx XACT tools.

Note: For more information about the XACT variable, see the documentation that accompanied the XACT package.

FNSAVE

The FNSAVE variable specifies where to save the AUTOSAVE files.

The DOS command to set the FNSAVE environment variable is

set FNSAVE=*drive*:*path*

where *drive* is the logical disk drive designator and *path* is the fully-qualified path to the directory to save the files. For example, C:\FNSAVE. The directory must exist.

FNTMP

The FNTMP variable specifies where to save the temporary files created by FutureNet.

The DOS command to set the FNTMP environment variable is

set FNTMP=*drive*:*path*

where *drive* is the logical disk drive designator and *path* is the fully-qualified path to the directory to save the files. For example, C:\FNTMP. The directory must exist.

DEFAULTLIB

The DEFAULTLIB variable specifies a system library to use instead of the default system library as defined in the LIB command.

The DOS command to set the DEFAULTLIB environment variable is

set DEFAULTLIB=*drive*:*library specification*

where *library specification* is a fully-qualified path and filename for the desired library.

Serial Number Card

20057035

Keep This Card

You must have your serial number when you call for Technical Assistance. Data I/O reserves the right to refuse service to anyone calling without a serial number.

This serial number is also required during installation of the current release of software and all future releases. Place this card with your manual or in another safe place for reference.

Technical Assistance

To help us provide quick and accurate assistance, please be at your programmer or computer when you call, and have the following ready:

- Product serial number
- Product version number
- Detailed description of the problem you are experiencing
- Error messages (if any)
- Device manufacturer and part number (if device-related)
- Product manual

For technical assistance, contact the appropriate Customer Support office listed below.

United States and Canada

Data I/O Customer Resource Center
Telephone: 1-800-247-5700
Fax: 206-882-1043

Japan

Data I/O Japan (Tokyo)
Telephone: 3-3432-6991
Fax: 3-3432-6093
Telex: 2522685 DATAIO J

Germany

Data I/O GmbH
Telephone: (0)89-858580
Fax: (0)89-8585810

United Kingdom

Data I/O Limited
Telephone: (0)734-440011
Fax: (0)734-448700

Other European Countries

Contact the office below and ask for the number of your local Data I/O representative:

Data I/O Limited
Telephone: +44-(0)734-440011
Fax: +44-(0)734-448700

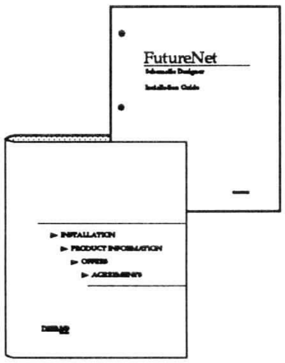
Other Countries Worldwide

Contact the office below and ask for the number of your local Data I/O representative:

Data I/O Intercontinental (U.S.)
Telephone: +1-206-881-6444
Fax: +1-206-882-1043
Telex: 4740166

FutureNet Documentation Map

Installation Guide
Information on installing FutureNet



FutureNet User Manual

Introduction — An introduction to FutureNet.

Understanding FutureNet — Descriptions of what a drawing needs to establish connectivity for the post processors.

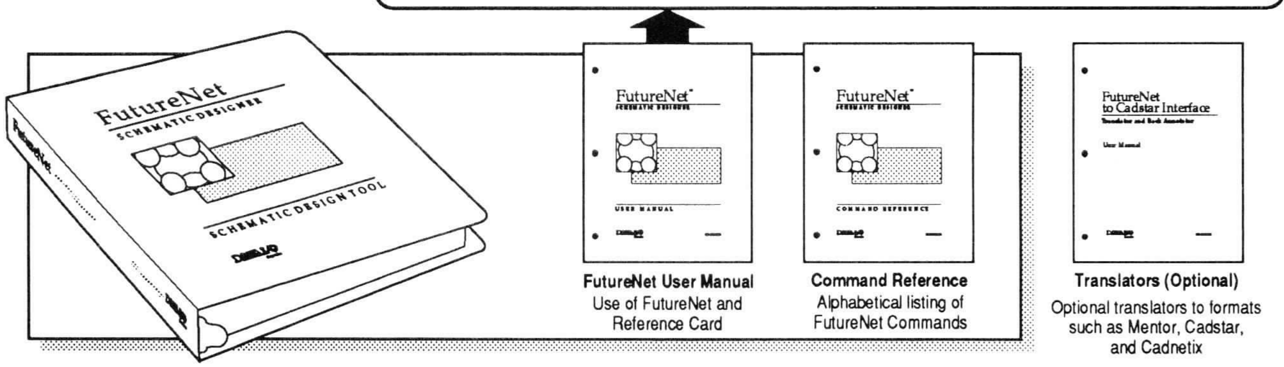
Using FutureNet — Information on using the FutureNet menus, help and command line.

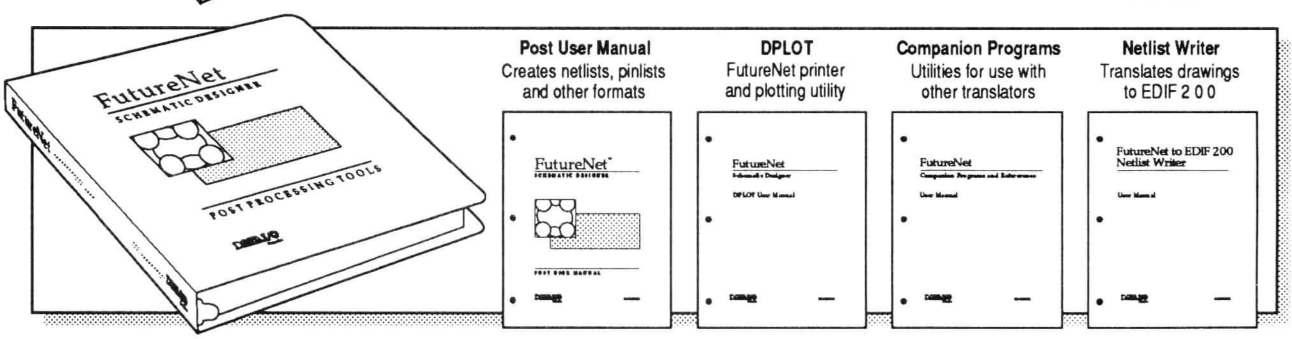
Menus and the Command Line — Basic operation of the menus and command line.

Mouse, Modes and Cursors — Information on the operation of the mouse, program modes, and the cursors used in FutureNet.

Tutorial — Nine sessions that introduce you to line drawing, libraries, alphanumeric fields and other FutureNet drawing elements.

Appendixes, **Glossary**, and **Index**





F1 Help is available online by pressing F1.

FutureNet Quick Install

The following installation procedures are for standard installations. More detailed information is provided in the *Installation Guide*.

Note: ***Update Customers:*** *FutureNet 6.1 does not require a security device. If you will be using Version 6.1 exclusively, you can remove any FutureNet security devices from your computer's parallel port.*

Install the Software

1. Place Disk 1 in a 5¼ in. high-density drive or a 3½ in. high-density drive. Assuming this drive is A, enter **a:\install** .
2. Answer the installation prompts and insert disks as required.

FutureNet Quick Start

The following information will help you get started quickly with FutureNet software. Complete operation is described in the FutureNet documentation set, outlined on the back of this card. Install FutureNet as described above or in the *Installation Guide*.

Starting FutureNet

To start FutureNet, enter **fn**. (For the standard memory version, enter **fn** from the **fn** directory. For the extended memory version, enter **fn** from the **fnx** directory.)

Exiting FutureNet

To exit FutureNet, save your work by entering **SAVEALL** then exit by entering **QUIT** . You can escape from most modes by pressing Esc .

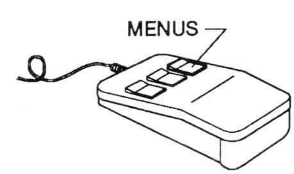
Help

Help is available by pressing F1 .

Menus

Menus are available by entering **MENU** on the command line or, in all modes except FAST, by pressing the right mouse button.

A menu with attributes is available from the FutureNet command line by entering **'A**.



Opening Drawings, Libraries and Areas

On the FutureNet command line, enter **DIR**. This command brings up a dialog box (window) that allows you to select from a directory listing to open a drawing, load a library or area, start a command file, or delete a file.

Saving Drawings and Areas

Enter **SAVE** ***drawing_name*** to save a drawing as a file named ***drawing_name*.dwg**.
Enter **[SAVE** ***area_name*** to save an area as a file named ***area_name*.ara**.

Setup

To view the setup configuration, on the FutureNet command line, enter **PROFILE**, or from the menus, select **Setup** and then **Profile**.

Emulating the Middle Mouse Button

On a two-button mouse, the middle mouse button can be emulated by pressing the left and right mouse buttons simultaneously.

Post Processing Tools

The FutureNet Post Processing Tools are available through the menus by selecting **EXPORT Generate Reports** from the top-level Command menu, or by entering **EXPORT** on the command line.

DATA I/O

Looking Ahead

In the next session, you'll learn about creating symbols similar to the ones you've been working with here. In addition, these topics are covered:

- Attributes and points of effect
- Alphanumeric text entries
- Command lists

Summary

This is a summary of the commands presented in session 1.

Library Commands

Command	Function
.LIB	Open the update library
.NOLIB	Close the update library
LIB	Open a reference library
NOLIB	Cancel all reference libraries except SYSTEM.SYM
.DIR	Display Symbol Library Directory

Symbol Commands

.L	Load a symbol from a library
*	Load a symbol already tagged for move
.M	Move a symbol
.C	Copy a symbol
.E	Erase a symbol
.R	Rotate a symbol
.RE	Reflect a symbol
.K	Cancel symbol tag

Area Commands

[D	Define an area
[C	Copy an area
[M	Move an area
[ERASE	Erase an area
[R	Rotate an area

[RE	Reflect an area
[K	Cancel area definition or area tag
[SAVE	Save an area
[LOAD	Load an area

Cursor Movement Commands

CURSOR *x,y*	Move cursor to absolute coordinates given
N *reference*	Move cursor to symbol reference
units Arrow-Key	Fast cursor movement the number of units specified and in the direction indicated

Drawing Commands

LOAD	Load a drawing
SAVE	Save a drawing
FILE	Display the current filename
CONTEXT	Restore editing context from last SAVE
CLEAR	Erase the drawing screen and reinitialize FutureNet
ERASE	Erase the drawing screen without resetting the drawing filename.
Q	Quit FutureNet and return to the operating system

The Mouse

Left Button

In MENU mode, tags symbols or areas when the cursor is in a valid location. Once a symbol or area is tagged, you enter MOVE/COPY/ERASE mode. Once inside of MOVE/COPY/ERASE mode, pressing the left button enters the function shown in the MODE status field.

Middle button

In MENU mode, reenters the command on the command line. In MOVE/COPY/ERASE mode, cycles through those functions.

If you are using a two-button mouse, you can simulate the middle button by pressing the left and right buttons simultaneously.

Right Button

In MENU mode, calls up the menus. In all other modes, cancels any function in progress and all in-progress actions, such as moving or copying.

Summary

You have now completed Session 8.

In summary, here are the commands you learned:

Command	Function
.S	Enters symbol definition mode.
.w,h	Creates a block symbol.
.Q	Exits symbol definition mode.
MXS	Moves the cursor along the X axis.
DYS	Draws a line along the Y axis.
DXYD	Draws a diagonal in dot units.
DXY	Draws a diagonal in display units.
DYL	Draws a line along the Y axis to the symbol cell boundary.
MXY	Moves the cursor along the X and Y axis.
DXR	Draws a line along the X axis to the right symbol cell boundary.
IS	Inserts a symbol element.
PgUp	Moves the first instruction to the target line.
PgDn	Moves a blank line at the end of the instructions to the target line.
# ↑ or ↓	Moves the instructions up or down one line or the number of lines specified by #.
IGxx	Inserts a graphic element.
.D	Deletes an instruction on the target line.
.I	Inserts an instruction on target line.
Home	Centers the workspace window on the coordinate marker.

Notes

Draw Instructions begin to draw from the end cursor position of the previous instruction, so they're like drawing on a piece of paper without lifting the pen. Move, Save and Restore Coordinates Instructions all move the cursor without drawing; they're commands for "picking up the pen." While most instructions are performed relative to the end cursor position of the last instruction, **MXYA** moves the cursor to absolute coordinates in the symbol cell. Save and Restore instructions allow you to return to a relative cursor position by entering a save instruction in the instruction list, then restoring it later.

Summary

You have completed symbol definition sessions. The *Command Reference* manual contains complete information on all of the symbol definition commands. In summary, here are the commands you learned:

Command	**Function**
BXR	Draws a horizontal line with a bubble to the right symbol cell boundary.
DA	Draws an arc.
DXD	Draws a horizontal line in dot units.
DXL	Draws a horizontal line left to the symbol cell boundary.
DXS	Draws a short horizontal line in display units.
DXYD	Draws a diagonal line in dot units to specified coordinates.
DY	Draws a vertical line in display units.
MXY	Moves the cursor to new coordinates in display units.
MXYD	Moves the cursor to coordinates in dot units.
MYD	Moves the cursor in dot units and changes the y coordinate.
MYS	Moves the cursor and changes the y coordinate.
RXY	Restores and moves the cursor to the previously saved x,y coordinates.
SD	Selects decrement mode.
SN	Selects no increment/decrement mode.
SXY	Saves the x,y coordinates.